Proceedings of the

Fifteenth Annual Symposium of the Pattern Recognition Association of South Africa

25-26 November 2004
Grabouw, South Africa

Proceedings of the

Fifteenth Annual Symposium of the Pattern Recognition Association of South Africa

25-26 November 2004
Grabouw, South Africa

Hosted by:
Department of Electrical Engineering
University of Cape Town
http://www.prasa.org

Edited by: F. Nicolls
ISBN 0-7992-2278-X

Member of the International Association of Pattern Recognition (IAPR)

Organisation

PRASA 2004 was organised by the University of Cape Town, Department of Electrical Engineering.

Organising Committee

Daniel Mashao (chair)
Gerhard de Jager
Fred Nicolls

Technical Committee

Fred Nicolls (chair)
Marelie Davel
Keith Forbes

Reviewers

Colin Andrew (Debeers)
Etienne Barnard (CSIR, UP)
Louis Coetzee (CSIR)
Marelie Davel (CSIR)
Gerhard de Jager (UCT)
Tania Douglas (UCT)
Keith Forbes (UCT)
Jerome Francis (UCT)
John Greene (UCT)
Ben Herbst (US)
Scott Hazelhurst (Wits)
Jonas Manamela (UNorth)
Tshilidzi Marwala (Wits)
Daniel Mashao (UCT)
Neil Muller (US)
Fred Nicolls (UCT)
Jules-Raymond Tapamo (UKZN)
Ben van Wyk (TUT)
Anton van Wyk (RAU)
Anthon Voigt (Debeers)

Table of Contents

Full papers

Poster abstracts

Texture Measures for Improved Watershed Segmentation of Froth Images

Gordon Forbes[1], *Gerhard de Jager*[2]

[1] Mineral Processing Research Unit
Department of Chemical Engineering
University of Cape Town
Private Bag, Rondebosch, 7701
gordon@dip.ee.uct.ac.za

[2]Digital Image Processing Group
Department of Electrical Engineering
University of Cape Town
Private Bag, Rondebosch, 7701
gdj@eng.uct.ac.za

Abstract

Luc Vincent's fast watershed algorithm has been successfully applied to determine the bubble size distribution in an image of froth when the bubbles are all of similar size [1]. This technique fails to work successfully when the image contains both tiny and large bubbles. A new technique is proposed which combines the use of a texture measure, the output of an initial watershed and a further watershed stage to successfully segment both tiny and large bubbles.

1. Introduction

Over the last few years, numerous advances have been made in applying machine vision technology to froth flotation. Flotation is a process used in many mines to concentrate the amount of desired mineral, before further processing (eg. smelting).

Maintaining operation of a flotation circuit at an optimal condition is not easy to achieve due to the large number of input parameters (air, level, reagents) as well as the large number of possible disturbances (mill performance, ore type changes) to each float cell. Typically, changes to the input variables are made by an experienced operator, based on a visual inspection of the froth. These operators look at features such as: bubble size, froth velocity, froth colour and froth texture.

2. Watershed for Bubble Segmentation

The watershed algorithm has been successfully used to segment individual bubbles in froth images [1, 2, 3, 4]. The froths on which the watershed algorithm was used typically consisted of large bubbles with a fairly consistent size. This is shown in Figure 1.

When the watershed segmentation is applied to images of froths that contain both large as well as tiny bubbles, the result is either an over-segmentation of the large bubbles (when minimal low-pass filtering is used in the pre-processing stage) or the under-segmentation of the very tiny bubbles (when more low-pass filtering is used). These two cases are shown in Figure 2 and 3.

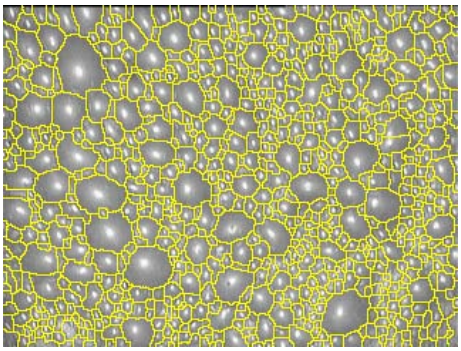


Figure 1: *An well segmented froth image. Note that all of the bubbles are of similar size.*

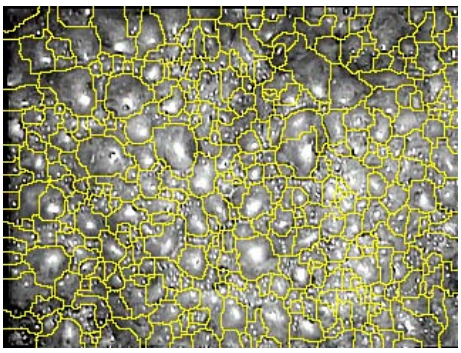


Figure 2: *An under-segmented froth image. Note that some of the regions contain many tiny bubbles.*

3. Classification of Tiny Bubbles

The output of the watershed algorithm is a set of blobs. In the case of under-segmentation, the output blobs are either a single bubble, or a collection of tiny bubbles. There is a distinct visual difference between the two types of blobs, which can be seen in Figure 4.

Because of this visual difference, it was expected that a texture measure would be able to distinguish between a blob that was a single bubble, and one which was in fact a

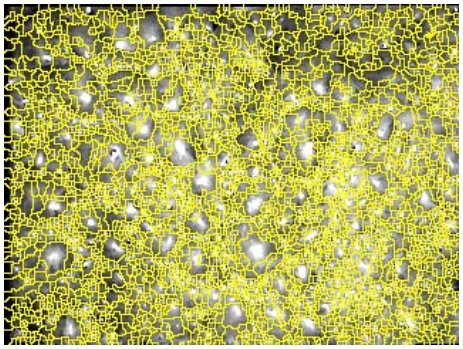


Figure 3: *An over-segmented froth image. Note that the larger bubbles have been erroneously divided into multiple regions.*

Figure 4: *Left: a single bubble. Right: many tiny bubbles.*

collection of tiny bubbles. A data set was created by hand that consisted of 108 examples of single bubbles and 107 examples of clusters of tiny bubbles (identified from the erroneous watershed segmentation).

Numerous texture measures are available to be used to classify these blobs. Such texture measures include: texture spectrum [5], grayscale co-occurrence matrices (GSCOMs) [6], Fourier methods [7, 8], Laws' texture measures [9] and others.

Since the blobs to be analysed were of irregular shape, the Fourier techniques where not used. The texture spectrum was not used as it was expected that the resultant texture spectrum would be too sparsely populated to give meaningful results when looking at very small blobs. Initial tests were conducted using Laws' texture measures and GSCOMs.

3.1. Laws' Texture Measures

Law's texture measures[9] were implemented using both the set of 25 filters generated from the following kernels:

```
L5 = [  1  4  6  4  1 ]
E5 = [ -1 -2  0  2  1 ]
S5 = [ -1  0  2  0 -1 ]
W5 = [ -1  2  0 -2  1 ]
R5 = [  1 -4  6 -4  1 ]
```

and the set of 9 filters generated from the following kernels:

```
L3 = [  1  2  1 ]
E3 = [ -1  0  1 ]
S3 = [ -1  2 -1 ]
```

These kernels were applied to both the single bubble and the tiny bubble datasets. The texture energy was calculated for every item in both datasets. The texture energy was normalised against the area of the blob being analysed so as to ensure comparable features. Thornton's separability index [10] was then used to determine which sets of features where best suited for discriminating between the two datasets. The results are shown in Table 1.

Features Used	Separability Index
E3E3 E3S3 S3S3 L3S3 L3L3	0.9581
E3E3 E3S3 S3L3 L3L3	0.9535
E3E3 S3E3 S3S3 S3L3 L3L3	0.9535
E3E3 S3S3 L3S3 L3L3	0.9535
E3E3 E3S3 S3E3 S3S3 S3L3 L3L3	0.9488

Table 1: *Separability indices for the dataset using 9 Laws filters.*

As can be clearly seen from Table 1, at least four features are required to achieve a good separability index.

3.2. Grayscale Co-occurrence Matrix Measures

Texture measures based on the GSCOM were used to discriminate between the two datasets of single bubbles and collections of tiny bubbles. The specific measures, based on the GSCOM, P, that were investigated included [6, 7]:

$$\text{Maximum Probability:} \quad \max(P_{ij}) \tag{1}$$

$$\text{Energy:} \quad \sum_{i,j} P_{ij}^2 \tag{2}$$

$$\text{Contrast:} \quad \sum_{i,j} (i-j)^2 P_{ij} \tag{3}$$

$$\text{Homogeneity:} \quad \sum_{i,j} \frac{P_{ij}}{1+|i-j|} \tag{4}$$

$$\text{Entropy:} \quad -\sum_{i,j} P_{ij} \log P_{ij} \tag{5}$$

Again, Thornton's separability index was used to determine which of these features were most suited to the classification of blobs as either a single bubble or a collection of tiny bubbles. Numerous features subsets provided 100% separability. This is shown in Table 2. All of the sets which achieved this level of separation included

Features Used	Separability Index
1 2 3 4 5	1.000
1 2 3 4	1.000
1 2 3 5	1.000
1 2 3	1.000
1 3 4 5	1.000
1 3 4	1.000
1 3 5	1.000
1 3	1.000
2 3 4 5	1.000
2 3 5	1.000
2 3	1.000
3 4 5	1.000
3 5	1.000
3	1.000
1 2 4 5	0.995

Table 2: *Separability indices for the datasets using GSCOM features.*

Feature	Separability Index
3	1.000
4	0.986
5	0.684
1	0.544
2	0.540

Table 3: *Separability indices using single GSCOM features.*

feature number 3, contrast. The separability indices for using a single feature are shown in Table 3.

Based on these results, a simple linear classifier was created that would be able to classify all of the blobs in a watershed segmentation as either a bubble or a collection of tiny bubbles. The results of running the classifier on a frame of froth is shown in Figure 5.

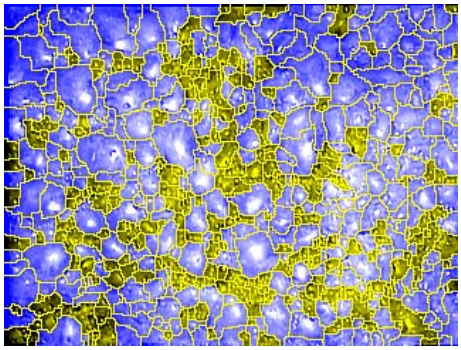


Figure 5: *Blue indicates blobs classified as single bubbles, yellow indicates blobs classified as collections of tiny bubbles. This is the same input image as in Figure 2.*

4. Contrast

A surface plot of the contrast measure (Figure 6 and 7) shows why this feature performs particularly well at discriminating between a single bubble and collections of tiny bubbles.

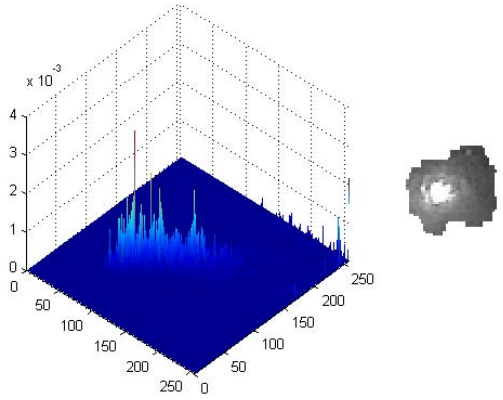


Figure 6: *Left: A typical surface plot of the GSCOM of a bubble. Right: The image it is based on.*

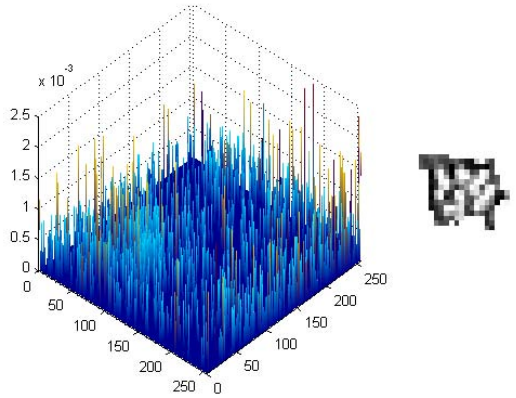


Figure 7: *Left: A typical surface plot of the GSCOM of a collection of tiny bubbles. Right: The image it is based on.*

It is clear from these figures that there is a distinct difference between the two GSCOMs. In particular, the GSCOM for the single bubble has high values along the diagonal. This is because most of the values in the image have neighbouring values which are very similar to each other. In blobs with many tiny bubbles, neighbouring pixels often have highly dissimilar values, resulting in a GSCOM with a close to uniform distribution.

It is because of these differences in the GSCOM that the contrast feature can successfully discriminate be-

tween the two types of blob. This is evident from the $(i-j)^2$ term which emphasises terms far away from the diagonal. Similarly, the homogeneity feature also performs well. It has the term $|i-j|$ in the denominator which emphasises terms along the diagonal of the GSCOM.

5. Optimising the Contrast Measurement

The traditional way of determining texture features is to first create the GSCOM, and then to calculate the desired features using the formulaes in 3.2.

This is not always necessary, especially in the case of the contrast feature. It can be calculated directly from the source image by using a single pass through the image. The squared difference between neighbouring pixels at each location are totalled and normalised against the size of the image.

6. 2nd Stage Watershed

One cannot simply create two watersheds (with different parameters) for a single image and combine the outputs. This is mainly due to the fact that there is no guarantee that the boundaries will lie on top of each other. In order to overcome this problem, the input image to the second watershed function can be modified by the output of the first in order to guarantee that the outlines of the identified blobs are properly aligned. The algorithm for merging the two watershed outputs is given below. The process flowsheet can be seen in Figure 11.

The first watershed is run on the input image with a large amount of low pass filtering. This results in the large bubbles being well segmented and groups of clusters of tiny bubbles that are under-segmented. An example of this is shown in Figure 2.

Each blob that has been identified is then classified as either a single bubble or a collection of tiny bubbles using the GSCOM contrast feature. An example of such a classification is given in Figure 5. From this classification, a mask is created which indicates which blobs are large bubbles. The mask also contains the boundary information for each of these blobs. An example of such a mask is shown in Figure 8.

The second stage of low pass filtering is applied to the original image (less than was initially used). The low-pass filtered image is then multiplied by the mask, resulting in the new input image for the 2nd stage watershed algorithm. This is shown in Figure 9.

Modifying the input to the second watershed such that the blobs that have been identified as bubbles have a maximal value (are peaks), and the edges around them have a minimal value (valleys), ensures that the watershed algorithm will result in a segmentation that will follow these specified edges. The new watershed output can then be merged with the original watershed output to generate a final segmentation which contains well segmented large and tiny bubbles. An example of such a segmentation is shown in Figure 10.

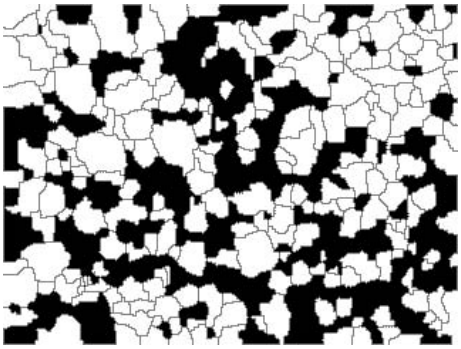


Figure 8: *The mask corresponding to the image in Figure 5.*

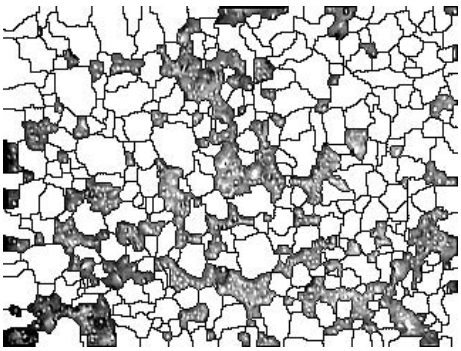


Figure 9: *The mask after it has been applied to the input image.*

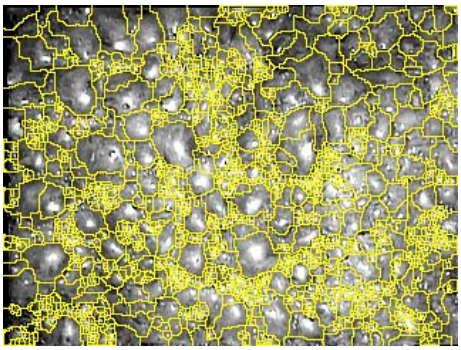


Figure 10: *Final watershed output.*

7. Discussion

Although the feature selection techniques result in 100% separability for the training data, there is no guarantee that all subsequent classifications will be 100% accurate.

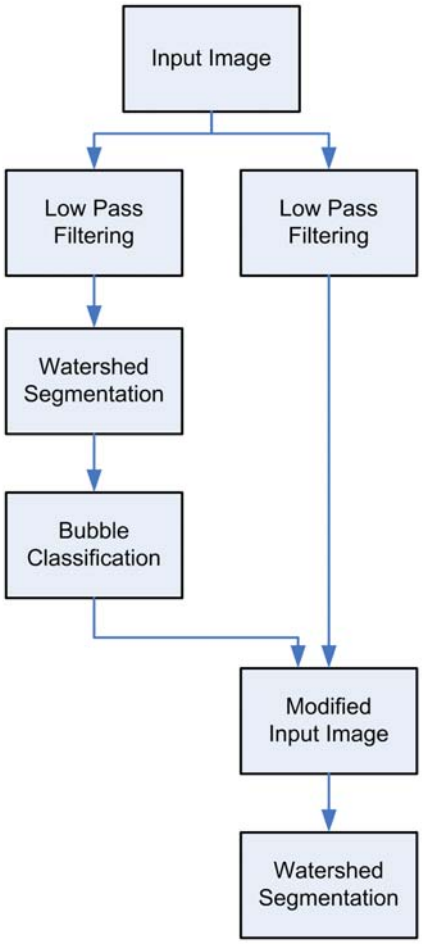


Figure 11: *Process flowsheet.*

This is because the training data is hand classified, and consists of examples of both bubbles and collections of tiny bubbles that clearly fall into one of the two categories. This allows the classifier to generalise well, but can result in the misclassifications of some blobs when a whole image is analysed.

Furthermore, it is very important to note that the algorithm is entirely constrained by the watershed algorithm as it is run in its first stage. If the blobs are oversegmented at this stage the system will fail to produce suitable results. For this reason it is necessary to use appropriate parameters (low-pass filtering, h-dome and thresholding values) for both watershed stages.

It is also important to test the algorithm on the two other extreme cases. Firstly, when the froth image consists only of tiny bubbles, and secondly, when the froth image consists only of large bubbles. This is important because the froths are dynamic and can change between a variety of states, all of which should be processed in an optimal way. These cases have both been tested, with the algorithm performing well.

A further modification to the algorithm which could result in an decrease in computation time under certain conditions is to not perform the second watershed stage if the original image is made up of more than a certain level of large bubbles. This would be the case when an image consists of large bubbles only. Under such conditions the second watershed stage would not detect many tiny bubbles anyway.

It is possible that the use of a two stage approach can bias the output bubble size distribution towards a bimodal distribution. This might erroneously occur when there are medium sized bubbles present in the froth as well as large and tiny bubbles. It is expected that the multi-stage extension of this work will handle this situation, at the expense of extra processing power. If the low-pass filtering is handled by successive filtering with a small kernel, then it will be possible to get all of the required low-pass filtered images on the creation of the low-pass filtered image for the first stage of watershedding.

One of the disadvantages of the algorithm is the increased number of parameters that are available for the user to adjust in order to obtain an optimal segmentation. These parameters fall into two categories, parameters associated with the watershed, and parameters associated with classification. Currently, the watershed parameters are kept constant, except for the amount of low-pass filtering that is performed. The low-pass filtering parameter for the second stage must be smaller than the value used in the first stage of watershedding. This places a limit on the possible values it could have. The selection of a threshold value for the classification of blobs as either single bubbles of collections of tiny bubbles could be automated by hand segmenting a large number of blobs into either single bubbles or collections of tiny bubbles. This dataset could then be used to automatically determine the optimal value for the thresholding level for a given froth.

This algorithm has shown promising results on numerous froths from a variety of ore bodies, ranging from copper to platinum. This ability to generalise makes the algorithm particularly powerful.

8. Conclusions

A new two-stage watershed algorithm that makes use of a texture based classifier can greatly improve the segmentation of images of froth when large and tiny bubbles are both present. The new algorithm has been successfully implemented on numerous froth types.

9. Acknowledgements

The authors would like to thank the following for their financial support: the NRF, the Department of Labour, Rio Tinto and UCT Department of Chemical Engineering.

10. References

[1] Benedict Wright, "The Development of a Vision-Based Flotation Froth Analysis System," M.S. thesis, University of Cape Town, September 1999.

[2] Jerome Francis, *Machine vision for froth flotation*, Ph.D. thesis, University of Cape Town, June 2001.

[3] Craig Sweet, "The application of a machine vision system to relate to froth surface characteristics to the metallurgical performance of a pgm flotation process," M.S. thesis, University of Cape Town, 2000.

[4] Pauli Sipari, "The Characterization of Flotation Froth Structure and Colour by Machine Vision - ChaCo," Tech. Rep., Helsinki University of Technology, 2002.

[5] D. He and L. Wang, "Texture unit, texture spectrum, and texture analysis," in *IEEE Transactions on Geoscience and Remote Sensing*, 1990, vol. 28, pp. 509–512.

[6] R. Haralick, "Statistical and structural approaches to texture," in *Proceedings of the IEEE*, May 1979.

[7] M. Tuceryan and A.K Jain, *The Handbook of Pattern Recognition and Computer Vision*, chapter Texture Analysis, pp. 207–248, World Scientific Publishing, 1998.

[8] J. M. Coggins and A. K. Jain, "A spatial filtering approach to texture analysis," in *Pattern Recognition Letters*, 1985, pp. 195 – 203.

[9] K.I. Laws, "Rapid texture identification," in *Proc. SPIE Image Processing for Missile Guidance*, 1980.

[10] Chris Thornton, "Separability is a learner's best friend," in *Proceedings of the Fourth Neural Computation and Psychology Workshop: Connectionist Representations*, 1997, pp. 40–47.

Tensor Voting on Sparse Motion Vector Estimation

Ian W. Guest

Department of Electrical Engineering
University of Cape Town, Cape Town
iwg@telkomsa.net

Abstract

In this paper, a tensor voting approach to robustly remove outliers and refine motion vector estimates on a video sequence is presented. The algorithm is based on tensor voting as presented and developed by Medioni and Tang [1, 2] with the novel addition of using the displacement vector as a tangential orientation to the 2D+t set of points. The results are compared to a ground truth set and are discussed.

1. Introduction

Video segmentation forms an intrinsic part of the computer vision problem. In order to solve this problem, intensity based segmentation and/or motion based segmentation can be used. This paper deals with one aspect of motion segmentation i.e. sparse motion vectors over a video scene, where a video scene is defined as a section of coherent video frames between cuts. The technique discussed is targeted to off-line processing of the video scene, and as such can make use of historic and future video frames in refining its decisions.

Sparse motion vector fields are a common concept in computer vision and there are numerous methods of obtaining estimates of them. The most common are the use of block comparison techniques [3]. In order to try use points that are not in featureless areas, a corner detector such as SUSAN [4] is used to preselect the control points. This provides a 2D+t (3D) set of input points, each one with a 2D motion vector that has been augmented to 3D where the 3rd dimension is the frame dimension.

Tensor Voting using a suitable kernel has been explored in numerous papers [5, 6, 7, 8, 9, 10, 11, 12], and has been applied to the sparse and dense motion vector problem successfully. The formulation is here used with some modifications to address the 2D+t sparse problem, but now making use of the motion vector not as a dimension to the problem, but as an orientation. Tensor Voting *encodes* the 3D set of input points into tokens. It then uses *communication* from surrounding tokens, to determine coherence or *saliency* as a basis for voting.

1.1. Contributions

In this paper, the well-documented *Tensor Voting* approach is taken to improve the sparse motion vector field. What is novel in this approach is the following:

- The utilization of the motion vector direction not as a dimension but as an orientation in the tensor voting framework.
- The use of tangential direction and not normal direction, as the normal direction to a 3D line is ambiguous while the tangential direction is not.

1.2. Organization

In section 2 the method of feature extraction is briefly described. This supplies the Tensor Voting Framework with the required input data. Section 3 deals with the background necessary to understand the Tensor Voting Framework. This is described only to give sufficient information of the Tensor Voting Framework is applicable to this problem. For more extensive information consult all of Medioni's works as referenced. Section 4 describes the Tensor Voting applied to the motion vector fields. Section 5 describes the computer simulation and uses a computer synthesized video set to determine the accuracies of the raw and processed motion vector fields.

2. Feature Extraction

The feature extraction framework uses control point selection to determine candidate locations in the image that should provide good block comparisons. It then makes use of a standard rectangular block comparison method to obtain motion estimates with a refining stage to obtain sub-pel resolution. This process is repeated over all the frames in the video scene, to obtain the 2D+t (3D) set of inputs i. Each of these i inputs have position $P_i = (x_i, y_i, z_i)$ and a direction $E_i = (u_i, v_i, k_i)$ where the z_i values increment in 3 units per video frame. The u_i and v_i values denote the motion vector in 2D, and this is augmented by $k_i = 3$. These raw values are also denoted by $E_{raw_i} = E_i$ and $P_{raw_i} = P_i$.

2.1. Control Point Selection

The control points are selected using the SUSAN method as described by Smith [4, 13]. This method looks for candidates in the image that have good corner characteristics such that the aperture problem and homogeneous region problems are to a large extent avoided. The detector makes use of a threshold to determine the number of points reported on an image. The method was developed for images, and makes use of intensity information to make its decisions. As such, no frame-to-frame information is used. Other corner detectors such as the ones used by the OpenCV library [14] and the Harris detector in the Gandalf library [15], may be used instead of SUSAN if need be.

Due to the fact that the corner detectors will produce a non-homogeneous spread of control points, a grid is overlaid on each frame and the number of control points in each grid cell is limited to one with preference given to the central one. In order to provide a reasonable spread, a fairly low threshold is chosen on the corner detector. There will still be featureless grid cells, and these are left empty.

In order to have an accurate ground-truth comparison, a synthesised image was made with a central rotating disk. The control points given by the corner detector are shown in Fig. 1.

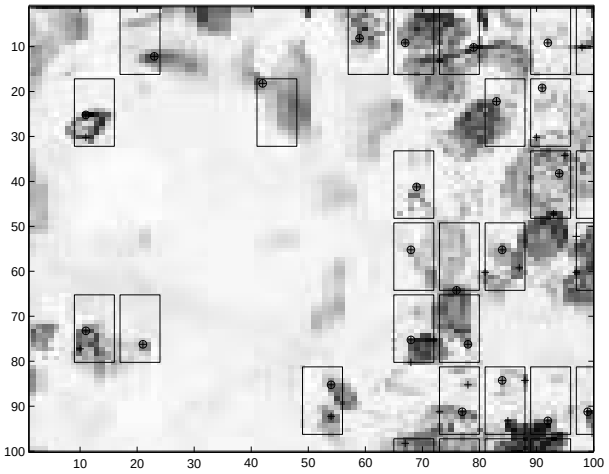


Figure 1: *Section of grid image with selected control points (o) and unselected points (+).*

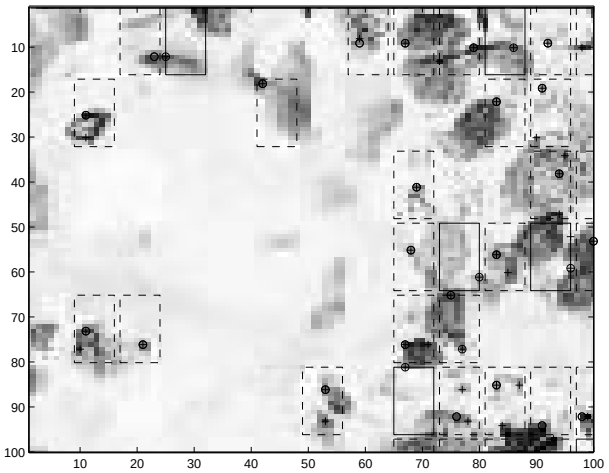


Figure 2: *Section of grid image with selected control points (o) and unselected points (+). Successful primed control points are shown in dashed cells.*

On subsequent frames, attempts are made to prime the control points in each cell with the previous frame's control point offset by the rectangular block comparison motion vector, as shown in Fig. 2.

2.2. Rectangular Block Matching

Standard block matching as described in Bhaskaran [3] is implemented with the rectangular block set at 9x9 (MxN) pixels, and the search area p as +-10 pixels. Let the pixels of the current frame be denoted as $C(x+k, y+l)$, and the pixels in the reference frame as $R(x+i+k, y+j+l)$. The cost function minimised is given in Eqn. 1

$$MAE(i,j) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)| \tag{1}$$

where $-p \leq i \leq p$ and $-p \leq j \leq p$. This describes a surface $MAE(i,j)$ where a *minimum* point defines the best match. This $MAE(i,j)$ surface also gives a number of other useful

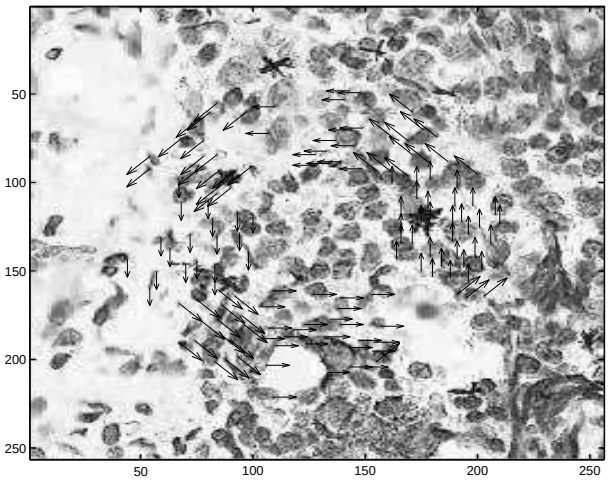


Figure 3: *Raw Motion Flow Field without refinement*

parameters once it has been transformed. The one transform is essentially an inversion (minima become maxima) and a normalization (such that all possible values lie between 0 and 1) given as

$$MAM(i,j) = 1 - \frac{MAE(i,j)}{qMN} \tag{2}$$

where q is the number of quantization levels per pixel. The $MAM(i,j)$ is then filtered using a simple high pass laplacian filter with a kernel

$$h(i,j) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{3}$$

This is similar to a matched filter looking for very *peaky* points in the surface, as well as displacing the surface such that smooth level areas are around zero (gets rid of DC). The resulting surface is described as:

$$FM(i,j) = MAM(i,j) * h(i,j) \tag{4}$$

We also define FM_{peak} as the peak value of $FM(i,j)$, and a *steepness* ratio $\frac{FM_{ratio}=FM_{peak}}{FM_{adjmax}}$ where FM_{adjmax} is the maximum intensity value directly adjacent to where FM_{peak} is located. If we make use of FM_{ratio} as a quality measure, and set the threshold to 1.2 such that anything above this threshold is taken as a good candidate and is allowed to propagate, we end with a flow field as shown in Fig. 3.

The resolution of the flow fields is one pixel. To enhance this, the surface $MAM(i,j)$ is used. The peak is found at a certain location i_{peak}, j_{peak}. the eight neighbors and the peak location is used to determine the *centre of mass* using the Eqn. 5.

$$x_{\text{offset}} = \frac{1}{Q} \sum_{i=-1}^{1} \sum_{j=-1}^{1} (i_{peak}+i) MAM(i_{peak}+i, j_{peak}+j)$$

$$y_{\text{offset}} = \frac{1}{Q} \sum_{i=-1}^{1} \sum_{j=-1}^{1} (j_{peak}+j) MAM(i_{peak}+i, j_{peak}+j) \tag{5}$$

where $Q = \sum_{i=-1}^{1} \sum_{j=-1}^{1} MAM(i_{peak}+i, j_{peak}+j)$. These offsets are added to the block matched offsets to try get

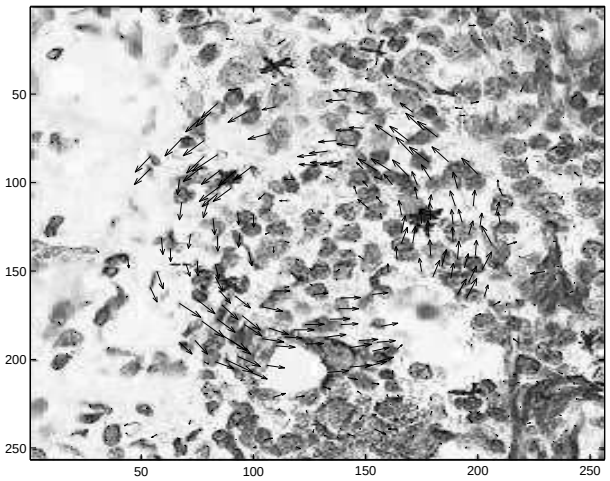


Figure 4: *Raw Motion Flow Field with refinement*

a more accurate point. Noise may affect the outcome, but the block matching essentially represents a spatial low pass filter of the size of the block, resulting in Fig. 4.

3. Tensor Voting Framework

The main underlying theory used in this paper relies heavily on second order tensor voting formulations. This theory has been derived and applied to various vision problems principally by Medioni [16], Tang [8, 9, 1], and Nicolescu [7, 17, 18].

The basis of tensor voting is to use a region of support $\mathcal{R}$ around an element to determine whether the element forms part of a structure like a curve, volume or junction while *simultaneously* allowing a measure of noise rejection. By using second order tensor representations, the second moment allows curvature and tangents on curves and surfaces to be described. Elements are represented in a tensorial way for first order tensors:

$$\mathbf{T}_i = (x_{1_i}, x_{2_i}, ..., x_{n_i}) \tag{6}$$

Extending this to the second order tensor, we get

$$\mathbf{K}_i = \mathbf{T}_i \mathbf{T}_i^T \tag{7}$$

3.1. Second Order Tensor Data Representation

Eqn. 7 is a symmetric representation, which implies that the eigenvalues are real and $1 \geq \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$. We can also represent $\mathbf{K}_i$ in an orthonormal fashion (due to its symmetry):

$$\mathbf{K}_i = \mathbf{Q}_i \Lambda \mathbf{Q}_i^T \tag{8}$$

where Λ is the diagonal matrix of eigenvalues, and $\mathbf{Q}_i$ are the right hand eigenvectors. If we drop the subscript, and look at the 3 dimensional element, we can represent Eqn. 8 as:

$$\mathbf{K} = \begin{bmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \hat{e}_1^T \\ \hat{e}_2^T \\ \hat{e}_3^T \end{bmatrix} \tag{9}$$

or multiplying out and grouping we get:

$$\mathbf{K} = (\lambda_1 - \lambda_2)\mathbf{S} + (\lambda_2 - \lambda_3)\mathbf{P} + \lambda_3 \mathbf{B} \tag{10}$$

where **B** is the ball component having no particular orientation. This can be visualised as a sphere and defined by $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T$. The plate component is given by **P** has no orientation around two of the three axis. This can be visualised as a disk/plate and defined by $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T$. The last component is the stick component given by **S** which is aligned to the $\hat{e}_1$ axis. This can be visualised as a stick/line and is defined by $\hat{e}_1\hat{e}_1^T$. In Eqn. 10, the eigenvectors and eigenvalues describe an ellipsoid as in Fig. 5.

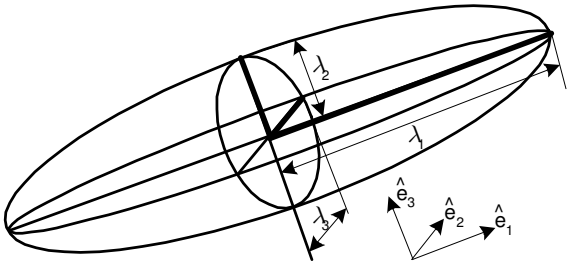


Figure 5: *3D ellipsoid tensor representation.*

For the 3D case the different components of this equation allow us to differentiate between a point that has no firm direction (**B**), points on surfaces or plates (**P**) and points that are on lines and curves (**S**). These coefficients to these terms are the saliency terms and have the following characteristics:

1. **Point saliency**. This is characterized by very similar eigenvalues ($\lambda_1 \approx \lambda_2 \approx \lambda_3$) and has no preferred direction. The saliency value is given by λ_3.
2. **Curve saliency**. This is characterized by two similar eigenvalues larger than the third ($\lambda_1 \approx \lambda_2 > \lambda_3$). The measure is $\lambda_2 - \lambda_3$ being large in value. This indicates a strong curve directionality (belonging to a line), and the normal unit vectors are given by $\hat{e}_2$ *and* $\hat{e}_3$. *The saliency value is defined as* $\lambda_2 - \lambda_3$.
3. ***Surface saliency***. *This is characterized by one eigenvalue larger than the second and third (*$\lambda_1 > \lambda_2 \approx \lambda_3$*). The measure is* $\lambda_1 - \lambda_2$ *being large in value. This indicates a strong directionality (belonging to a surface), and the normal unit vector is given by* $\hat{e}_3$. *The saliency value is defined as* $\lambda_1 - \lambda_2$.

The 2D case, which has the smallest dimensionality, is defined as:

$$\mathbf{K} = (\lambda_1 - \lambda_2)\mathbf{S} + \lambda_2 \mathbf{B} \tag{11}$$

where only points and curves are present, not surfaces as there are no plate tensors.

3.2. Second Order Tensor Data Communication

Tensor voting makes use of data or tensor communication in order that elements at various points can vote at other points. A vote decays with *distance* and with *curvature*. A kernel that can describe this allows the *Vote Strength* to be described as

$$VS_{stick}(s, \kappa) = \exp^{-\frac{(s^2 + \alpha\kappa^2)}{\sigma^2}} \tag{12}$$

where s is the arc length, κ is the curvature, α is a curvature scaling factor and σ is the radial distance scaling factor. This form of vote strength is applicable if the orientation and position

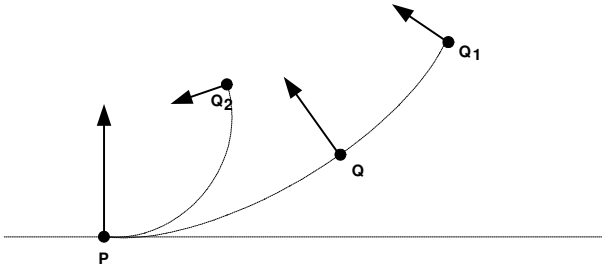


Figure 6: *2D vote strength if directional information is known.*

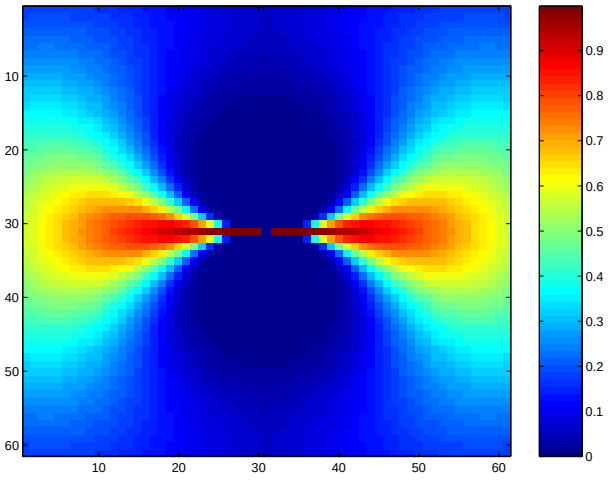


Figure 7: *2D* $VS_{stick}(s, \kappa)$.

of the voter is known explicitly as is the case of the stick data representation **S**.

If we look at the 2D case, the reduction in vote strength in Fig. 6 of voter P on votee Q is seen to get less for increased radial distance (Q_1) and increased curvature (Q_2). If the effect over the xy plane in 2D is mapped, then Fig. 7 indicates very little strength broad-side to the voter (high curvature) and a general radial decay (greater distance).The radial distance and curvature are given as:

$$r = \frac{l}{2sin(\theta)}; \quad \kappa = \frac{1}{r}; \quad s = 2r\theta \tag{13}$$

Furthermore, the *scale* is denoted by σ. This determines the decay of the vote strength with distance and curvature. An additional constant α also appears to scale the radial decay and curvature decay in relation to each other.

Up until now, we have been dealing with the vote *strength*. In order to make this into a full tensor form as specified in Eqn. 11, the stick vote becomes:

$$\mathbf{V}_{stick} = VS_{stick}\vec{N}\vec{N}^T \tag{14}$$

We can extend Eqn. 14 to 3D by noting that the stick vote strength VS_{stick} is radially symmetric around the x axis. Essentially, all dimensions higher than 2D can be reduced to a 2D problem by rotating the 2D plane defined by the vector $\vec{M}$ and the receiver point Q such that it is the 2D plane defined by xy in Fig. 8.

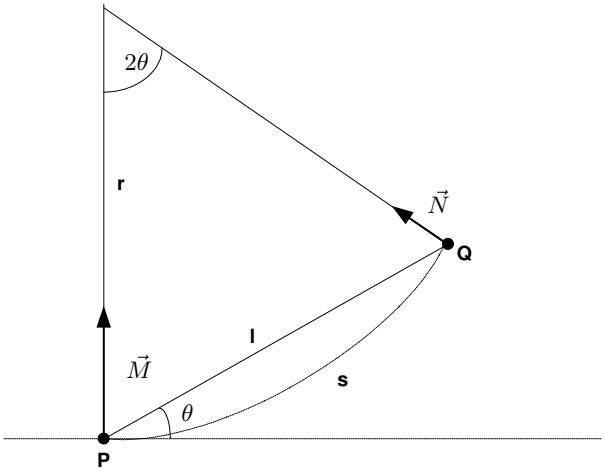


Figure 8: *2D stick geometry.*

In this application, only stick voting is used, so the ball (**B**) and plate (**P**) terms are not further elaborated on. More information can be found in the references noted in the beginning of this section.

The votes are iteratively done on the votees, where the votee set comprises of all the elements $\mathbf{T}_i$. The voters are drawn from the same set within a radius such that contributions beyond this radius are negligible (function of σ). All the voter's votes are tensorially added, and then the eigenvalues and eigenvectors are determined allowing the mentioned saliencies to be computed. These saliencies determine the characteristics of the votee point in relation to its surrounding elements. In the 3D case, a votee may be characterised as being an independent point, or point within a volume with its *point* saliency, or as a point on a 3D surface by its *surface* saliency, or as a point on a curve or line with its *curve* saliency.

4. Motion Vector Estimation

Once the input elements are found in both position P_i and direction E_i, they are encoded as i tensor elements $\mathbf{T}_i$ with tangential directions. If we determine a voter at point P and a votee at point Q (both elements from P_i), the respective tangential unit vectors are $\vec{V}$ and $\vec{U}$ (both unit vector elements from E_i). We shift the whole system such that P is at the origin. A vector $\vec{B} = \vec{P} - \vec{Q}$ now describes the votee position. We can convert the tangential vector $\vec{V}$ into the normal form using $\vec{M} = (\vec{V} \times \vec{B}) \times \vec{V}$. Using the normal vector $\vec{M}$, we can now compute the normal vote strength at the votee point Q according to Eqn. 12. Due to the fact of trying to preserve the direction of E_i in the region $\mathcal{R}$ around the votee, the direction of the voter ($\vec{V}$) is preserved at the votee site Q. The stick vote becomes:

$$\mathbf{V}_{tstick} = VS_{stick}\vec{V}\vec{V}^T \tag{15}$$

In this formulation, all votes are collected per votee and summed, including the votee voting for itself (perfectly aligned — zero distance in the kernel). The eigenvectors and eigenvalues are found for all tensor elements $\mathbf{T}_i$. In this formulation, with all the vectors $\vec{V}$ pointing in the same direction, the saliency sought is a high $sal = \lambda_1 - \lambda_2$, and the direction given by $\hat{e}_1$. The maximum saliency is found over all i, and used to

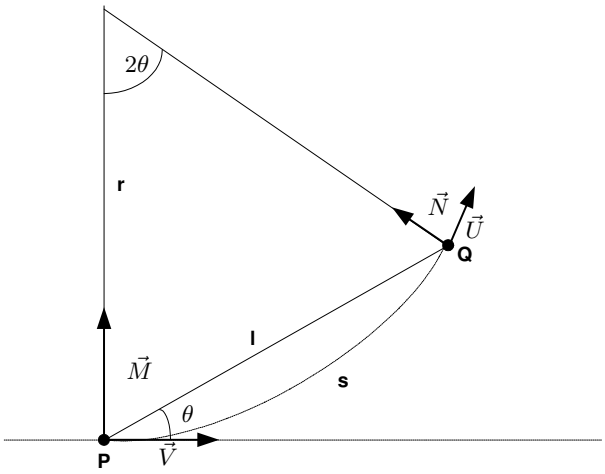


Figure 9: *2D stick geometry with tangential vectors.*

set a threshold of 1/3 of the maximum saliency. Any votee locations found to have saliencies below this are discarded (noise rejection), and the rest have their E_i replaced with $\hat{e}_1$ for each i. The E_i must be scaled such that $k_i = 3$. This forms the new improved 3D set of directions E_{ref_i}.

5. Simulation and Results

A computer simulation in MATLAB was written to process the synthetic image using the 'tissue.png' image as shown in Fig. 1. The central disk section of this image was rotated from frame to frame, while the rest was held static. For the synthetic image, the exact ground truth was determined for comparative results and for each P_i an actual E_{actual_i} was determined.

The synthetic image was processed with the control point extraction, block matching, and tensor voting, to get a refined set E_{ref_i} for each P_i that was not rejected as having a saliency that was too low. Comparisons were made between the raw and actual and refined and actual motion vector fields as contained in E_{actual_i}.

The method of comparison between the actual motion vectors and calculated motion vectors was to look at the dot product between the two motion vectors as given by $\alpha_{ref_i} = acos((u_{ref_i}, v_{ref_i}, 1) \bullet (u_{actual_i}, v_{actual_i}, 1))$. The same can be found for the raw values as $\alpha_{raw_i} = acos((u_{raw_i}, v_{raw_i}, 1) \bullet (u_{actual_i}, v_{actual_i}, 1))$. The mean and standard deviation values in degrees are given in Table 1.

Table 1: *Comparative angular errors.*

Case	Mean Error	STD Error
Raw,$\sigma = 15$,$\alpha = 1000$	10.0	9.8
Ref,$\sigma = 15$,$\alpha = 1000$	7.3	8.8

An improvement is clearly noticeable compared to the raw measurements, with an improvement of 37%.

6. Conclusions

Tensor voting can be adapted to simultaneously remove outliers, and refine sparse motion vector fields over a video scene. A minor adaptation of the tensor voting framework was minor, with only 2 degrees of freedom (σ and α) needing to be considered. An overall improvement in motion vector field accuracy was achieved on the synthetic image where ground truth was known.

7. Acknowledgements

Due acknowledgement is given to the Foundation for Research and Development (FRD), and to TSS for financial support.

8. References

[1] Chi-Keung Tang and Gerard Medioni, "Curvature-augmented tensor voting for shape inference from noisy 3D data," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 24, no. 6, JUNE 2002.

[2] Chi-Keung Tang and Gerard Medioni, "Robust estimation of curvature information from noisy 3D data for shape description," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 426 – 433.

[3] V Bhaskaran, K Konstantinides, *Image and Video Compresision Standards - Algorithms and Architectures*, Kluwer Academic Publishers, 1997.

[4] S.M. Smith, J.M. Brady, "SUSAN - a new approach to low level image processing," Tech. Rep.TR95SMS1, 1995.

[5] Jiaya Jia and Chi-Keung Tang, "Inference of segmented color and texturedescription by tensor voting," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 26, no. 6, JUNE 2004.

[6] Pierre Kornprobst and Gerard Medioni, "A 2D+t tensor voting based approach for tracking," in *Proceedings. 15th International Conference on Pattern Recognition*, pp. 1092 – 1095.

[7] Mircea Nicolescu and Gerard Medioni, "Layered 4D representation and voting for grouping from motion," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 25, no. 4, APRIL 2003.

[8] Chi-Keung Tang and Gerard Medioni, "Extremal feature extraction from 3-D vector and noisy scalar fields," pp. 95 – 102, Oct 1998.

[9] Chi-Keung Tang, Gerard Medioni, Mi-Suen Lee, "Epipolar geometry estimation by tensor voting in 8D," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 502 – 509.

[10] Wai-Shun Tong, Chi-Keung Tang, Gerard Medioni, "First order tensor voting, and application to 3-D scale analysis," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I–175 – I–182.

[11] Wai-Shun Tong, Chi-Keung Tang, Philippos Mordohai, and Gerard Medioni, "First order augmentation to tensor voting for boundary inference and multiscale analysis in 3D," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 26, no. 5, MAY 2004.

[12] Wai-Shun Tong, Chi-Keung Tang, and Gerard Medioni, "Simultaneous two-view epipolargeometry estimation and motion segmentation by 4D tensor voting," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 26, no. 9, SEPTEMBER 2004.

[13] S.M. Smith, "Edge thinning used in the susan edge detector," Tech. Rep.TR95SMS5, 1995.

[14] "Open source computer vision library reference manual (opencv)," 2001.

[15] Philip F McLauchlan, *Gandalf: The Fast Computer Vision and Numerical Library*, Imagineer Software Ltd.

[16] Gerard Medioni Chi-Keung Tang Mi-Suen Lee, "Tensor voting: Theory and applications," in *12eme Congres Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, February 2000.

[17] Mircea Nicolescu and Gerard Medioni, "4-D voting for matching, densification and segmentation into motion layers," in *Proceedings. 16th International Conference on Pattern Recognition*, pp. 303 – 308.

[18] Mircea Nicolescu and Gerard Medioni, "Motion segmentation with accurate boundaries - a tensor voting approach," in *Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I–382 – I–389.

Visual Hull Surface Estimation

Phillip Milne Fred Nicolls Gerhard de Jager
Digital Image Processing Group
Department of Electrical Engineering
University of Cape Town
South Africa
phillip@dip.ee.uct.ac.za

Abstract

This paper describes a technique used to approximate the surface of an object's visual hull. The hull's basic structure is initially represented by a partitioned 3D space using voxels. The marching cubes algorithm then assigns polygonal patches to surface voxels depending on a certain criteria. It is less computationally intensive to manipulate a few triangular patches than a number of six-faced voxels. The visual hull model is further refined by applying a binary search to the vertices of each surface that improves the positional accuracy of each vertex. The method is applied to a small plastic cat using a 5-camera system and results are shown.

1 Introduction

In computer graphics and machine vision, it is often necessary to model real world objects. The information contained in a 3D model can be useful in systems that require any one of the following:

- Multimedia Content: making movies or computer games with 3D objects.
- Classification: deciding on the type of shape of an object.
- Recognition, e.g. recognising a person's gestures.

Information on the structure of an object is contained in any 2D image of that object (an image of a small plastic cat can be seen in figure 1). The information from a single view can be used towards approximating a 3D model of the object. The accuracy of the model is improved with an increase in the number of different views available.

Figure 1: Multiple image views of a small plastic cat.

This paper discusses some of the concepts needed in order to build a 3D model of an object from silhouette images. These silhouette images are obtained from an accurately calibrated camera system [4] and used to generate a voxel based model. This model is used as an initial estimate of the visual hull of an object. Surface voxels are then replaced with triangular patches. This has the effect of smoothing the visual hull. A binary search further refines the accuracy of the visual hull.

2 The Visual Hull

An object's *visual hull* is a geometric representation of its structure. This representation is an upper bound estimation and is not necessarily an exact copy of the object. Visual hulls cannot capture concavities, such as the inside of a tea cup.
The visual hull can be computed from the silhouettes of an object taken from differing viewpoints. Silhouettes are extruded to create cone-like volumes which intersect to form the object's visual hull. The accuracy of the visual hull is refined as more views are added.
Sample silhouette images can be seen in figure 2.

These silhouettes were obtained by thresholding grey scale images.

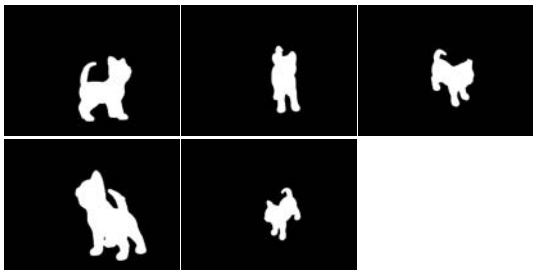


Figure 2: Five silhouette images of a plastic cat

3 Voxel Representation

One method to represent an object's visual hull is to create a spacial occupancy map using a number of volume elements or *voxels*. The concept involves starting with an initial cubic volume and "carving away" parts that are not included in the visual hull. Voxels are the 3D equivalent to the pixels that make up a 2D image. A single voxel element is a scaled cube with 3D coordinates in some coordinate system. It has ordered vertices (as shown in figure 3) that can each be assigned a different value. These values decide the occupancy of the voxel.
The three voxel occupancy categories are:

- Voxel is completely inside the visual hull
- Voxel is completely outside the visual hull
- Voxel has the surface of the visual hull running through it

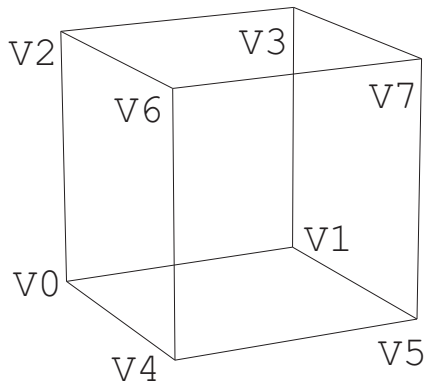


Figure 3: A voxel with ordered vertices.

The size of the smallest voxel determines the resolution of the visual hull. As the number of subdivisions increases, so does the amount of processing time require to compute the model.
Using a voxel structure to represent an object has certain flaws that can be overcome provided that the system has a basic knowledge on the shape of the object being modeled. When objects are snake-like or have sharp peaks, they appear discontinuous. This flaw can be overcome by making the voxels smaller and hence increasing the resolution of the visual hull.

3.1 The Octree Structure

An octree is a tree data structure [1] that can be used to represent the voxel based visual hull model. This involves specifying an initial voxel, known as the *universe cube*. The universe cube is split into 8 suboctants. Each suboctant is iteratively subdivided until some predefined limit is reached. This idea is more clearly illustrated in figure 4.

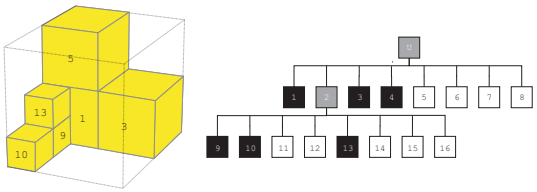


Figure 4: A simple volume represented by an octree and the corresponding tree structure with nodes [**?**].

Figure 5 illustrates how the accuracy of the visual hull can be improved by increasing the number of subdivisions. The first visual hull was computed with 2 subdivision levels, the second with 3 and the third with 4. The larger the number of subdivisions, the more computationally expensive the algorithm becomes.

4 Surface Approximation

Using a number of voxels to represent a visual hull is a simple concept, and easily implemented. The large number of voxels used to make up the visual hull causes other systems using these models to become sluggish. A solution to this problem is to apply the *marching cubes* algorithm to all surface voxels of the visual hull.
The marching cubes algorithm approximates a polygonal surface that passes through the surface

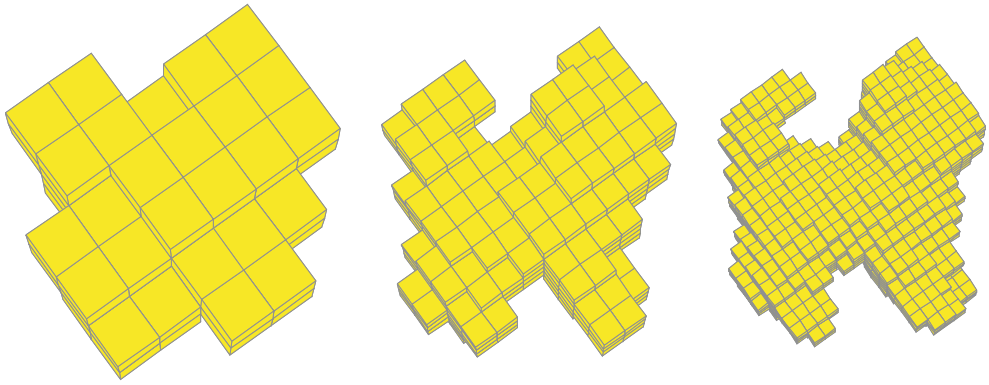


Figure 5: The 3D voxel based visual hull model of the imaged cat at 3 different subdivision levels.

voxel. It also results in a smoothing of the voxel corners along the surface of the visual hull [5].

4.1 Marching Cubes

The marching cubes algorithm was first used to visualise mathematical equations. It assigns a surface to a voxel depending on the voxels corner values and how they are arranged. Surfaces are made up of a number of triangular patches. The three vertices of the triangle are ordered using the *right hand rule*. This means that they are ordered anticlockwise about the surface normal. The marching cubes idea can be more easily illustrated by its 2D equivalent *marching squares*.

4.1.1 Marching Squares

Figure 6 shows how a line is matched to a square, depending on its corner configuration. Solid black circles indicate that a corner is part of the target, while the absence of a circle indicates that a corner is background.
Four images of the same silhouette view that have been placed inside a *universe square* are depicted in figure 7. Each universe cube has been subdivided 3 times. Black indicates the computed 2D visual hull while grey is the colour of the actual visual hull. Each image is described below:

A The silhouette image view inside a universe square.

B The 2D implementation of a voxel based visual hull.

C The marching squares visual hull model.

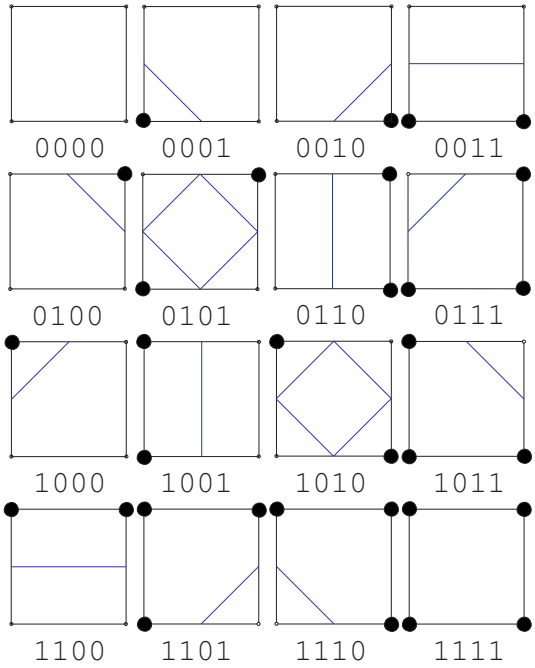


Figure 6: Marching squares.

D The marching squares model is refined using a binary search.

The concept of a binary search, shown in figure 7.D, is used in the final implementation of the marching cubes algorithm and is discussed in more details in a later section.

4.1.2 Cube Categories

The corner values of the voxel determine the type of surface that passes through the voxel. Of the 256 corner combinations, there are 15 unique categories

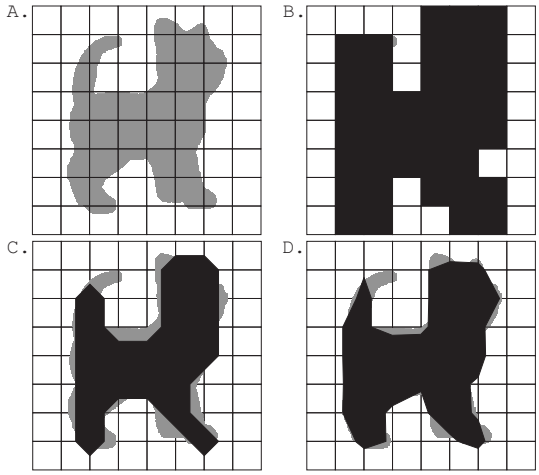


Figure 7: Marching squares sample.

that voxels can be classified into [3]. Each different category has a different surface which passes through it (see figure 8).

4.1.3 Ambiguities

Using the 15 surfaces illustrated in figure 8 alone results in various ambiguities [6]. These are easily visible in the resultant visual hull in the form of holes [2]. Figure 9 shows why these ambiguities occur, and also shows how they can be overcome. Using the ideas displayed in this figure, a new table of 33 surfaces can be created. (see figure 10). Using this table solves the problems that arise in ambiguous cases. Figure 11 shows the effects of applying the marching cubes algorithm to the voxel based visual hull models in figure 5. These models are smoother than the voxel based versions from figure 5. The patches are coloured according to their voxel corner category. The improvement on the voxel model is quite evident in the middle marching cube visual hull model.

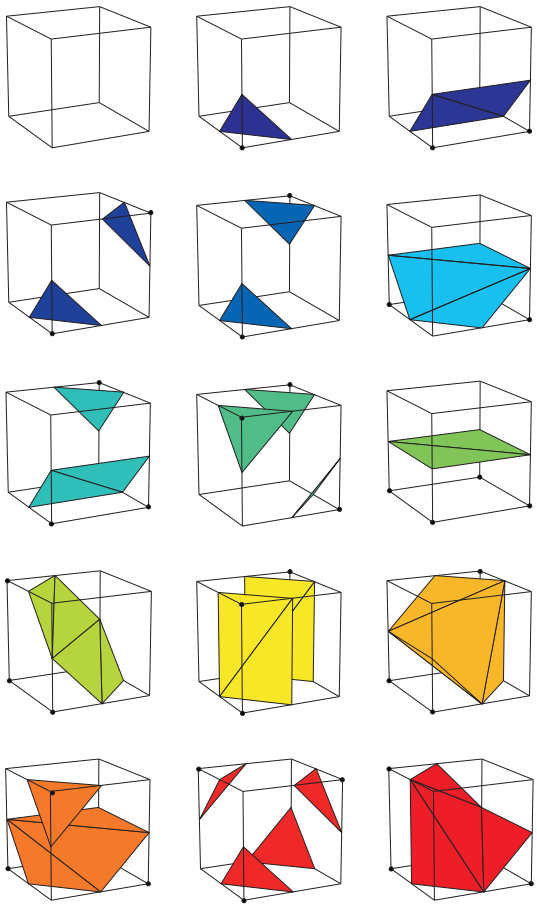


Figure 8: The 15 marching cube surfaces [7].

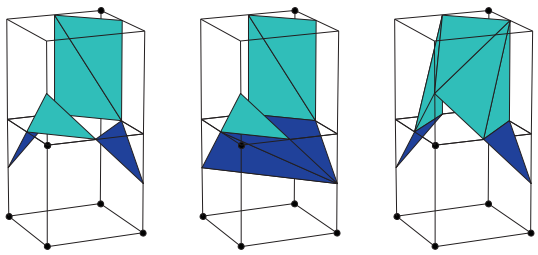


Figure 9: One of the ambiguous marching cube surfaces.

4.1.4 Refining the Surface

The accuracy of the visual hull model can be improved by applying a binary search to the vertices of each triangular patch making up the hull surface. This requires that each vertex in the assigned surface be projected back into the silhouette images and tested. The vertices are then moved along the cube edges in a direction determined by the result of their projection test. Figure 7D. shows the effect that a single iteration of this search has on a 2D model.

The result of applying this binary search to the marching cubes visual hull from figure 11 can be seen in figure 12. The effects are more noticeable in the first visual hull model. The higher the number of voxel subdivisions, the less noticeable the effects of this search become.

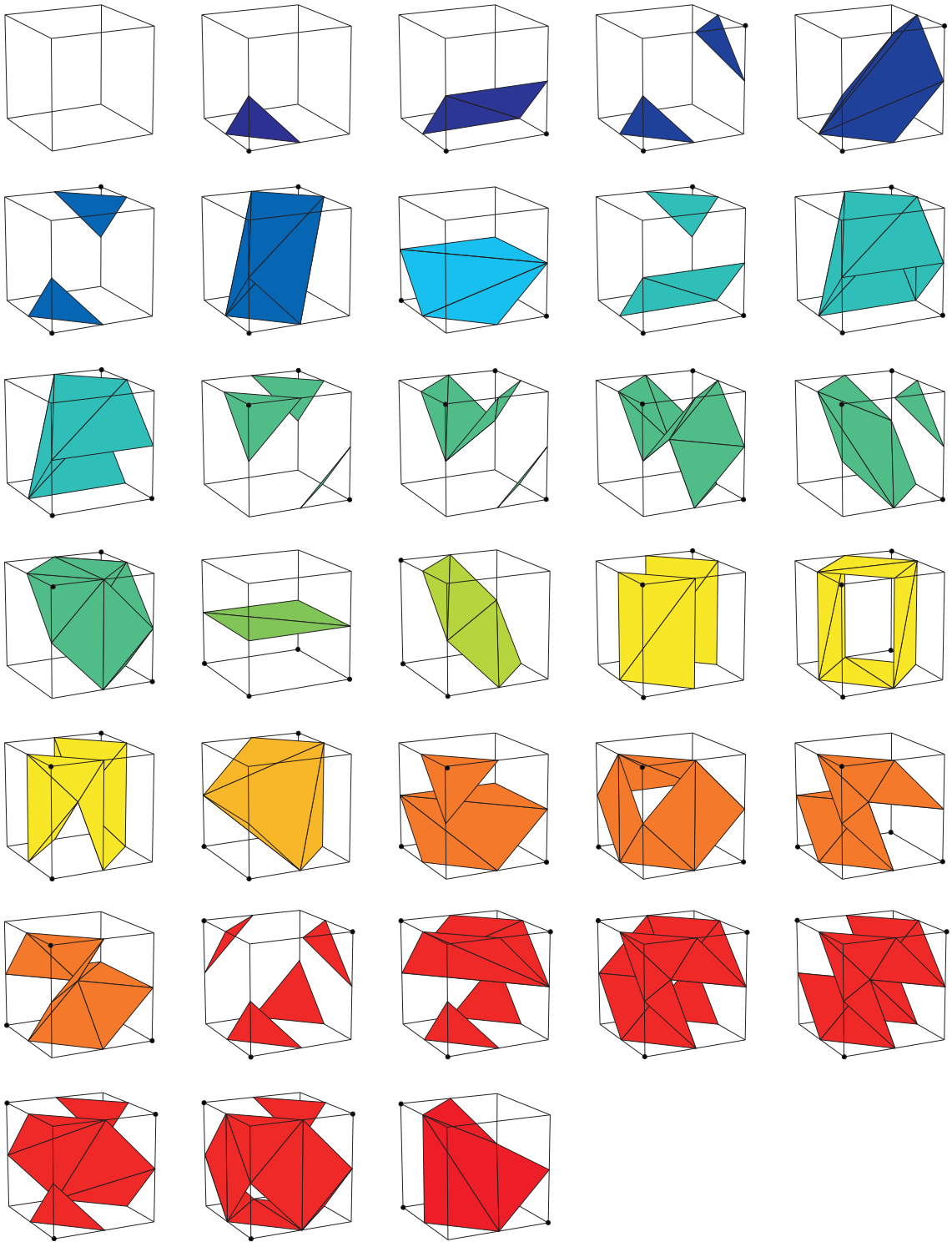


Figure 10: The 33 marching cube surfaces[6]

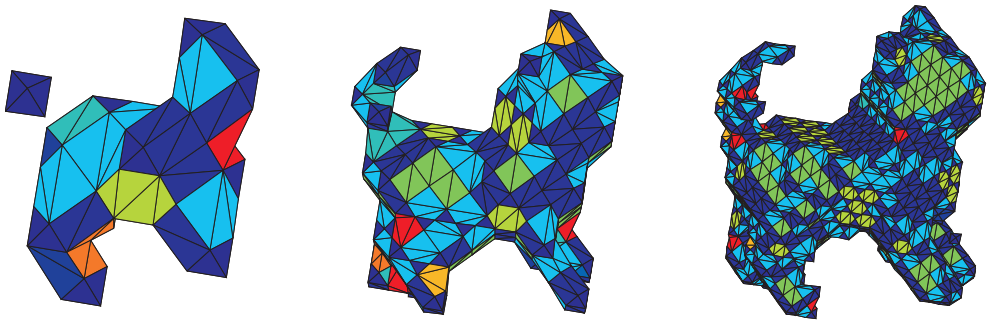


Figure 11: The 3D marching cube visual hull model of the imaged cat at 3 different subdivision levels.

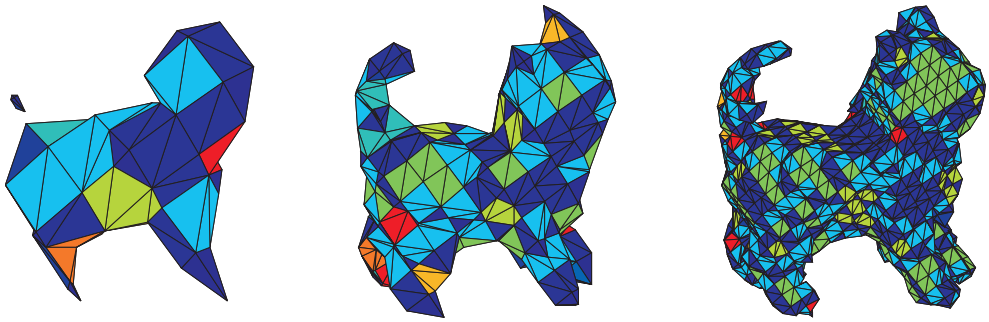


Figure 12: The 3D marching cube visual hull model of the imaged cat at 3 different subdivision levels, with 4 iterations through a binary search.

5 Conclusion

Marching cubes reduces the amount of time required by a computer to manipulate a visual hull. It produces more accurate models with smoother surface than the voxel based method (comparing figure 5 with figures 11 and 12). The binary search element in the algorithm improves the visual hull accuracy on voxel models with low subdivision levels, but its effects are less noticeable at higher levels.

6 Acknowledgements

I would like to thank the De Beers Group Technical Support for their financial support and input.

References

[1] Narendra Ahuka and Jack Veenstra, *Generating octrees from object silhouettes in orthographic views*, Pattern Analysis and Machine Intelligence **11** (1989), no. 2, 137–149.

[2] Ken Brodlie and Jason Wood, *Computer graphics*, Recent Advances in Volume Visualization (2001).

[3] R. Scateni C. Montani and R. Scopigno, *Discretized marching cubes*, Internet Paper.

[4] Keith Forbes, Anthon Voigt, and Ndimi Bodika, *An inexpensive, automatic and accurate camera calibration method*, Proceedings of the Thirteenth Annual South African Workshop on Pattern Recognition, PRASA, 2002.

[5] William E. Lorensen and Harvey E. Cline, *Computer graphics*, Marching Cubes: A High Resolution 3D Surface Reconstruction **21** (1987), no. 4, 163–166.

[6] Antonio Wilson Vieira Thomas Lewiner, Helio Lopes and Geovan Tavares, *Efficient implementation of marching cubes' cases with topological guarantees*, Ph.D. thesis, Pontifical Catholic University, INRIA, 2004.

[7] Kwan-Yee Kenneth Wong, *Structure and motion from silhouettes*, Ph.D. thesis, University of Cambridge, 2001.

Structure and motion from SEM: a case study

Fred Nicolls

Department of Electrical Engineering, University of Cape Town
fnicolls@eng.uct.ac.za

Abstract

This paper describes an attempt to reconstruct a 3-D object from a set of 35 images captured using a scanning electron microscope. Point matching over overlapping triples of views is used to obtain an initial reconstruction, which is refined using bundle adjustment with the added knowledge that the sequence is closed. Intrinsic camera parameters are estimated via autocalibration under an affine assumption. Good results for the final metric reconstruction are obtained.

1. Introduction

Modern computer vision provides many techniques for reconstructing 3-D objects from uncalibrated sequences of images. This paper describes an attempt to solve the structure and motion problem for a set of images captured from a scanning electron microscope (SEM).

The test object used is a small aluminium block, of the order of 0.1mm in size, obtained by crudely milling away portions of a larger aluminium slab. To obtain multiple views, this object was mounted on a turntable and rotated in the view of the SEM. Figure 1 shows two frames from the sequence, which was 35 frames long in total. The images, each of dimension 1024×768, were taken at approximately equal angle increments of about 10 degrees. However, the motion of the turntable was neither precisely controlled nor monitored, and upon inspection it was evident that there was also no single axis of rotation.

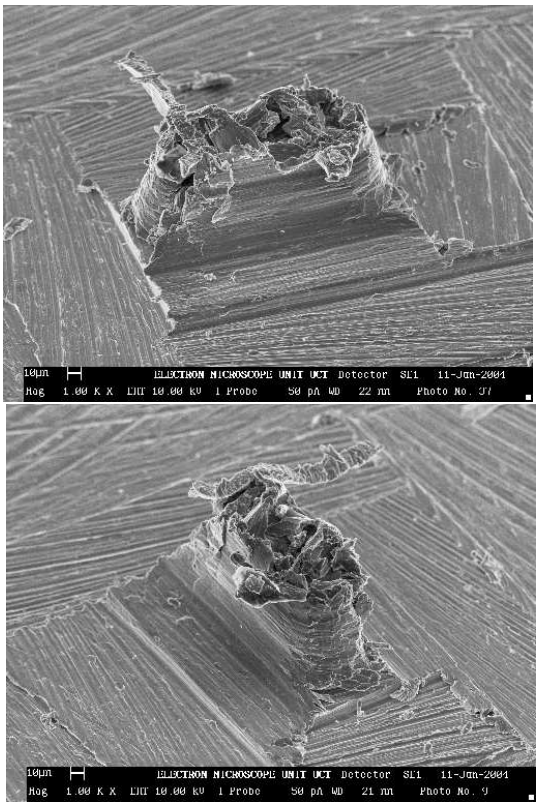


Figure 1: Two frames from the 35 frame input image sequence.

A scanning electron microscope operates under very different principles from optical imaging systems, and one cannot take it for granted that the assumptions made in computer vision will be appropriate. This issue is discussed in Section 2. From the outset, however, we make the assumption that a geometrical optics model is appropriate — if this is not the case then we expect to find inconsistencies in the application of the theory. In a sense, to obtain an accurate reconstruction is probably one of the best ways to validate that the assumptions of projective geometry are appropriate.

The approach taken in this work is to use standard feature-point based structure and motion techniques to obtain a projective reconstruction of the scene and the effective camera positions. This includes conventional outlier rejection, both for 2-view matches as well as for 3-view triplets. The only real complication in the reconstruction relates to the length of the sequence: there is no obvious way to combine the results from all 35 camera views into the reconstruction. Nonetheless, a simple method of reconstructing overlapping triples and stitching them together was adopted which, while not being without problems, provides a reasonably good reconstruction. A metric upgrade is then obtained by autocalibration. Details on all these methods are presented in Section 3.

The final result is a reconstruction of 3-D points on the surface of the object, as well as estimates of the locations of the (effective) cameras used to capture the images. These results are presented in Section 4.

2. Scanning electron microscopy

In scanning electron microscopy (SEM), a beam of electrons is used to form an image of a specimen. Since the SEM is a point-source (type 1) scanning microscope, at any time the illuminating beam is focused to a small spot on the object. This results in a signal which can be detected. The spot is scanned across the specimen and an image built up.

Depending on the configuration and the detectors used, the signal can provide information on a number of physical characteristics of the specimen, such as topography and atomic composition. In the secondary electron mode of operation, the beam results inelastic excitation of atoms to such high energy levels that electrons can overcome the work function and escape. These secondary electrons (which themselves have low energies $\leq$ 50eV) are then usually collected by a Everhart-Thornley detector. This detector consists of a positively-biased grid which attracts the secondaries, accelerates them onto a scintillator, and records the response. Figure 2 depicts the configuration for the beam at two different positions on the specimen.

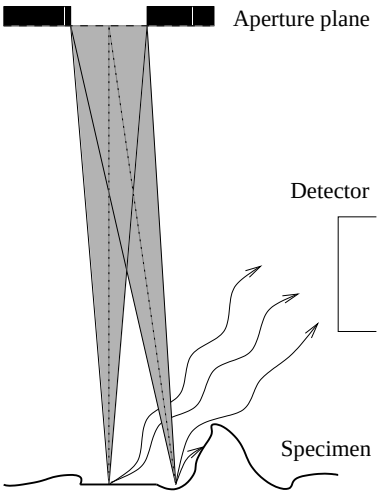


Figure 2: A SEM detecting secondary electrons.

It is not obvious that a SEM will produce an image that bears any resemblance to optical images — the mechanisms of image formation are entirely different. However, in [6, 5] it was demonstrated (for an autofocus application) that there is indeed some degree of equivalence: the notion of a point-spread function can be developed for a SEM, and is entirely analogous to that of conventional light optical systems. Furthermore the SEM conforms to the basic principles of geometric optics, although often with parameter values vastly different from those commonly encountered in optics. The apparent ease with which non-specialists interpret scanning electron micrographs bears testimony to this similarity.

Interestingly, with regard to photometric comparison one should regard the detector in SEM as being equivalent to a (possibly diffuse) light source in optical imagery, and the electron beam as being equivalent to the camera. This can be deduced though closer inspection of Figure 2: when the topography of the specimen causes an occlusion between the position of the beam on the specimen and the detector, the number of secondary electrons reaching the detector is decreased. This causes a reduction in the signal, which appears as a region of reduced intensity, or a shadow, in the resulting image.

3. Reconstruction method

The sequence of operations used to reconstruct the object from the image sequence is now described.

Good introductions to the theory and techniques of multiple view geometry can be found in [2] and [4]. The implementation "recipes" found in the latter are exceedingly useful during implementation. A fairly thorough exposition of how to perform reconstructions from indeterminate length video sequences is provided in [7]. A general review of autocalibration is presented in [3].

The process starts by applying the Harris corner detection to each of the images in the sequence. In total 1000 corners are extracted per image, each with a sufficiently high strength, and none of which are closer than 7 pixels to each other.

The corners from each image are then used to find point correspondences between each view and its neighbouring views, both forward and backward in the sequence. Block matching is used with a normalised correlation distance measure and blocks of size 15×15. With a minimum match value of 0.8 it was found that around 100 matches were found for each of the images in the sequence.

The fundamental matrix linking adjacent views is estimated using RANSAC on the correspondences. The images are subsequently rectified, and a guided matching stage performed. This increases the number of good correspondences from 100 to around 500. The minimum match value used here is the same as for the initial matching stage.

Since for a single set of corners the matching was performed both forwards and backwards, we effectively have correspondences over tracks of length three spanning triplets of images. Robust estimates of the trifocal tensor linking all sets of three adjacent views are then made, using RANSAC with an inlier distance of 1.25 pixels. It was found that about 300 matches per triplet survive this operation.

The trifocal tensor provides a strong constraint, and it is unlikely that any of the surviving matches will be incorrect. The triples therefore provide excellent initial values

for a complete reconstruction process.

At this stage a number of options can be exercised — each triplet can be used independently to provide a reconstruction of the points in the scene, but no uniformly best method appears to exist for how to combine these separate reconstructions. In this work a simple method of extending by resectioning was adopted. Since adjacent triples overlap by two views, a reconstruction from the first triple can be extended to the next simply by estimating the final camera of the new triple in the projective coordinate system of the first. That is, for a new triple the two known cameras can be used to estimate the world locations of the next set of points, from which a world-to-image transformation can be obtained corresponding to the new camera. This process of resectioning can be continued until the entire sequence has been reconstructed.

A problem with this approach is that the reconstruction is subject to drift, as there is no mechanism to prevent a gradual accumulation of errors as views are added. The effect of this drift would decrease if points were tracked for longer than just over image triples, but this would require the development of an affine (or projective) invariant match procedure. Further details relating to drift in reconstructions can be found in [1].

A simple method to reduce this drift makes use of the fact that the sequence is closed, with the first image being the natural neighbour to the last. It is easy to include this information into a bundle adjustment stage which uses the reconstruction described previously as an initial estimate of the solution. With 35 views containing around 12000 points in total this is a large problem, and consumes a large amount of computer memory. Since the observation matrix is banded and diagonal-dominant some reduction in computation could be achieved if desired. Alternatively we could just use a subset of the available points — there are far more than are needed for an accurate reconstruction of structure and motion in any case.

The final outcome of the procedure described is a projective reconstruction of points on the surface of the object, as well as locations of the cameras used to generate the images. This reconstruction differs from a Euclidean one by an unknown homography. To upgrade the reconstruction to metric, we need some information about the effective camera parameters used during the image capture. Since for the scanning electron microscope we have no means of explicitly calibrating either the intrinsic or extrinsic camera parameters, autocalibration at this point is essential.

The microscope settings were not changed while the object was rotated in the view, so an assumption of a fixed camera is appropriate. Also, it being a precise piece of equipment there is every reason to believe that the pixel skew can be assumed to be zero, and the pixels square. It is important that this latter assumption be made — turntable motion represents a critical motion sequence (CMS) in multiple view geometry, and without constraints a projective ambiguity arises in the component of the reconstruction in the direction of the screw axis [4, p.492].

One final assumption that is made is that the SEM can be represented as an affine imaging system, so the last row of each camera matrix is [0 0 0 1]. This assumption was made explicit when it was found that a full projective autocalibration produced unstable results when applied to the problem. It is believed that this instability is caused by near singularities in the projective autocalibration procedure when the actual camera geometry tends to be affine, although no attempt was made to validate this claim. Although not an assumption that we wanted to make at the outset, it had been observed throughout this project that the SEM imaging system did indeed seem to be affine.

An affine camera can be decomposed as

$$P_A = \begin{pmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{R}} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix},$$

where the leading matrix in this decomposition contains the intrinsic camera parameters. This decomposition forms the basis of an autocalibration procedure proposed by Quan [8], which was used in this work. In short, an upper triangular matrix $\mathbf{Z}$ is found which, when applied to all the cameras, yields a decomposition that satisfies the constraints on the intrinsic parameters. The elements of $\mathbf{Z}$ are found by a nonlinear least squares procedure, initialised by the identity matrix.

4. Results

It is difficult to present a reconstruction of a point cloud in print — it is useful to be able to navigate around it (with for example a VRML viewer) to convince oneself that the reconstruction is accurate. Nonetheless, Figure 3 shows a reconstruction of the structure and motion of the aluminium block sequence *before* making use of the assumption that the sequence is closed. The lines in the figure indicate the principal rays of the estimated cameras. Since these cameras are assumed affine they effectively lie on the plane at infinity. Close inspection reveals that the reconstruction is not sharp, and that points corresponding to the same image features in different triplets appear as distinct and different. The RMS reprojection error for this reconstruction is 0.3204 pixels.

If we make use of the fact that the sequence is closed, with the first image following the last, then visually better results are obtained. Figure 4 shows reconstructions from roughly equivalent viewpoints both before and after including this assumption. It is clear that fine structures are reconstructed more accurately and with less spread in the second case. The RMS reprojection error for the closed sequence reconstruction is 0.3252 pixels.

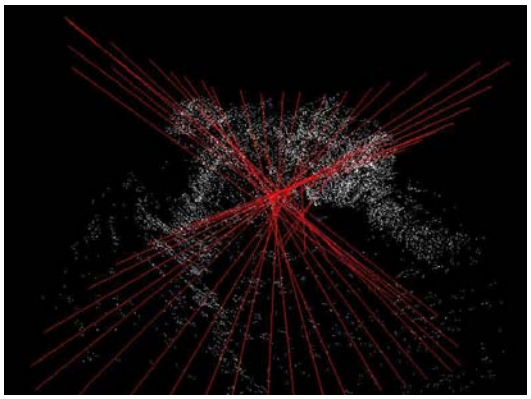
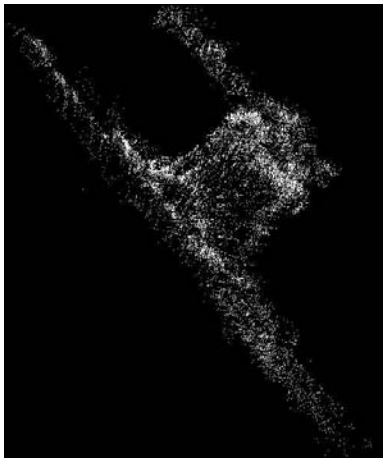


Figure 3: Reconstruction of points and camera views without the assumption that the sequence is closed.

(a) Without closed sequence assumption.



(b) With closed sequence assumption.

Figure 4: Effect on the reconstruction of using the assumption that the sequence is closed.

5. Conclusion

The research presented in this paper is at an early stage, and the number of improvements that could be made are almost too many to number. The most immediate improvement would be obtained by tracking the points over longer durations. This would probably necessitate the development of an affine (or projective) invariant feature matching procedure.

The assumption of the SEM being an affine imaging system appears to be valid and accurate. In retrospect this may have been a reasonable assumption to make at the outset: from a geometrical point of view the field-of-view of the images is around 1mm, while the working distance is 22mm. This corresponds a deviation from parallel projection by of the order of only about 1°. At higher magnifications the deviation may be expected to reduce even further.

6. References

[1] K. Cornelis, F. Verbiest, and L. Van Gool. Drift detection and removal for sequential structure from motion algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1249–1259, October 2004.

[2] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[3] A. Fusiello. Uncalibrated Euclidean reconstruction: a review. *Image and Vision Computing*, 18:555–563, 2000.

[4] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition, 2003.

[5] F. C. Nicolls, G. de Jager, and B. T. Sewell. Use of a general imaging model to achieve predictive autofocus in the scanning electron microscope. *Ultramicroscopy*, 69:25–37, August 1997.

[6] Frederick Nicolls. The development of a predictive autofocus algorithm using a general image formation model. Master's thesis, University of Cape Town, Rondebosch 7700, South Africa, February 1996. http://www.dip.ee.uct.ac.za/∼nicolls.

[7] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, September 2004.

[8] Long Quan. Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision*, 19(1):93–105, May 1996.

7. Acknowledgements

The author wishes to thank Trevor Sewell from the UCT Electron Microscopy Unit for capturing the SEM dataset used in this work.

A Method for Efficiently Re-estimating Camera Distortion Parameters

Ruby van Rooyen and Neil Muller

iThemba LABS, P. O. Box 722, Somerset West, 7129, South Africa

Abstract

At iThemba LABS, we use stereo vision techniques to accurately position the a patient for proton therapy. We chose to use conventional zoom lenses due to cost reasons. However these lenses have a high distortion factor. Also, due to maintenance work and other activities, we cannot assume that the lenses will not be disturbed between sessions. Thus we need to have a simple and efficient manner to recalculate the distortion parameters of the lenses before each treatment session.

1. Introduction

1.1. Problem Description

Proton radiotherapy is a useful treatment method for a number of lesions. The dose distribution properties of the proton beam allow for high doses to be delivered to the target volume while keeping the dose to the surrounding tissue to a minimum. Due to the high cost associated with this treatment, it is often reserved for lesions that are difficult to treat with conventional radiotherapy techniques, especially lesions close to critical structures. For more information see for example [1].

iThemba LABS has been involved with proton therapy for over ten years. Due to cost restrictions, iThemba LABS uses a fixed beam-line to deliver the proton dose, and uses a robotic manipulator to position the patient. The position of the patient during setup and treatment is monitored by a number of cameras and stereo techniques are used to calculate the patient's position at any time. A critical issue is the high positioning accuracy required. For further discussion on the system and some of the previous work on the vision aspects see [2], [3] and [4].

Amongst other things, for the degree of accuracy required, we need to have an accurate model of the distortion due to the lenses. Since the lenses used are conventional zoom lenses, distortion is a major factor in the system.

Furthermore, although we can accurately measure distortion before the cameras are mounted in the treatment room, the close proximity of the cameras to the the working area means that we cannot assume that the camera parameters do not change over time. Thus a number of additional checks are needed to ensure that the distortion model is correct and to update this model if need be. Since these checks will be done by the radiographers supervising the treatment, they need to be both simple and reasonably fast. A low degree of user involvement in this check is also desirable.

2. Obtaining initial distortion model

2.1. Distortion Pattern

In the design of an automated distortion correction technique, the main objectives of feature detection is accuracy, efficiency and robustness [5]. This can be achieved by keeping the complexity of the feature detection method as low as possible. Thus, we use circular features as the perspective projection of a circle is always a circle or an ellipse. Sub-pixel accuracy of the feature location can be achieved real-time and the complexity of the method is linear with the number of pixels [2].

Distortion is most clearly seen as the curvature of lines in a grid pattern. Consequently, we have designed a similar pattern for the computation of the distortion parameters (shown in figure 1). The distortion pattern is constructed by uniformly spacing 8 mm diameter circular targets in a rectangular grid. A group of three torus shaped targets are placed at various row positions in the upper left corner of the pattern. This allow for fast calculation of the orientation vector and the repetition of the pattern allows for the use of various zoom settings. To ensure high accuracy, the colours of the circles are selected such that a high contrast between the circles and the background is obtained [5].

2.2. Distortion Correction

2.2.1. Distortion Model

To get the corrected pixel coordinates of a feature that is to be used for 3D reconstruction, a lens model is used to compute the distortion parameters required [6]. This model is given by

$$(x_u, y_u) = (x_d, y_d) + \delta(\kappa, P)\,,$$

where (x_u, y_u) are the undistorted image coordinates, (x_d, y_d) are the distorted input coordinates and (κ, P) are the respective radial and tangential distortion parameters.

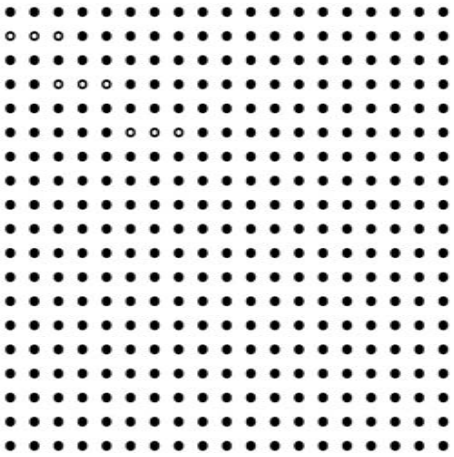


Figure 1: Distortion Correction Pattern.

The distortion factors given by:

$$\delta_x(\kappa, P)i = \overline{x}_d \sum_{l=1}^{\infty} \kappa_l \overline{r}_d^l + \left[P_1(\overline{r}_d + 2\overline{x}_d^2) + 2P_2\overline{x}_d\overline{y}_d\right] \left[1 + \sum_{l=1}^{\infty} P_{l+2}\overline{r}_d^l\right]$$

and

$$\delta_y(\kappa, P) = \overline{y}_d \sum_{l=1}^{\infty} \kappa_l \overline{r}_d^l + \left[2P_1\overline{x}_d\overline{y}_d + P_2(\overline{r}_d + 2\overline{y}_d^2)\right] \left[1 + \sum_{l=1}^{\infty} P_{l+2}\overline{r}_d^l\right]$$

where $\overline{x}_d = x_d - x_p$ and $\overline{y}_d = y_d - y_p$ with (x_p, y_p) the principle point and $\overline{r}_d^2 = \overline{x}_d^2 + \overline{y}_d^2$.

Although most distortion models ignore tangential distortion, it has been shown that, by allowing the centre of radial distortion (c_{xr}, c_{yr}) to be different from the principle point (x_p, y_p) of the lens, a good approximation for the decentring distortion is obtained [7]. Thus the distortion model becomes:

$$(x_u, y_u) = (x_d, y_d) + (\overline{x}_d, \overline{y}_d)\delta(\kappa) \text{ and}$$

$$\overline{r}_d^2 = \overline{x}_d^2 + \overline{y}_d^2 = (x_d - c_{xr})^2 + (y_d - cyr)^2\,,$$

$$\delta(\kappa) = \kappa_1\overline{r}_d^2 + \kappa_2\overline{r}_d^4 + \ldots\,,$$

where κ_1 and κ_2 are the first and second radial distortion parameters.

Numerical stability requires that the second order term κ_1 be found first [8], even if the higher order terms are required.

2.2.2. Distortion Algorithm

The distortion correction model will be used as a filter rather than being applied to the whole of the image for correction.

The off-line process used to calculate the distortion model parameters and the distortion centre is similar to the algorithms suggested in [9].

Distortion Algorithm:

1. Extract the distorted dot pattern [2].
2. Calculate the dot centroids in the distorted image. [2]
3. Find the point correspondence between the distortion centroid and the centroids in the corrected image.
4. Estimate an initial distortion centre and corrected image centre [9].
5. Calculate the expansion coefficients and image centre.
6. Adjust the distortion parameters to reduce the error.
7. Repeat from 5 until convergence.

An initial approximation of the distortion centre is estimated by interpolating the point where the curvature is zero. The expansion coefficients are calculated by finding the polynomial transform that orients the centroids of the grid dots to a straight line along the x and y direction:

$$y_{ij} = b_1^x x_{ij} + b_{0i}^x \,; 1 \le j \le K_i\,,\ 1 \le i \le L_x\,,$$

where (x_{ij}, y_{ij}) is the centroids of the dots on the corrected image, L_x is the number of rows in the grid and K_i is the number of dots in row i. This calculation is similar for the y direction.

Since the dot pattern conforms to a strictly rectangular grid, we assume that all the grid lines (along a particular axis) have the same slope. Thus b_1^x represents the horizontal slope for all grid lines in the x direction of the corrected image, and b_{0i}^x is the different intercept value of each grid line [9].

For the mapping from radial distortion space to corrected space, a non-linear least squares optimisation algorithm is used to fit the distorted grid lines to straight lines [9],[7].

After the model parameters are computed they must be used for real-time distortion correction. We construct a look-up tabel to map distorted input coordinate to the corrected output coordinates. This look-up table is calculated from an inverse mapping of the distortion correction model, and gives coordinates corresponding to pixel values in the distorted image [9].

3. Updating the Model

3.1. Vault environment

We use the same texture printed on a portable planar object. This printed example will be held up to each camera in turn. Obviously the translation and rotation of this object relative to the camera will not be consistent between the cameras and across sessions. We assume that the distortion parameters observed in the treatment room are close to those calculated outside the treatment room. As we will check the model frequently, we can alternatively use the previous session's calculations as the starting information.

We can easily extract straight lines from the resulting image, using the grid arrangement of the markers. Using these straight lines, we can easily test whether the distortion model is still valid.

Given that we only sparsely sample the lines, and that we are no longer assured that we have good coverage of the camera view, we can encounter problems using the purely line-based approaches to modelling the distortion correction. Noise issues due to the treatment room environment and damage to the cameras can also impact on the estimation if we restrict ourselves to purely straight lines. However, we can use the entire object to calculate the updated distortion parameters, using the method described in [10] and [11].

To do this, we first estimate the transformation parameters and then optimise over both the transformation and distortion parameters to minimise the error between the predicted image and the observed image.

3.2. Estimate of Transformation parameters

We know that the observed image is distorted. However, since we model the distortion as radial, and we know the approximate centre of distortion, we can safely assume that the distortion near this point is small. This allows us to use linear models to estimate the transformation parameters.

For our purposes, we assume a pinhole camera model. The transformation in homogeneous coordinates is given by

$$\lambda \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} f & s & u_x \\ 0 & \frac{f}{d} & u_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & T_x \\ r_4 & r_5 & r_6 & T_y \\ r_7 & r_8 & r_9 & T_z \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 0 \\ 1 \end{bmatrix}$$

where $\begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$ is a rotation matrix, $\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$ is the translation vector and $\begin{bmatrix} x_p \\ y_p \\ 0 \\ 1 \end{bmatrix}$ are the points on the distortion pattern. It is known that determining the full transformation from point correspondences on a plane is under-determined (although it was shown in [12] that it is possible given suitable additional geometric information). However, we only need the 2D projective transformation given by

$$\lambda \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = M \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}.$$

(The final matrix entry is set to 1 to ensure uniqueness). Given four point correspondences, this can be easily solved (see [13] for example).

In our case, since we are dealing with a regular grid, we do not need image registration. Given the image coordinates four points forming a single square of the grid, $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$ and $\begin{bmatrix} x_4 \\ y_4 \end{bmatrix}$ we can solve for the transformation using the following system

$$\begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \lambda_3 x_3 & \lambda_4 x_4 \\ \lambda_1 y & \lambda_2 y_2 & \lambda_3 y_3 & \lambda_4 y_4 \\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{bmatrix} = M \begin{bmatrix} 0 & d & 0 & d \\ 0 & 0 & d & d \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Since all other points on the grid can be expressed as $\begin{bmatrix} \alpha d \\ \beta d \end{bmatrix}$ where α, β are integers, this transformation is accurate up to an unknown translation (It is also accurate only up to a rotation of $90°$, but since the grid is symmetrical on both x and y axes, this is not important). Since the distortion correction is concerned with the matching of straight lines, this unknown translation does not effect any of the resulting calculations and can be ignored.

3.3. Optimisation

This estimate for the transformation is sensitive to noise (which can be compensated for by using multiple points) and will include some error since we neglect the effect of distortion. Thus we need to optimise both this transformation and the distortion parameters to ensure a good fit. Given a set of observed points $\mathbf{o}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i \in [1, N]$ on the image, and the corresponding grid points $\begin{bmatrix} \alpha_i d \\ \beta_i d \end{bmatrix}$ where the α's and β's can easily be calculated by either counting points or calculating distances from the points used to obtain the transformation, we define the error function

$$E(\kappa, P, M) = \sum_{i=1}^{N} \left(\mathbf{o}_i - \left(M \begin{bmatrix} \alpha_i d \\ \beta_i d \end{bmatrix} + \delta(\kappa, P) \right) \right)^2$$

where $\delta(\kappa, P)$ is the distortion function defined earlier. The parameters of E are the distortion parameters and the projective transformation.

We assume that the distortion parameters, while not static, vary slowly with time. Thus we can use the already calculated distortion model as a starting point for

optimisation. The estimated transformation parameters are used as a starting point for the true transformation parameters. Thus we can be reasonably confidant that optimisation will start from a position close to the global minimum. Conventional optimisation techniques can be used to obtain the final parameters with comparative ease. The usual problems associated with this type of high-dimensional optimisation are eliminated by ensuring that the initial estimate is close to the correct starting point.

We optimise over this sparse set of points and then use this result as a starting point for the optimisation over the full image. Since the initial optimisation is over a limited number of grid points, the overall complexity is fairly small and the calculation can be done quickly. The final optimisation should start from very close to the correct solution and converge quickly.

4. Conclusions

We show how we can easily update the distortion model for our cameras using a simple method. This allows us to detect and compensate for changes to the distortion parameters that may occur over time. As the distortion correction calculations can significantly impact on the overall accuracy of the system, the ability to easily correct these parameters if needed is a necessary component of the system design.

5. References

[1] S. Webb, *The physics of three-dimensional radiotherapy: Confromal radiotherapy, radiosurgery and treatment planning*, Insitute of Physics Publishing, Bristol and Philadelphia, 1993.

[2] R. van Rooyen, "Fast, robust detection of circular retro-reflective targets," in *Proceeding of the Fourteenth Annual Symposium of the Pattern Recognition Association of South Africa*, Nov. 2003, pp. 21–26.

[3] Evan de Kock, Brian O'Kennedy and Neil Muller, "Calibrating a stereo rig and ct scanner with a single calibration object," in *Vision, Modeling and Visualization 2002*, G. Greiner, H. Niemann, T. Ertl, B. Girod and H.-P. Seidel, Eds., 2002, ISBN 3-89838-034-3.

[4] Evan de Kock, Neil Muller, Denys Maartens, Jan van der Merwe, Deon Muller, Ruby van Rooyen, Andre van der Merwe, Jan Eksteen, Neil von Hoesslin, Dirk Wagener and Jan Hough, "Integrating an industrial robot and multi-camera computer vision systems into a patient positioning system for high-precision radiotherapy," in *35th International Symposium on Robotics*, Mar. 2004.

[5] Jun-Sik Kim, Ho-Won Kim and In-So Kweon, "A camera calibration method using concentric circles for vision applications.," in *The 5th Asian Conference on Computer Vision*, Jan. 2002.

[6] Sing Bing Kang, "Semiautomatic methods for recovering radial distortion parameters from a single image," Tech. Rep. CRL 97/3, Digital Equipment Corporation, Cambridge Research Lab, May 1997.

[7] G. Stein, "Lens disortion calibration point correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 602–608.

[8] Ben Tordoff and David W. Murray, "Violating rotating camera geometry: The effect of radial distortion on self-calibration," in *15th International Conference on Pattern Recognition*, Anil K. Jain, Svetha Venkatesh and Brian C. Lovell, Eds. IEEE, 2000, pp. 1423–1427, IEEE Computer Society.

[9] Chao Zhang, James P. Helferty, Geoffrey McLennan and William E. Higgins, "Nonlinear distortion correction in endoscopic video images," in *2000 IEEE International Conference on Image Processing*, Sept. 2000, pp. 439–442.

[10] T. Tamaki, T. Yamamura and N. Ohnishi, "Correcting distortion of image by image registration," in *The 5th Asian Conference on Computer Vision*, Jan. 2002.

[11] T. Tamaki, T. Yamamura and N. Ohnishi, "Unified approach to image distortion," pp. 584–587, 2002.

[12] V. Fremont and R. Chellali, "Direct camera calibration using two concentric circles form a single view.," in *International Conference on Artifical Reality and Telexistence*, Dec. 2002.

[13] R. Szeliski, "Visual mosaics for virtual enviroments," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 22–30, Mar. 1996.

Visual Hulls from Single Uncalibrated Snapshots Using Two Planar Mirrors

Keith Forbes[1] Anthon Voigt[2] Ndimi Bodika[2]

[1]Digital Image Processing Group
Department of Electrical Engineering
University of Cape Town
Private Bag, Rondebosch, 7701
kforbes@dip.ee.uct.ac.za

[2]Automation and Informatics Group
GTS Technology
De Beers Group Services
P. O. Box 82851, Southdale, 2135
{Anthon.Voigt, Ndimi.Bodika}@debeersgroup.com

Abstract

Two mirrors are used to create five views of an object: a view onto the object, two reflections and two reflections of reflections. The five views are captured in a single snapshot. Epipolar geometry of the object's five silhouettes is determined directly from the image without knowing the poses of the camera or the mirrors. The epipolar geometry provides constraints on the pose of each silhouette, allowing the pose of each silhouette to be computed in a common reference frame using only the silhouette outlines. Once the pose associated with each silhouette has been computed, a five-view visual hull of the object can be computed from the five silhouettes. By capturing several images of a rigid object in different poses, sets of five silhouettes can be combined into a single silhouette set in which the pose of each silhouette is known in a common reference frame. This allows visual hulls of an arbitrary number of views to be computed if more than one image is used. The method is applied to an ornamental cat, and experimental results are shown.

1 Introduction

The *visual hull* provides a relatively simple means for creating an approximate 3D model of an object: it is the largest object that is consistent with a set of silhouettes of the actual object whose poses (positions and orientations) are known in a common reference frame. Silhouettes of an object are typically captured by multiple calibrated cameras positioned at different viewpoints that are well-spaced about the viewing hemisphere. The pose associated with each camera (and its silhouette) is usually pre-computed using a calibration object [2].

If the poses of the silhouettes are not known, then it is not, in general, possible to compute the poses of the observed silhouettes from the silhouettes themselves [1]. However, if some information about the poses is known, then in some cases it is possible to compute the poses of the silhouettes using only the silhouettes as input. This is done by computing a set of poses that is *consistent* with the silhouettes: if one silhouette implies that a volume of space is empty, another silhouette in the set cannot indicate that some part of the volume contains the object.

The computer vision literature holds several examples of computing silhouette poses from silhouettes under certain constraints. For instance, Mendonça et al. [5] use a turntable to create silhouettes with circular motion. The poses associated with each silhouette are computed from the silhouettes using the constraint of circular motion. Okatani and Deguchi [6] use a camera with a gyro sensor so that the orientation component of each silhouette pose is known. The translational component is then computed from the silhouettes.

In this work, we use two mirrors to create a scene consisting of a real object and four virtual objects (see Figure 1). Two of the virtual objects are reflections of the real object, and the other two virtual objects are reflections of these reflections. Five views of the object are captured in a single snapshot of the scene. Segmenting the image yields five silhouette views of the object. Without knowing the poses of either the camera or the mirrors in advance, the poses associated with each of the five silhouettes are computed from the silhouettes. This allows us to to create a five-view visual hull model of the object from a single snapshot. If the object is rigid, then we can obtain several five-view silhouette sets of the object in different poses. This can be done by moving the object, the camera, or the mirrors. We then use our previously described method [3] to combine the silhouette sets into a single silhouette set in which each silhouette's pose is known in a common reference frame. This allows us to create a visual hull model from the single set that is a better approximation to the 3D shape of the object than visual hulls created from any of the original five-view sets.

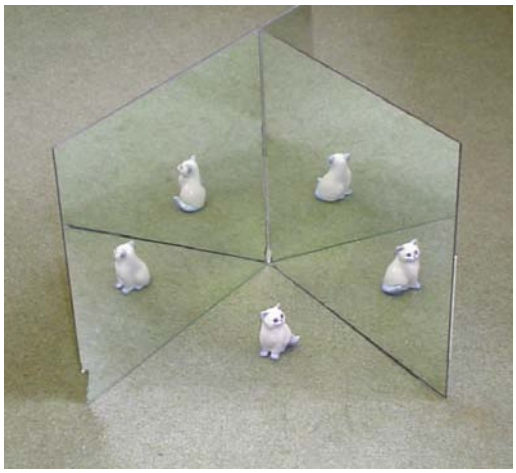


Figure 1: Two mirrors are used so that five views of the object can be seen.

Unlike other methods for computing visual hulls, our method does not require specialised equipment (no turntables, calibration objects, or synchronised cameras are required), and does not require special care to be taken in positioning any of the apparatus, since silhouette poses are computed from the image. We use two 300 mm × 300 mm × 3 mm off-the-shelf, bathroom-style mirrors and a four megapixel consumer-grade digital camera. (The increase in the number of pixels of digital cameras over the last few years has made the proposed method possible: the resolution of each of the five silhouettes within a single image is reasonably high.) The mirrors are positioned so that two of the sides are next to each other, and the angle between the mirrors is approximately 72° (as in Figure 1). (In principle, one could create more than five views by having a smaller angle between the mirrors, but in this work we restrict ourselves to the five-view case.) The object is placed in front of the mirrors so that five views of it can be seen. To aid segmentation, a background is chosen that contrasts with the colour of the object.

The computation of the silhouette poses is based on the observation that the epipolar geometry* relating two silhouette views of an object and its reflection in a single image can be computed without knowing the relative poses of the silhouettes, and without knowing point correspondences. We use an orthographic projection model for each of the silhouettes, since the variation in depth associated with each silhouette is small with respect to the distance to the camera. The pose parameters are divided into several components which are solved one by one: viewing direction, silhouette roll, and translation.

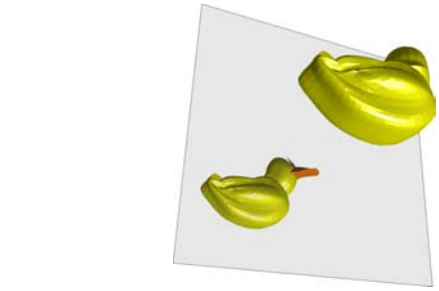
The remainder of this paper is organised as follows. Section 2 describes how the epipolar geometry relating silhouette views of an object and its reflection can be computed directly from the silhouette outlines. This observation is the basis for our method of determining silhouette poses for a double mirror setup. The geometry of the double mirror setup is described in Section 3, and terminology and view relationships used in the remainder of the paper are introduced. Section 4 shows how silhouettes can be scaled to create a good approximation to the silhouette image that would be observed by an orthographic camera. Section 5 explains how the poses of each of the five silhouettes visible in an image of the scene can be determined from the silhouette outlines. Experimental results are presented in Section 6, and the paper is summarised in Section 7.

2 Silhouette Epipolar Geometry with One Mirror

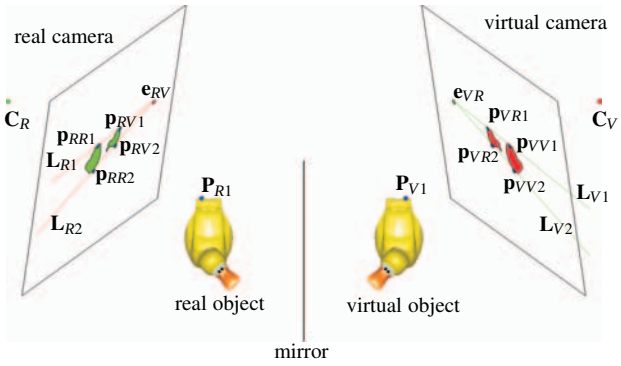
In this section, we show that the epipolar geometry relating the silhouette views of an object and its reflection can be determined by considering only the outline of the object and the outline of its reflection in a single image; neither the pose of the camera with respect to the mirror, nor point correspondences are required. This discussion uses perspective projections, however we will assume orthographic projections (a special case in which the cameras are at infinity) for the pose estimation described in Section 5.

Figure 2(a) shows a camera's view of a duck and its reflection.

*Definitions of terms relating to the geometry of multiple silhouette views are omitted in this paper because of space constraints. Please refer to our previous paper for these definitions [3].



(a)



(b)

Figure 2: Reflection of a duck in a mirror: (a) shows the image seen by the real camera, (b) shows the silhouette views seen by the real camera and by the virtual camera that is the reflection of the real camera.

Figure 2(b) shows the camera and the observed silhouette views of the real object and the virtual object (the reflection of the real object). The figure also shows a virtual camera (the reflection of the real camera) and the silhouettes that such a camera would observe. Note that the image observed by the virtual camera is simply a reflection of the image observed by the real camera. Although the virtual camera does not exist, we can determine the images that it would observe from the real camera's image: the virtual camera's view of the real object is simply the mirror image of the real camera's view of the virtual object. We can therefore consider the two available silhouettes (captured by the real camera) to be two views of the real object: one from the real camera's viewpoint, and one from the virtual camera's viewpoint.

Consider a plane π_1, perpendicular to the mirror, passing through the camera centre $\mathbf{C}_R$ and tangent to the top of the real object at $\mathbf{P}_{R1}$. Since π_1 is perpendicular to the mirror, it will also contain the reflections of $\mathbf{C}_R$ and $\mathbf{P}_{R1}$: $\mathbf{C}_V$ and $\mathbf{P}_{V1}$. Since the plane contains both camera centres, it will also contain both epipoles $\mathbf{e}_{RV}$ and $\mathbf{e}_{VR}$. The projections of $\mathbf{P}_{R1}$ and $\mathbf{P}_{V1}$ will lie on the silhouette outlines for both cameras, since π_1 is tangent to both objects. The real camera's projections of $\mathbf{P}_{R1}$ and $\mathbf{P}_{V1}$ are the points $\mathbf{p}_{RR1}$ and $\mathbf{p}_{RV1}$. Since $\mathbf{p}_{RR1}$, $\mathbf{p}_{RV1}$ and the epipole $\mathbf{e}_{RV}$ lie in both the image plane and π_1, they are collinear.

Now consider a second plane π_2 that is perpendicular to the

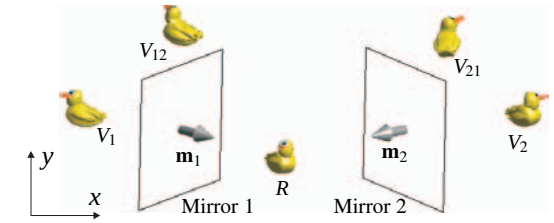
mirror, passes through the camera centre $\mathbf{C}_R$, but is tangent to the *bottom* of the real object at $\mathbf{P}_{R2}$. ($\mathbf{P}_{R2}$ is not shown in Figure 2(b) since it is on a hidden surface of the duck.) The plane π_2 can be used to show that $\mathbf{p}_{RR2}$, $\mathbf{p}_{RV2}$ and $\mathbf{e}_{RV}$ are collinear.

The points $\mathbf{p}_{RR1}$, $\mathbf{p}_{RV1}$ can be determined in an image by simply finding the line $\mathbf{L}_{R1}$ that is tangent to both silhouettes in the image. The epipole need not be known. The line $\mathbf{L}_{R2}$ is the tangent line on the other side of the silhouettes. The intersection of $\mathbf{L}_{R1}$ and $\mathbf{L}_{R2}$ is the epipole.

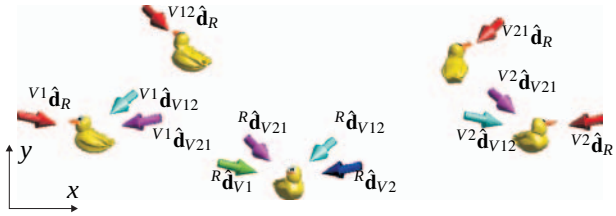
In the case of an orthographic projection, the epipoles lie at infinity. In this case the slope of the tangent lines is a projection of the camera's direction.

3 Double Mirror Scene Geometry

We use a scene with two mirrors for two reasons: (1) to obtain multiple silhouette views of an object, and (2) to obtain multiple silhouette tangent lines that provide sufficient information about the pose of the silhouettes that the poses of the silhouettes can be computed. Figure 3(a) shows the camera's view of a scene with two mirrors. The figure introduces some of the terminology that will be used later in this paper to determine the silhouette poses. Four virtual objects are also shown in the figure: V_1 is the reflection of R in Mirror 1; V_2 is the reflection of R in Mirror 2; V_{12} is the reflection of V_2 in Mirror 1; and V_{21} is the reflection of V_1 in Mirror 2. The normal vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ for each of the two mirrors are shown.



(a)



(b)

Figure 3: A double mirror scene in which five objects can be seen: (a) shows the positions of the two mirrors, and (b) shows various viewing direction vectors that are used in this work.

In Section 5, each of the five objects will be used to define a reference frame, and the mirror normals will be used to transform directions between reference frames. To determine the reflected direction $\mathbf{d}_r$ from a unit direction $\hat{\mathbf{d}}$ using unit mirror normal $\hat{\mathbf{m}}$ the following equation is used.

$$\mathbf{d}_r = \mathbf{d} - 2\hat{\mathbf{m}}(\hat{\mathbf{d}} \cdot \hat{\mathbf{m}}) \tag{1}$$

Figure 3(b) shows viewing directions in different reference frames. In each object's reference frame there are five views corresponding to the five available silhouettes (the silhouettes segmented from the image captured by the real camera). There are therefore 25 viewing directions that correspond to available silhouette views. The figure shows only the viewing directions that are used for computing the silhouette poses in Section 5. (The five directions of the form $[0,0,1]^T$ are omitted since they point directly into the page.) Note that the silhouettes observed in the reference frames of V_1 and V_2 are mirror images of the available silhouettes. Since we will be dealing with orthographic projections, the viewing direction $-\mathbf{d}$ will observe the non-mirrored silhouettes observed by viewing direction $\mathbf{d}$. We can therefore select viewing directions so that we only deal with non-mirrored silhouettes.

We use the superscript notation used by Forsyth and Ponce [4] to indicate the reference frame associated with a vector so that ${}^A d_B$ indicates the viewing direction of camera B onto object A (i.e. in the reference frame of A). The camera view that captures the observed silhouette of an object defines the reference frame so that ${}^A d_A = [0,0,1]^T$ for all A. The reference frame x- and y-axes are aligned with the image axes.

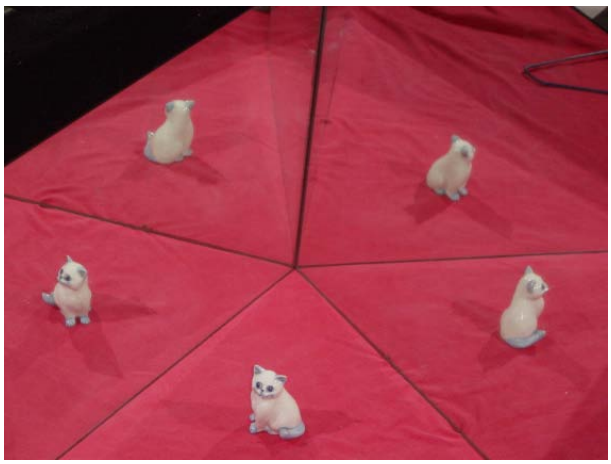
4 Approximating Orthographic Views

Figure 4(a) shows an example of an image of a scene. The segmented image consisting of five silhouette outlines is shown in Figure 4(b). Figure 4(c) shows an approximation to an orthographic projection of the scene that is derived from the perspective projection shown in Figure 4(b). Epipolar tangency lines are shown in Figures 4(b) and (c).

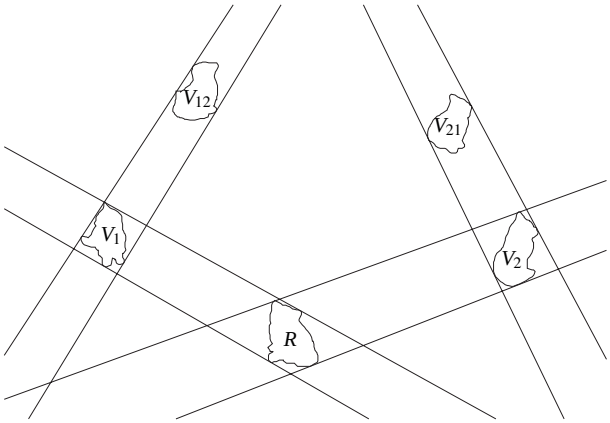
In order to compute the pose associated with each silhouette, we treat the silhouettes as if they were produced by an orthographic projection. This is done for the purpose of simplicity; the orthographic imaging model yields tractable equations without any significant affect on accuracy.

Consider moving the camera backwards from the scene and zooming so that the size of the silhouette of the real object R stays the same. As the camera is moved backwards and zoomed, the silhouettes of the virtual objects will become larger. The change in shape of the silhouettes is negligible, since the depth variation of the objects is small compared with the distances to the initial camera position. When the camera is moved back to infinity, we have orthographic imaging conditions. A very good approximation to the silhouettes that would be seen by such an orthographic camera is created by scaling the silhouettes of the virtual objects in the image captured by the real camera.

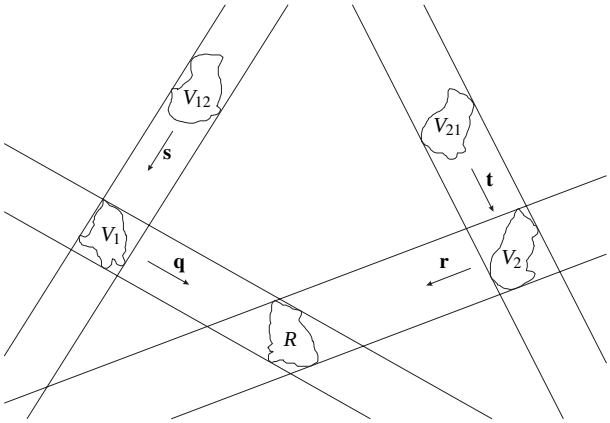
The results of scaling to create an orthographic projection are shown in Figure 4(c). The epipolar tangent lines are adjusted so that they are parallel and tangent to the silhouette. The direction of the new epipolar lines is that of the sum of two unit vectors parallel to each of the original epipolar lines. The silhouette of the reflection is then scaled so that it is tangent to both new epipolar



(a)



(b)



(c)

Figure 4: Images of a scene: (a) shows the raw image, (b) shows the segmented image with silhouette outlines and epipolar tangency lines, and (c) shows the derived orthographic image that would be seen by an orthographic camera.

lines. This process is applied to the four silhouettes of the virtual objects.

The 2D vectors **q**, **r**, **s**, and **t** indicated in Figure 4(c), are parallel to the tangent lines. These 2D vectors are determined from the tangent lines which are in turn determined directly from the silhouette outlines.

Note that **q** is a projection of ${}^R\mathbf{d}_{V1}$ onto the *xy*-plane. Once **q** is known, only the *z*-component of ${}^R\mathbf{d}_{V1}$ is unknown. Similarly, **r** is the projection of ${}^R\mathbf{d}_{V2}$; **s** is the projection of ${}^{V1}\mathbf{d}_{V12}$; and **t** is the projection of ${}^{V2}\mathbf{d}_{V21}$.

5 Computing Silhouette Poses

To determine the silhouette poses we exploit the constraints imposed by the directions of **q**, **r**, **s**, and **t** which are measured directly from the silhouette outlines. This allows us to compute the viewing directions for all five views of the real object. Once the viewing directions are known, the remaining component of orientation, the silhouette roll, can be computed. The translational component can then be determined for a given silhouette by using two other silhouettes to enforce the epipolar tangency constraint.

5.1 Viewing Direction

Here, we aim to compute the five viewing directions in the reference frame of *R*. To determine the five viewing directions we set up equations that specify the directions of **s** and **t** in terms of the direction vectors ${}^R\mathbf{d}_{V1}$ and ${}^R\mathbf{d}_{V2}$. Since only the *z*-components of ${}^R\mathbf{d}_{V1}$ and ${}^R\mathbf{d}_{V2}$ are unknown (the *x*- and *y*-components are determined from **q** and **r**), we will have two equations in two unknowns. This will allow us to solve for the *z*-components, and then to solve for the remaining direction vectors and mirror normals whose values are given in terms of the elements of ${}^R\mathbf{d}_{V1}$ and ${}^R\mathbf{d}_{V2}$.

The viewing direction ${}^R\mathbf{d}_{V1}$ consists of three components:

$$ {}^R\mathbf{d}_{V1} = \begin{pmatrix} {}^Rd_{xV1} \\ {}^Rd_{yV1} \\ {}^Rd_{zV1} \end{pmatrix}. \tag{2} $$

Note that ${}^Rd_{xV1}$ and ${}^Rd_{yV1}$ are selected so that $({}^Rd_{xV1}, {}^Rd_{yV1})^T = \mathbf{q}$. The magnitude of **q** is unimportant. Unit magnitude may be chosen, for instance.

The mirror normal $\mathbf{m}_1$ is given by the equation

$$ \mathbf{m}_1 = 1/2({}^R\hat{\mathbf{d}}_{V1} - {}^R\hat{\mathbf{d}}_R), \tag{3} $$

where ${}^R\mathbf{d}_R = [0,0,1]^T$, and the hat symbol indicates unit norm so that $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$. Similar equations are formulated for ${}^R\mathbf{d}_{V2}$ and $\mathbf{m}_2$.

In order to determine an expression for **s**, ${}^{V1}\hat{\mathbf{d}}_{V12}$ must be determined in terms of the elements of ${}^R\mathbf{d}_{V1}$ and ${}^R\mathbf{d}_{V2}$. This is done by starting with the vector ${}^{V12}\hat{\mathbf{d}}_{V12} = [0,0,1]^T$ and reflecting it first with Mirror 1, then with Mirror 2 and then again with Mirror 1, so that it is in the appropriate reference frame. This can be done, since the mirror normals are known in terms of ${}^R\mathbf{d}_{V1}$ and ${}^R\mathbf{d}_{V2}$. The viewing direction ${}^{V2}\hat{\mathbf{d}}_{V12}$ is the reflection of ${}^{V12}\hat{\mathbf{d}}_{V12}$ in Mirror 1:

$$ {}^{V2}\hat{\mathbf{d}}_{V12} = {}^{V12}\hat{\mathbf{d}}_{V12} - 2\hat{\mathbf{m}}_1({}^{V12}\hat{\mathbf{d}}_{V12} \cdot \hat{\mathbf{m}}_1). \tag{4} $$

The viewing direction ${}^{R}\hat{\mathbf{d}}_{V12}$ is the reflection of ${}^{V2}\hat{\mathbf{d}}_{V12}$ in Mirror 2:

$$ {}^{R}\hat{\mathbf{d}}_{V12} = {}^{V2}\hat{\mathbf{d}}_{V12} - 2\hat{\mathbf{m}}_2({}^{V2}\hat{\mathbf{d}}_{V12} \cdot \hat{\mathbf{m}}_2), \quad (5) $$

The viewing direction ${}^{V1}\hat{\mathbf{d}}_{V12}$ is the reflection of ${}^{R}\hat{\mathbf{d}}_{V12}$ in Mirror 1:

$$ {}^{V1}\hat{\mathbf{d}}_{V12} = {}^{R}\hat{\mathbf{d}}_{V12} - 2\hat{\mathbf{m}}_1({}^{R}\hat{\mathbf{d}}_{V12} \cdot \hat{\mathbf{m}}_1). \quad (6) $$

Similar equations are derived for ${}^{V1}\hat{\mathbf{d}}_{V21}$, ${}^{R}\hat{\mathbf{d}}_{V21}$, and ${}^{V2}\hat{\mathbf{d}}_{V21}$ to create an expression for $\mathbf{t}$.

Equation (6) can be used to express ${}^{V1}\hat{\mathbf{d}}_{V12}$ in terms of the components of ${}^{R}\mathbf{d}_{V1}$ and ${}^{R}\mathbf{d}_{V2}$. The unknowns are ${}^{R}d_{zV1}$ and ${}^{R}d_{zV2}$: ${}^{R}d_{V1x}$, ${}^{R}d_{V1y}$, ${}^{R}d_{V2x}$, and ${}^{R}d_{V2y}$ are computed from the epipolar tangency lines. Similarly, ${}^{V2}\mathbf{d}_{V21}$ can be formulated in terms of ${}^{R}\mathbf{d}_{V1}$ and ${}^{R}\mathbf{d}_{V2}$. Since the ratios ${}^{V1}d_{yV12}/{}^{V1}d_{xV12}$ and ${}^{V2}d_{yV21}/{}^{V2}d_{xV21}$ can be formulated in terms of the components of ${}^{R}\mathbf{d}_{V1}$ and ${}^{R}\mathbf{d}_{V2}$, and can also be measured from the silhouettes, two equations can be set up to solve for the two unknowns ${}^{R}d_{zV1}$ and ${}^{R}d_{zV2}$. The Matlab Symbolic Toolbox failed to find a solution to these two equations. However, solutions were found for ${}^{R}d_{zV1}$ in terms of ${}^{R}d_{zV2}$, and for ${}^{R}d_{zV2}$ in terms of ${}^{R}d_{zV1}$. (The equations are too large to reproduce here.) This allows us to set up an equation in terms of ${}^{R}d_{zV1}$. A value for ${}^{R}d_{zV1}$ can be determined by uniformly sampling direction vectors with appropriate x- and y-components so that the unknown angle varies between 0° and 360°. A few iterations of the bisection method can be applied in the vicinity of a root to obtain a result that is accurate to machine precision. When more than one root exists, we choose the solution that results in a consistent set of silhouettes.

5.2 Silhouette Roll

The roll is specified using the up vector, which is the direction of the silhouette's y-axis in the common reference frame. The up vector is $[0,1,0]^T$ in the view's reference frame. Once the mirror normals are known, the up vectors in the reference frame of R can easily be computed by appropriately reflecting the up vectors, starting in each view's reference frame.

5.3 Translational Component

Once the orientation of the silhouettes is known, the translational component of pose can be computed using the epipolar tangency constraint. Since the orientation is known, the epipolar directions are known. The translational component must be selected so that the silhouette is tangent to the projections of the epipolar tangency lines of the other silhouettes. The translational component is overconstrained, since each other silhouette provides two constraints (corresponding to the two outer epipolar tangencies). To estimate the translational component, we start by determining the translational component for the silhouette of V_1, using R as a reference frame. The mean of the translational components indicated by each of the two outer epipolar tangencies is used. The translational components for the remaining silhouettes are computed one by one by selecting the translational component so that the silhouette is tangent to projections of outer epipolar tangencies from silhouettes of R and V_1.

6 Experimental Results

Six images of an ornamental cat were used to test the proposed method. The cat was placed in six different poses.

6.1 Five-View Visual Hulls

Figure 5 shows the six five-view visual hulls computed from the six images. Silhouette poses were computed using the proposed method. The visual hulls give a reasonable gross approximation to the 3D shape of the cat.

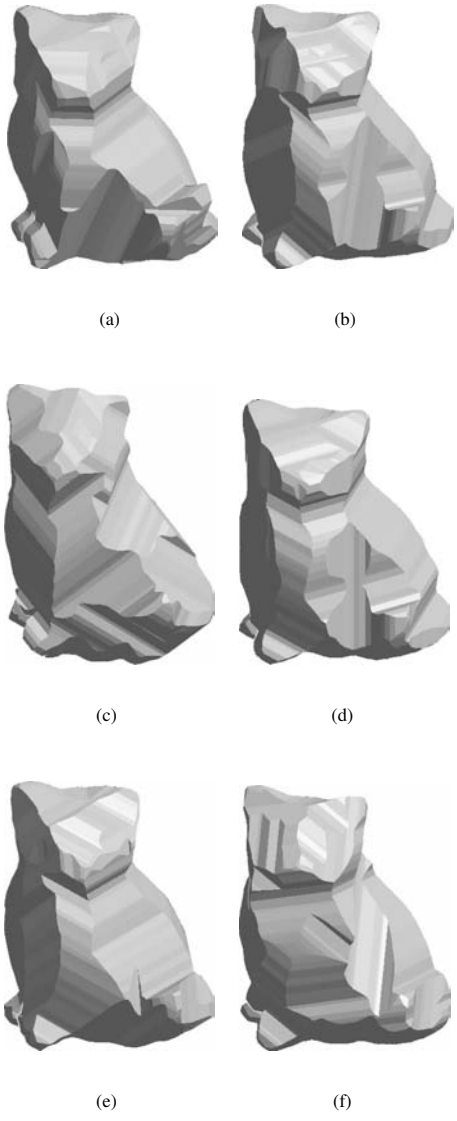


Figure 5: Six five-view visual hull models of an ornamental cat. The cat was placed in six different poses, but the visual hulls are shown aligned to aid comparison.

Figure 6 shows five silhouettes of the cat that were segmented from one of the images. Epipolar lines computed using the silhouette poses are shown in the figure. The epipolar lines are approxi-

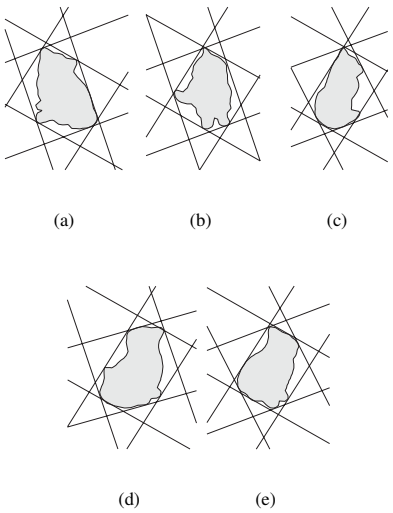


Figure 6: Five silhouettes showing epipolar tangency lines corresponding to the other four views in the set.

mately tangent to the silhouettes indicating that the silhouettes are consistent with the computed poses.

6.2 Visual Hulls from Merged Sets

The six silhouette sets were merged into a single silhouette set using our previously described method [3]. The silhouette sets were initially scaled so that the resultant visual hulls were of unit volume and the principal axes were origin-aligned. Silhouette inconsistency was then minimised by simultaneously adjusting the pose and scale parameters associated with each silhouette.

Three views of the 30-view visual hull with and without texture are shown in Figure 7. The texture was obtained by mapping regions of the input images to the faces of the polyhedral visual hull.

7 Summary

A novel method for forming visual hulls from single images has been presented. The method requires only easily obtainable apparatus: a digital camera and two planar mirrors. The mirrors are used to create multiple views of an object that are captured in a single image.

Neither the mirror poses, nor the camera pose are known in advance: silhouette poses are determined directly from the silhouette outlines in the image. No iterative multi-dimensional searches are required to determine the pose parameters. A simple, bounded, one-parameter search is required to solve one equation. Closed form solutions are given by the remaining equations.

Qualitative results have been presented showing visual hull models created using the proposed method. In addition, multiple silhouette sets obtained from several images of the object in different poses have been combined by determining the poses of all silhouettes in a common reference frame. Experimental results showing the visual hull model from such a merged silhouette set have been presented.

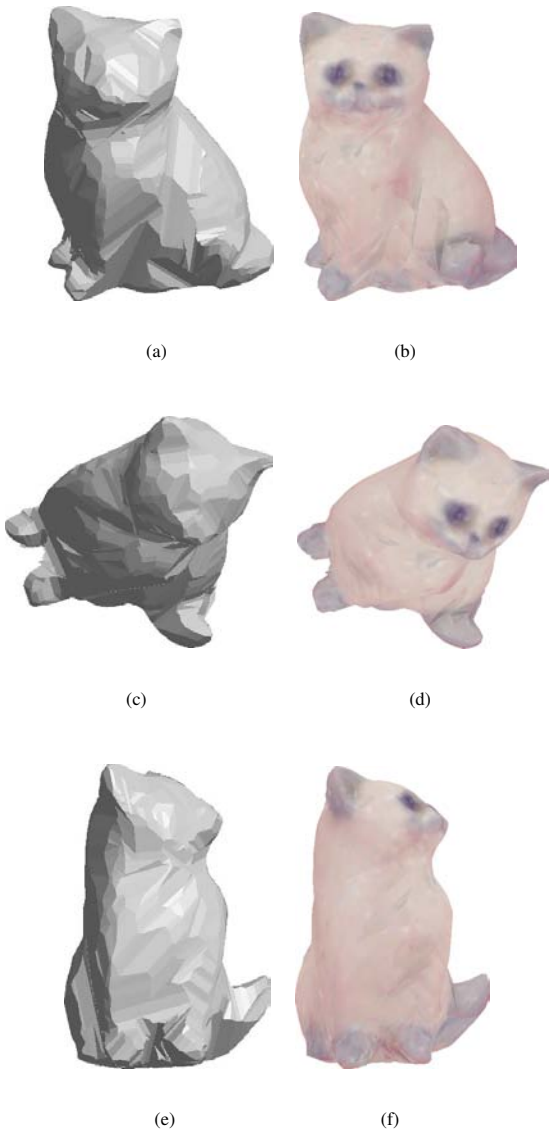


Figure 7: Three views of the 30-view visual hull: the first column shows the 3D shape and the second column shows the textured model.

References

[1] A. Bottino and A. Laurentini. Introducing a new problem: Shape-from-silhouette when the relative positions of the viewpoints is unknown. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11), November 2003.

[2] K. Forbes, A. Voigt, and N. Bodika. An inexpensive, automatic and accurate camera calibration method. In *Proceedings of the Thirteenth Annual South African Workshop on Pattern Recognition*, 2002.

[3] K. Forbes, A. Voigt, and N. Bodika. Using silhouette consistency constraints to build 3D models. In *Proceedings of the Fourteenth Annual South African Workshop on Pattern Recognition*, 2003. Available at `www.prasa.org`.

[4] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[5] P. Mendonça, K.-Y. Wong, and R. Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. Pat. Anal. and Mach. Intel*, 23(6), 2001.

[6] T. Okatani and K. Deguchi. Recovering camera motion from image sequence based on registration of silhouette cones. In *Proceedings of the 6th IAPR Workshop on Machine Vision Applications (MVA2000)*, pages 451–454, 2000.

Inverse Synthetic Aperture Imaging using a 40 kHz Ultrasonic Laboratory Sonar

A. J. Wilkinson, P. K. Mukhopadhyay, N. Lewitton and M. R. Inggs

Radar Remote Sensing Group
Department of Electrical Engineering
University of Cape Town, South Africa
ajw@eng.uct.ac.za pradip@rrsg.ee.uct.ac.za

Abstract

A sonar system operating at 40 kHz in air has been developed to allow the capture of acoustic data in a laboratory environment. The system can serve as a teaching tool for students in seismology, sonar and radar, as well as a useful tool for the development and testing of signal and image processing algorithms. The system can be used for monostatic or bistatic modes of imaging. Range compression is achieved by deconvolution filtering which compensates for the linear system effects of the transducers and other components. A deconvolution filter is generated via a calibration technique in which the system response is measured by pointing the transmitting transducer directly at the receiving transducer. Results are presented which demonstrate the capability of the system for range profiling and 2-D imaging, using the inverse synthetic aperture technique whereby the scene to be imaged is moved across the beam of the sensor. The focused image is obtained by synthetic aperture azimuth focusing / migration techniques. The range and azimuth resolutions achieved with system are discussed.

1. Introduction

Obtaining seismic or radar data in real geophysical applications is a time-consuming and expensive process. For those working in this field, the availability of a smaller scale laboratory measurement system operating in air or water [1] is of potential benefit as it allows easy experimentation with different imaging geometries.

A sonar system operating in air at 40 kHz has therefore been developed as a means of acquiring acoustic data in the laboratory environment. The system has proved useful both as a teaching tool for students learning about signal processing techniques employed in seismology, sonar and radar, as well as a tool for developing and testing new signal processing algorithms.

Traditional seismic imaging usually involves gathering data acquired from a multitude of sensors, which together form a large discretely sampled receiving aperture. A high resolution image is obtain by *migration* processing [2, 3].

In the related fields of radar and sonar, the technique known as *synthetic aperture radar/sonar* (SAR or SAS), [4], involves the formation of an aperture by moving a single sensor past a scene of interest along a track. Alternatively an aperture may also be synthesized by keeping the sensor stationary, and relying on the movement of the object of interest past the sensor [5], as depicted in Figure 1 - this approach is known as inverse synthetic aperture imaging (ISAR or ISAS), and is the basis of the imaging technique applied to the 40 kHz sonar described in this paper.

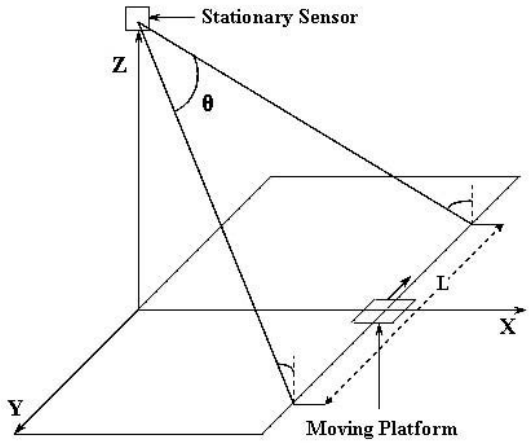


Figure 1: Inverse Synthetic Aperture imaging geometry.

The paper is structured as follows: firstly the system hardware and ISAS imaging geometry are briefly described; thereafter the signal processing and calibration techniques developed to focus the images are described; lastly results from an ISAS experiment are discussed.

2. System Hardware

The philosophy behind the sonar hardware was to design a system based around a PC sound, which could be easily operated from within the MATLAB programming environment, and which would also demonstrate hardware techniques, such as frequency heterodyning, employed in radar systems. The initial prototype was developed as an undergraduate student project at the University of Cape Town [6].

A centre frequency of 40 kHz was chosen for the transmitted pulse, which travels at approximately 340 m/s in air; the corresponding wavelength of 8.5 mm is a practical dimension for the scale of imaging. Piezoelectric transducers were readily available with bandwidths of 4 kHz, corresponding to an achievable range resolution of about 4 cm.

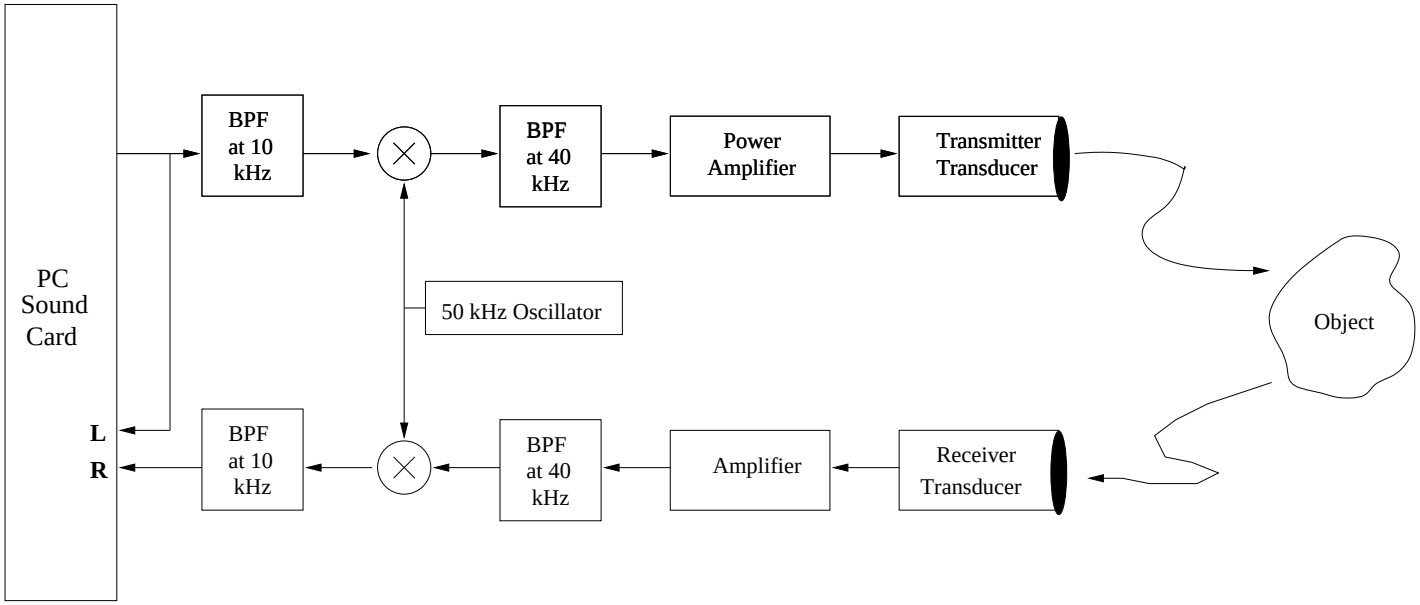


Figure 2: Sonar hardware block diagram.

2.1. Receiver-Transmitter Hardware

A block diagram of the system hardware is shown in Figure 2. MATLAB is used to control the PC sound card, which generates a chirp pulse with spectral components in the audio range between 8 kHz and 12 kHz. This pulse is then mixed with a 50 kHz oscillator, generating a lower sideband at 40 kHz and an upper sideband at 60 kHz. The 40 kHz sideband is passed through a bandpass filter, amplified and fed to the transmitter transducer.

The pulse radiates, is reflected from the scene, and the echo is received by a receiving transducer. The transducers are a shown in Figure 3. The two-way 3 dB beamwidth is approximately 40 degrees. The received signal is amplified and translated down again into the audio range of the sound card, sampled and stored for subsequent digital signal processing. The MATLAB "wavplay" and "wavrecord" commands are used to generate and record the waveforms. The sample rate is set to 44.1 kHz, the maximum rate available on most PC sound cards. In MATLAB, it was not possible to ensure accurate syncronization between the start of playing and recording. This problem was circumvented by simultaneously recording both the transmitted and the received echo on the left and right channels of the sound card as shown in Figure 2, and correlating these recordings.

2.2. Moving Platform

To implement the inverse synthetic aperture technique, the scene must be moved across the beam as illustrated in Figure 1. A wooden platform on wheels (Figure 4) was constructed with dimensions 2.5x1.0 m. The platform can be pulled manually by means of a cord, and its position along its track recorded by an accurate odometer wheel [7], which can be seen on the left hand side of Figure 4. The transmitter is synchronized to the odometer, such that pulses are transmitted and recorded at regular spatial intervals along the track.

Figure 3: Piezoelectric 40 kHz transmitter and receiver transducers.

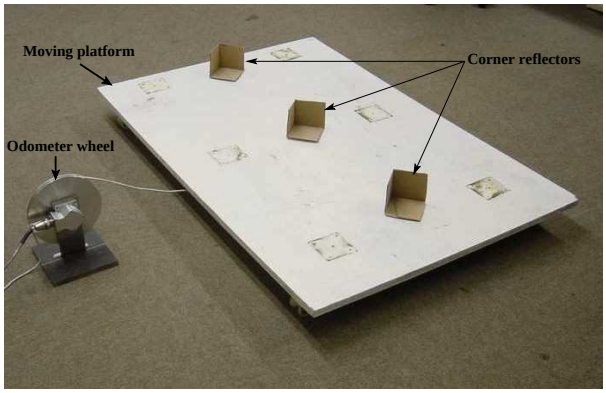


Figure 4: Moving platform containing three corner reflectors.

3. Signal Processing Techniques

The focused image is formed using correlation-type processing, carried out in two steps:

1. range compression using either matched filtering (for optimal signal to noise ratio) or deconvolution processing with frequency domain windowing (for optimal point target response).
2. azimuth focusing using standard migration processing (also known as synthetic aperture focusing), which again may be tailored to give either optimal SNR or optimal point target response.

The results of matched filtering theory under white noise conditions show that the signal to noise ratio is proportional to the energy in the received signal, whereas the resolution is a function of the signal bandwidth [4].

The transmitted pulse was chosen to be a chirp pulse, commonly used in radar applications to satisfy the simultaneous requirements of high energy and wide bandwidth. In the chirp waveform, the instantaneous frequency of a sine wave is swept linearly over time, modelled by,

$$v_{tx}(t) = rect\left(\frac{t}{T}\right) \cdot \cos\left(2\pi\left[f_0 t + 0.5Kt^2\right]\right)$$

where f_0 is the centre frequency, T is the pulse length and $K = \frac{B}{T}$ is chirp rate in Hz/sec. A pulse length of $T = 8$ ms was used corresponding to a physical extent in air of 2.7 m.

The recorded complex baseband signal may be modelled by

$$V_{bb}(f) = H_s(f)\zeta(f + f_0) + N(f + f_0)H_{rec}(f)$$

where $\zeta(f + f_0)$ is a basebanded version of the analytic representation of the impulse response of the target scene, and $N(f + f_0)$ is the basebanded noise referred to the input of the first amplifier. All linear system effects affecting the target response are modelled by the equivalent baseband system transfer function

$$H_s(f) = P(f)H_1(f)H_2(f)H_3(f)H_4(f)H_5(f)H_6(f)$$

where H_1 models the first BPF at 10 kHz, H_2 the BPF and power amplifier at 40 kHz, H_3 the transmitting transducer, H_4 the receiving transducer, H_5 the receiver amplifier and BPF at 40 kHz, and H_6 the BPF at 10 kHz. The function $P(f)$ models the baseband transmitted pulse. The noise is shaped by receiver transfer function $H_{rec}(f) = H_5(f)H_6(f)$.

3.1. Range Profiling

The pulse bandwidth was chosen to be 4 kHz, covering the passband of the transducers. For optimal SNR, under additive noise conditions, the received echo is processed by matched filtering, i.e. the output signal is computed by

$$v_0(t) = \mathcal{F}^{-1}[H_{MF}(f) \cdot V_{bb}(f)]$$

for which $H_{MF}(f) = H_s^*(f)/\sqrt{S_n(f)}$ where $S_n(f)$ is the power spectral density of the noise. If passband is fairly flat, with white noise, then $H_{MF}(f) \approx P^*(f)$.

Although optimal for SNR, matched filtering results in an undesirable point target response if the passband of the transducers are not flat in magnitude and linear in phase. This is the case in this sonar system, as the desired 4 kHz bandwidth extends into the roll-off of the transducers.

Instead, a deconvolution processing approach was adopted, in which the baseband signal is passed through the filter $H(f) = [1/H_s(f)]\, rect(f/B)$. A special calibration procedure was developed in which the transmitter transducer was aimed directly at the receiving transducer, separated by a distance d equal to two metres. This allowed direct recording of the total system response, allowing $H_s(f)$ to be obtained.

With this filter, the processed baseband response is given by

$$V(f) = rect(f/B)\zeta(f + f_0)$$

For a point target at range r modelled by $\zeta(t) = \zeta_0\delta(t - \tau)$ where ζ_0 is the refection coefficient, and $\tau = 2r/c$, the processed frequency response is

$$V(f) = rect(f/B)\zeta_0 e^{-j2\pi(f+f_0)\tau}$$

with a corresponding $Sa()$ shaped time response

$$v(t) = \zeta_0 BSa\left(\pi B(t - \tau)\right) \cdot e^{-j2\pi f_0 \tau}$$

The corresponding 3dB range resolution, is $\delta R \approx \frac{0.89c}{2B} = 3.8$ cm. Frequency domain windowing may also be applied to reduce the sidelobe levels in the response. Application of a Hamming window reduces the first sidelobe to 41 dB below the peak, at the expense of main lobe broadening by a factor of 1.5.

3.1.1. Range Compression Processing Steps

Because of the lack of precise syncronization between transmitting the chirp and the start of recording, an additional step is required in which the range profiles time aligned by correlating the recorded echo with the recording of the transmitted waveform. The sequence of steps used to form a range profile are listed below.

Step 1	The chirp emitted from the output of the sound card and the received echo, are recorded on the left and right input channels and stored in vectors v_{tx} and v_{rx}. These signals lie in the 8-12 kHz intermediate frequency band, centred on frequency $f_{IF} = 10$kHz.
Step 2	The received signal v_{rx} is correlated with v_{tx}, in the frequency domain, i.e. $V = FFT(V_{rx}) \cdot FFT(V_{tx})^*$.
Step 3	The deconvolution filter is applied (its formation is described below).
Step 4	A Hamming window is applied.
Step 5	The signal is converted to a complex analytic signal, by zeroing out the negative frequency components, and inverse transforming to the time domain.
Step 6	The complex baseband signal is formed by multiplying by $e^{-j2\pi f_{IF} t}$. This operation translates the spectral components down to baseband .

The creation of the deconvolution filter involves the following steps:

- The transducers are pointed towards one another with a separation distance of $d = 2$m.
- Steps 1 and 2 above are carried out to obtain a time aligned recording.

- To reduce noise, several echoes are averaged and stored in vector v_{ave}, but this must take place after step 2 has been applied, as averaging requires the data to be time aligned.
- A deconvolution filter is created, i.e. $H_d = [1/H_s]e^{-j2\pi f t_d}$ where $H_s = FFT(v_{ave})$ and $t_d = d/c$. A linear phase correction is included to compensate for the 2m separation of the transducers. This will ensure that application of the deconvolution filter to a received echo does not result in a 2m range shift.

3.2. Azimuth Focusing

Azimuth focusing was achieved using standard *time domain synthetic aperture processing* [4], which is computationally inefficient, but accurate for non-linear trajectories. A point in the focused image is constructed by coherently adding the data collected along the hyperbolic contour in the range compressed data matrix. In complex baseband form, this requires phase correcting each data point prior to addition. The azimuth resolution is a function of the azimuth spatial bandwidth i.e. $\delta x \approx \frac{1}{B_x}$, where $B_x \approx \frac{4\sin(\theta/2)}{\lambda_0}$ is the bandwidth in cycles per metre, θ is the azimuth beamwidth to be processed in radians, and λ_0 is the wavelength. For a two-way idealized beamwidth of approximately 40 degrees, the spatial bandwidth is approximately 162 cycles/metre, and the expected azimuth resolution is approximately 6 mm.

As with the range response, the sidelobe levels of the resulting azimuth point target response may be tailored by appropriately weighting of the summed echoes.

4. Experimental Results

To demonstrate the potential for inverse synthetic aperture imaging, a target scene (photograph in Figure 4) consisting of three corner reflectors constructed from stiff cardboard, was dragged across the beam as illustrated in Figure 1. Pulses were transmitted at 5 mm intervals as the scene progressed along the track, satisfying the azimuth Nyquist criterion $\Delta x < \frac{1}{B_x}$. Additionally, every 10 cm, the corner reflectors were carefully manually rotated about their phase centres to point in the direction of the transducers, hence simulating the angle independent response of ideal point targets.

A plot of a single downrange profile showing the three targets after range compression using a deconvolution filter with Hamming window, and compensation for R^2 loss is shown in Figure 5. The 3 dB range resolution was measured to be 5.83 cm, close to the expected value of 5.55 cm. A comparison between the deconvolution and matched filter responses is shown Figure 6. In both cases a Hamming window was applied. The pulse compression of deconvolution filter response is improved compared to the match filtered response.

The range compressed profiles were assembled into a matrix, displayed in Figure 7. The point targets result in characteristic hyperbolic shaped signatures.

These data were then compressed in azimuth, and the resulting image is shown in Figure 8. Figure 9 shows a cross section, cutting through the 2nd target in the azimuth direction. The same cross section is shown on a dB scale in Figure 10. The 3dB width of the azimuth compressed main lobe was measured to be 0.82 cm, slightly greater than 0.55 cm value calculated with the approximate 40 degree beamwidth. The difference is attributed to lack of accurate compensation for the amplitude and phase response of the transducer beams, and deviations in the target trajectory from a straight line.

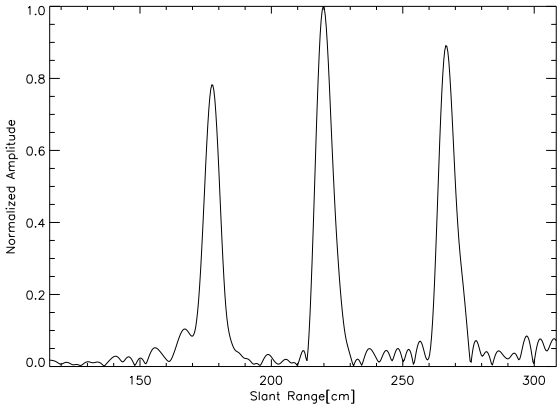


Figure 5: Plot of range compressed profile intersecting three corner reflector targets.

One factor affecting performance of the system was a pulse to pulse fluctuating phase shift in the observed point target response which was the result of variations in the propagation medium. It was observed that even light air disturbances, would result in phase shifts of as much up 20% of a wavelength. A still air environment does however have sufficient phase stability for coherent imaging.

	Expected [cm]	Measured [cm]
Range Resolution	5.55	5.83
Azimuth Resolution	0.55	0.82

Table 1: 3-dB resolutions in range and azimuth.

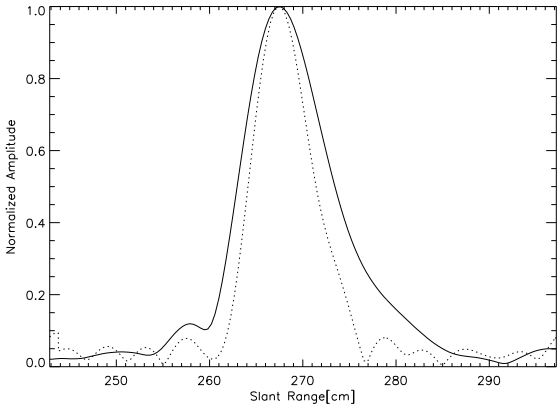


Figure 6: Comparison between matched filter and deconvolution filter responses - in both cases a Hamming window was also applied.

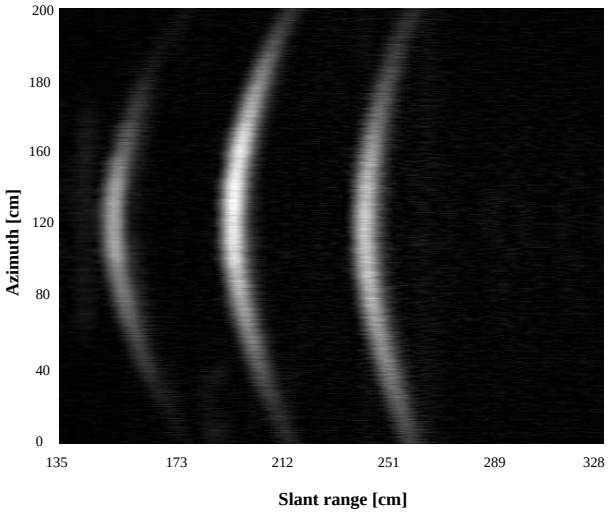


Figure 7: Range compressed data matrix, prior to azimuth compression.

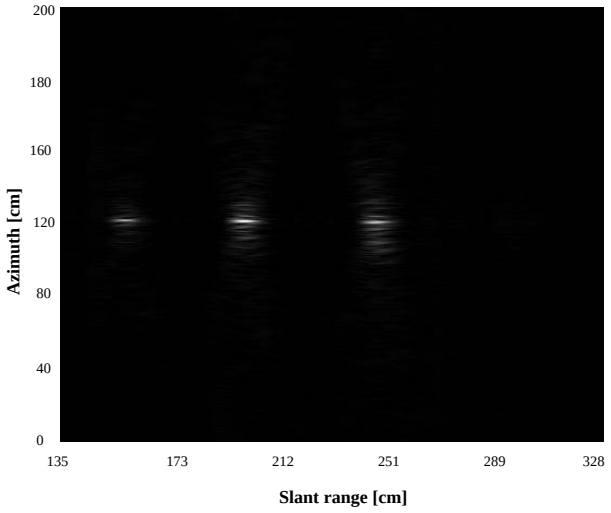


Figure 8: Image after azimuth compression.

5. Conclusions

This work has demonstrated the use of a 40 kHz air-based sonar for conducting coherent imaging experiments in a laboratory scale environment. The results show the application in range profiling and 2-D inverse synthetic aperture imaging. The system has proved to be an excellent tool for teaching and experimenting with radar and sonar concepts.

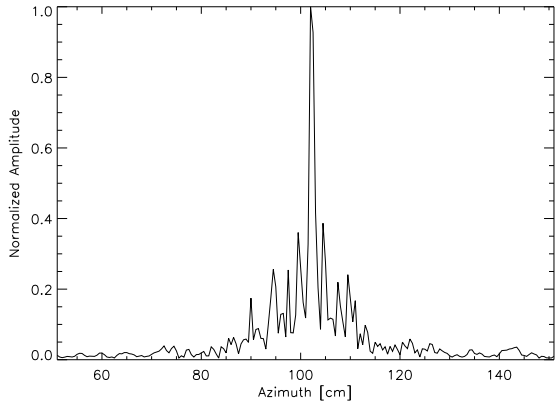


Figure 9: Cross section of a single target in azimuth direction.

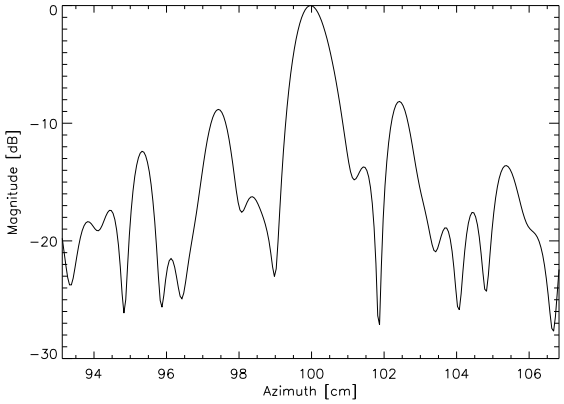


Figure 10: dB plot of the azimuth cross section.

6. References

[1] Mason, I. Osman, N. Liu, Q. Simmat, C. and Li, M., “Broadband Synthetic Aperture Borehole Radar Interferometry”, Journal of Applied Geophysics, Vol. 47,299–308, 2001.

[2] Yilmaz, Ö., “Seismic Data Analysis, Society of Exploration Geophysicists”, Vol. I, 2001.

[3] Berkhout, A. J., “Wave field extrapolation techniques in seismic migration, a tutorial”, Geophysics, Vol. 46(12),1638–1656, 1981.

[4] Bamler, R. and Schattler, B., “SAR Data Acquisition and Image Formation, Ch. 3 in book Geocoding: ERS-1 SAR Data and Systems”,1993.

[5] Wehner D. R., “High Resolution Radar”, Ch. 7, Artech House, 1987.

[6] Korda, S. and Trninic, M., “Design and Construction of an Ultrasonic Radar Emulator”, BSc Thesis, Department of Electrical Engineering, University of Cape Town, 2002.

[7] Nyareli, T., “Development of a Cable Odometer with a Network Interface”, MSc Thesis, Department of Electrical Engineering, University of Cape, 2003.

Fast Stopping in Support Vector Machine Classifiers

Neil Muller

Department of Applied Mathematics
University of Stellenbosch
Private Bag X1, Matieland Stellenbosch 7602 South Africa
neil@dip.sun.ac.za

Abstract

Support Vector Machines have received a lot of attention as a non-linear classifier of late. For realistic datasets, however, the number of support vectors becomes unmanageably large. Most of the proposed approaches to solve this involve approximating the decision boundary. In this article, we propose a simple approach for minimising the number of computations needed to classify a new pattern. For several classes of problems, this can dramatically reduce the time taken to classify elements far from the decision boundary.

1. Introduction

Support Vector Machines (see for example [1, 2]) have received a great deal of attention as a classifier for pattern recognition in the last few years. For instance, in face recognition, support vector machines have been used for pose estimation (see [3]), face detection in complex scenes (see [4]) and feature detection with a face (see [5] or [6]). A recent survey of applications is listed in [7].

One of the major attractions of the support vector machine approach is the easy extension to non-linear problems by means of the so-called "kernel trick" (see for example [8]). This flexibility comes with a price: the non-linear support vector machine is much more expensive to evaluate.

Most of the effort on support vector machines has focused on the computational cost of training the support vector machine (see for example [9]). Comparatively little attention has been focused on the use of the support vector machines, and most of the effort has been spent on approximating the final decision boundary (see [10] or [11]).

In many pattern recognition problems, many of the test cases will be far from the decision boundary. By detecting those cases early, we can stop the evaluation of the support vector machine early and this significantly improve the computational cost of evaluating the support vector machine.

2. Support Vector Machines

2.1. The Linear case

We briefly describe the linear separable case. For a more complete description, see for example [12].

Consider the labelled training set (of size N) $(\mathbf{x}_i, y_i)$ with $y_i = \{-1, +1\} \forall i$. We assume it is linearly separable. Thus there exists some $\mathbf{w}$ and b so that

$$y_i \left(\mathbf{x}_i^T \mathbf{w} + b\right) - 1 \geq 0.$$

We look for the separating hyper-plane which maximises the margin. It can be shown that this reduces to maximising

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \left(\mathbf{x}_i^T \mathbf{w} + b\right) + \sum_{i=1}^{N} \alpha_i$$

subject to

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{1}$$

and

$$\sum_{i=1}^{N} \alpha_i y_i = 0.$$

This is a constrained quadratic programming problem and can be solved using standard optimisation techniques.

Once the solution has been obtained, $\mathbf{w}$ is easily calculated from 1 and then a new sample $\mathbf{z}$ can be classified by evaluating

$$\text{sign}\left(\mathbf{w}^T \mathbf{z} + b\right).$$

2.2. The Kernel trick

Support Vector Machines can easily be extended to non-linear problems by means of the so-called kernel trick. We note that the vectors in the linear case occur only in dot-products. Thus, if we have some non-linear mapping $\phi(\mathbf{x})$, then we only need to be able to calculate $\phi(\mathbf{x})^T \phi(\mathbf{y})$.

Fortunately, it can be shown that there exists a large class of functions $K(\mathbf{x}, \mathbf{y})$ so that $K(\mathbf{x}, \mathbf{y}) =$

$\phi(\mathbf{x})^T \phi(\mathbf{y})$ for some $\phi(\mathbf{x})$. Thus we can replace all occurrences of $\mathbf{x}^T\mathbf{y}$ with $K(\mathbf{x},\mathbf{y})$ and calculate the SVM in the new vector space $\phi(\mathbf{x})$. In this case, evaluating the SVM for a new sample $\mathbf{z}$ reduces to

$$\text{sign}\left(\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}_i,\mathbf{z})\right)+b\right) \tag{2}$$

The function $K(\mathbf{x},\mathbf{y})$ is called the kernel.

It is well known that any $K(\mathbf{x},\mathbf{y})$satisfying Mercer's condition can be used as a kernel for SVM's (see for example [8] or [12]).

3. Fast stopping in SVM's

For non-linear support vector machines, we need to express the decision boundary as a combination of the training set elements. For large training sets, the number of evaluations needed to classify a new sample then becomes intractable.

Several approaches have been proposed to reduce the computational cost of classifying new samples, such as that in [11]. These approaches involve approximating the support vector decision boundary, however.

We note, however, that in a large class of problems (face detection, for instance), most of the new samples will be far from the decision boundary. Thus, if we stop evaluation as soon as it is clear that further support vectors will not change the classification, we should significantly improve the classification performance.

Noting that the decision boundary is expressed as 2 and noting that, for the kernel trick to work, we implicitly assume that there exists a $\phi(\mathbf{x})$ so that

$$K(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y}).$$

We use the standard linear algebra result that

$$K(\mathbf{x},\mathbf{y}) \leq \sqrt{K(\mathbf{x},\mathbf{x})\,K(\mathbf{y},\mathbf{y})}. \tag{3}$$

Using 3 and 2, we note that, given

$$P = \sum_{i=1}^{Q}\alpha_i y_i K(\mathbf{x}_i,\mathbf{z}), \tag{4}$$

the decision will be unchanged if

$$\sum_{i=Q+1}^{N}|\alpha_i y_i|\sqrt{K(\mathbf{x}_i,\mathbf{x_i})}\sqrt{K(\mathbf{z},\mathbf{z})} \leq |P+b|. \tag{5}$$

Since $|\alpha_i y_i|\sqrt{K(\mathbf{x}_i,\mathbf{x}_i)}$ can be precalculated for all the support vectors, evaluating this remainder term involves N multiplications and additions only. Furthermore, since this is a cumulative sum, we can easily evaluate the remainder starting from any Q.

We sort the support vectors by $\|\alpha_i y_i\| sqrt K(\mathbf{x}_i,\mathbf{x}_i)$, so that the LHS of 5 decreases as rapidly as possible.

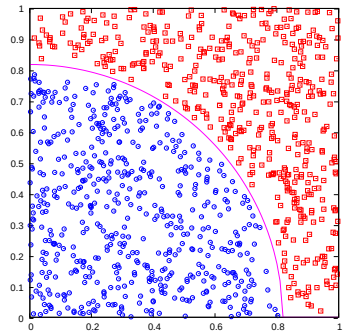


Figure 1: Synthetic Training Data.

Since the $\sqrt{K(\mathbf{x}_i,\mathbf{x}_i)}$ terms can be calculated as soon as the support vector as been trained, the cost of their calculation is not part of the evaluation cost. Thus, to construct the cumulative sum, we need to evaluate $\sqrt{K(\mathbf{z},\mathbf{z})}$ and then use N multiplications and additions. Therefor, the additional computational cost when we don't stop the support vector machine is $O(N)$. The cost of evaluating the kernel is some function of the dimensionality D of the data, so without early stopping the cost of testing an item is $Nf(D)$, whereas in our case its $cf(D)+O(N)$ where, in most cases, $c \ll N$.

4. Examples

Test runs were done in octave (see [13]), using a support vector machine toolkit available from [14].

4.1. Synthetic Data

We generate a synthetic example in 2D of 1000 training points and 100 test points that is separable by a quadratic boundary. We train a simple quadratic SVM to separate the classes. The training data is shown in figure 1.

Training the support vector machines results in a system with 145 support vectors.

We evaluate the classification performance using both the full SVM approach and our modified example. Unsurprisingly, both methods return a 100% classification success rate.

The full classifier takes 86 seconds to classify the data on a 800MHz Pentium III, while the modified algorithm takes 80 seconds. The small gain in performance is due to the low dimensionality of the data, so the kernel evaluation cost is quite low.

4.2. MNist database

To demonstrate the technique on real world data, we use the MNIST database provided by [15]. This database consists of a normalised and labelled subset of the NIST Special Databases 1 & 3 of handwritten digits. Each element of the database is a 28x28 pixel images, giving

features vectors with 784 elements.

Since we are not interested in accuracy, we have not tried to optimise the support vector parameters to maximise accuracy. We used a simple polynomial kernel of degree 5. We trained two support vector machines, one for class one against all other classes, and another for testing class two against all other classes. In each case, we used a subset of 2000 of the labelled training data, giving support vector machines with 268 and 422 support vectors respectively. We tested with 1000 elements from the test set.

The support vector machines achieve accuracy of 60 % for the 1 versus the rest case and 65 % for the 2 versus the rest case. The running time was 1765 seconds and 2806 seconds respectively for the full case, and 982 seconds and 1223 seconds respectively for the modified case.

5. Conclusions

We demonstrate how, for problems where many of the tested samples will be far from the decision boundary, evaluating an approximation of whether the remaining support vectors can change the decision allows us to abort computation early.

As this evaluation can be done cheaply, with many elements being recalculated, for a large class of real-world problems, this significantly improves the performance of the classifier. As we evaluate the full SVM on decisions close to the boundary, there is no loss of accuracy as well. The cost of checking the approximation is small in comparison to evaluating the full SVM, so the penalty paid is not a significant factor in this case.

This approach can also be combined with many of the existing approximation approaches, with the corresponding loss of accuracy.

6. References

[1] C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 1–25, 1995.

[2] Edgar E. Osuna, Robert Freund and Federico Girosi, "Support vector machines: Training and applications," Tech. Rep. A. I. Memo 1602, MIT Artifical Intellegence Laboratory, Mar. 1997.

[3] Jeffery Ng and Shaogang Gong, "Multi-view face detection and pose estimation using a composite support vector machine across the view sphere," in *IEEE International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, 1999.

[4] Edgar Osuna, Robert Freund and Federico Girosi, "Training support vector machines: an application to face detection," in *CVPR' 97*, 1997.

[5] Jeffrey Huang, David Ii, Xuhui Shao and Harry Wechsler, "Pose disriminiation and eye detection using support vector machines (SVM)," pp. 528–536. Springer-Verlag, New York, 1998.

[6] F. Smeraldi, N. Capdeviele and Bigün, "Facial features detection by saccadic exploration of the Gabor decomposition and support vector machines," in *Proceedings of the 11th Scandinavian Conference on Image Analysis*, 1999.

[7] Hyeran Byun and Seong-Whan Lee, "Applications of support vector machines for pattern recognition: A survey," in *Pattern Recognition with Support Vector machines*, Seong-Whan Lee and Alessandro Verri, Eds. IEEE, Aug. 2002, IEEE Computer Society.

[8] Bernhard Schölkopf, "Statisitcal learning and kernel methods," Tech. Rep. MSR-TR-2000-23, Microsoft Research, Microsoft Coporation, Feb. 2000.

[9] Edgar Osuna, Robert Freund and Federico Girosi, "An improved training algorithm for support vector machines," in *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP97)*, Sept. 1997.

[10] Christopher J. C. Burges, "Simplified support vector decision rules," in *International Conference on Machine Learning*, 1996, pp. 71–77.

[11] B. Schölkopf, P. Knirsch, A. Smola and C.J.C. Burges, "Fast approximation of support vector kernel expansions and an interpretation of clustering as approximation in feature spaces," in *DAGM Symposium Mustererkennung*, R. J. Ahler, P. Levi, M. Schanz and F. May, Eds., Berlin, Germany, 1998, pp. 124–132, Springer-Verlag.

[12] Alex J. Smola and Bernhard Schölkopf, "A tutorial on support vector regression," Tech. Rep. NC2-TR-1998-030, ESPRIT Working Group in Neural and Computational Learning II, Oct. 1998.

[13] ," GNU Octave Homepage, http://www.octave.org/.

[14] Anton Schwaighofer, ," Matlab SVM toolkit, www.cis.tugraz.at/igi/aschwaig/software.html.

[15] ," The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/.

A simple method for visualizing labelled and unlabelled data in high-dimensional spaces

J. R. Greene

Department of Electrical Engineering, University of Cape Town
Rondebosch, 7001, South Africa.

jrgreene@eng.uct.ac.za

Abstract

The low-dimensional visualisation of high-dimensional data is a valuable way of detecting structure (such as clusters, and the presence of outliers) in the data, and avoiding some of the pitfalls of blind data manipulation. Projection based on principal component analysis is widely employed and often useful, but it is a variance-preserving projection which takes no account of class labels, and may, for this reason, hide significant structure.

Here we present a very simple method which appears to yield useful visualizations for many datasets. It is based on a random search for a linear transformation, and projection into a two-dimensional visual space, which maximises an objective measure of class separability in the visual space. The method, which can be thought of as a variant of projection pursuit with a novel interest measure, is demonstrated on datasets from the UCI Repository. Tentative interim results are also given for a proposed extension based on spectral clustering, for extending the method to unlabelled data.

1. Introduction

The low-dimensional visualization of high dimensional data is often used to gain insight into the presence of structure (such as clustering) or to identify outliers in the data, to aid in the selection of classifiers and generally to try to avoid some of the pitfalls inherent in blind data manipulation. A popular and frequently effective approach is to project the data onto its first few principle components. However this is a variance-preserving linear transformation which takes no account of class labelling, and a principal component projection may be far from optimal for revealing structure. It should be noted that a recent proposal (1) for modified versions of principal component projection addresses this problem and in many (but not all) cases provides markedly superior visualizations either through a normalisation (half-whitening) approach or by biasing the projection taking class labels into account.

Here we present a very simple, but often highly effective, visualization strategy based on random linear transformation and projection into a two-dimensional visual space. Multiple random visualisations are performed, and that one selected which maximises a simple objective measure of separation in the visual space. In detail, it is based on the following principles and observations.

2. Outline and motivation

1. Complex nonlinear mappings are powerful in revealing distributional structure, but they are also capable of constructing false structure – artefacts of the transformational process (in a process somewhat analogous to overfitting in supervised learning). We therefore opt for the conservative strategy of using only linear transformations. These are partially structure-preserving, in that they cannot break up existing clusters (i.e. make a unimodal distribution multi-modal). Since we are interested in transformations which involve projection into a space of lower dimensionality they can, of course, create false clusters by the adventitious superposition of points in the projected space.

Clearly the set of linear transformations is not sufficiently powerful for visually revealing structure in cases where the data resides on a nonlinear manifold of the representation space. However powerful methods (auto-associative neural networks,

multidimensional scaling, Self-organising maps, Sammon mapping, Principal Curves, Laplacian eigenmaps, ISOMAP, kernel PCA and other manifold learning methods) exist to deal with this case. The remarkably limited range of benchmark data typically used to justify and demonstrate these methods (e.g. toy data such the 'swiss roll', images of facial expressions, 2-D projections of rotated 3-D objects, and hand written numerals such as in the NIST data set) suggest that the kind of nonlinear inter-variable correlation which gives rise to such manifolds is uncommon in real-world classification data sets. It has been our experience that, at least in the data sets typically used for classifier benchmarking, well-chosen linear transformation and projection onto R2 often suffices to reveal clusters and outliers, and nonlinear manifold learning approaches do not contribute to significantly improved performance.

2. Multiplication of the data by a matrix of random numbers spans the space of all possible linear transformations. Normal- or uniformly-distributed numbers seem an obvious choice; we found the latter slightly more effective in rapidly finding good projections for class-separation. To project [p × d] data onto R2 we postmultiplied by a uniformly random [d × 2] matrix. Adventitious correlation between the columns of the random matrix (especially in the case of lower-dimensional data) often resulted in a strongly elliptical projection, so we used singular value decomposition to transform the matrix into one with uniformly-random but mutually orthogonal columns of unit length (i.e. an orthonormal random matrix), ensuring that the projected data fully spans the projection space.

3. Manually observing a sequence of random projections is sometimes enlightening but tedious so we sought to automate the selection of the most favourable projection. Initially we tried Thornton's Separability Index *s* as a suitable criterion. This is a simple, rapidly computable measure of class separation, defined as the fraction of points which share a class label with their nearest neighbour. In a previous paper (2) we showed its effectiveness in feature subset selection; here we apply it to the planar transformed representation and use it as a surrogate of visual separability. Empirical tests showed that it performs well in this role, with one potential limitation, relating to its limited 'dynamic range'.

For realistic data *s* is never less than 0.5 (s = 0 would require artificially constructed data such as an interlaced pair of grids of points with opposite class membership). At the other extreme, *s* 'saturates' at a value of unity. Complete separation, such that each point has a nearest neighbour of the same class, results in *s* = 1, which does not change with further distancing of the class centroids; thus *s* is insensitive to class separation as soon as the nearest-neighbour criterion has been realised. The limited range over which s is effective sometimes results in a projection which is not optimally effective in visual terms. To counter this we considered using a different criterion of separability – that of the 'hypothesis margin' *h*. This is the summed difference between the inter-class and intra-class nearest-neighbour pairwise distances. The hypothesis margin is an unbounded measure of separability so it does not exhibit the saturation effect exhibited by *s*. On the other hand it is less effective when the data classes overlap with a horizontal asymptote of zero. We conjectured that due to this complementary aspect of their behaviour, a hybrid separation index *hs* consisting of the sum of *h* and *s* might correlate better with visual separability over a wide range of datasets. This conjecture seemed to be supported by informal tests over a wide range of datasets from the repository. However the difference is rather marginal, so for speed and simplicity, Thornton's Separability Index *s* is used as a criterion of separation in the examples shown below.

4. The search for the optimal projection is non-convex, with many local maxima. Stochastic and evolutionary search methods are effective in this situation, and our first effort was to use Population-based Incremental Learning (a simple but powerful abstraction of the genetic algorithm). It was noticed however that frequently good solutions were obtained in the first generation, with little or no subsequent improvement. It appears

that this is one of the situations (feature selection can be another) where extremely sparse random searches in vast search spaces can nevertheless be effective, due to the multiplicity of local maxima which are competitive with the global optimum. If a substantial fraction of possible solutions are acceptable, it is likely that one of them will rapidly be encountered in a sparse random search. Pure random search, with its minimal computational overhead, may actually be more efficient in this situation than a more directed guided stochastic search (or a more principled search based on mathematical programming or gradient descent, which is almost certain to be entrapped at a sub-optimal point.

The algorithm is summarised in the following pseudocode:

Given data X [p x d] , t [p x 1], ± 1
and a maximum number of iterations *maxiter* (a user-chosen parameter, typically 30-100)

iter = 0
max_sepindex = 0

while iter < maxiter & max_sepindex < 1
create a uniformly random [p x 2] matrix R
and transform it into an orthonormal matrix
  R = R_{orth}
  X_p = X*R_{orth}
  s = sepindex(X_p, t) (separability index)
  if s > max_sepindex
    max_sepindex = s
    R_{best} = R
  end
  iter = iter +1
endwhile

Plot column 1 vs Column 2 of X_p = X*R_{best}

The plots in the appendix show the results of applying the method to six datasets from the UCI repository. It can be seen that an effective visualisation results, with the dichotomous clustering of the data being vividly evident in the case of the Breast Cancer, Wine, Heart and Thyroid datasets. In the case of the Thyroid data it is further evident that the classes, though readily separable, require a nonlinear discriminant.

3. Extension to unlabelled data

Using class separability in visual space as a figure of merit requires labelled data. We tried extending the method to the visualisation of unlabelled data by using a spectral method (3) for partial dichotomous labelling of the data in the original representation space. This is achieved by a thresholding of the 2nd ('*Fiedler*') eigenvector of an affinity matrix formed by normalising a kernel matrix constructed using the data. It is based on the observation that points clearly belonging to different clusters are likely to have markedly different values in the Fiedler eigenvector. The subset of points so labelled is used to find the optimal transformation for visual mapping and this is applied transductively to the whole dataset.

The method proposed is as follows(3):

1. Construct a gaussian kernel (with the spread factor set to a user-determined factor of the modal nearest-neighbour distance as determined by histogram).

2. Use additive normalisation to convert the kernel to a Laplacian stochastic affinity matrix L.

3. Perform an eigenstructure decomposition on L.

4. Sort the second eigenvector (the 'Fiedler eigenvector') L2 by value and reorder the dataset instances accordingly.

5. Label a small fraction of the topmost and bottom-most instances with opposite class labels, leaving the instance in-between unlabelled. This results in a partial labelling of he dataset, with the labelled instanced being assigned to distinct classes with a high degree of confidence.

6. Perform the random projection method outlined above using only the instances labelled in the step above.

7. Plot all the points are plotted using the random transformation matrix which was found to optimise the visual separation of the points on the subset of data in the previous step.

This process requires a small modification to the calculation of visual separability, since with the limited labelled dataset constructed a separability index of 1 is rapidly reached, resulting in a loss of information about the merits of further transformations. Instead of the more complex hybrid separability index proposed above, Thornton's separability index was augmented, once it reached the value of 1, by adding the Euclidean distance between class means. Thus an unbounded measure of separability results, which continues to guide the search in the direction of greater separation even when complete separability has been achieved for the spectrally-labelled data.

A problem remains with the above proposal, which is still under investigation. Spectral clustering methods, though very powerful, depend fairly critically on the parameter σ of the gaussian kernel used to determine the affinity matrix, and it is unclear, in the present setting, how an optimal value for σ should be obtained. To circumvent this problem and test the proposed method in broad principle, an approximation was used in which the leading eigenvalue of the input correlation matrix X^TX was used as a surrogate for the Fiedler vector.

Results were very promising with the Wisconsin breast-cancer dataset, and excellent visualizations were obtained (virtually indistinguishable from the results illustrated above) without the use of class labels. However further work is required to see how well the method generalises to other datasets. It is conjectured that a robust solution will require the use of the full spectral method with a more principled approach to selecting σ. A possibility under investigation is to set the value of σ equal to a fixed fraction of the modal pairwise nearest neighbour distance as determined by a histogram. Another possibility is to adjust sigma using a maximisation of the 'eigengap strategy'. These are all under investigation.

4. Conclusion

A simple but effective method has been presented for visualising structure (dichoomous clustering, outliers) in high-dimensional data making use of class labels, and demonstrated on some datasets. It is related to well-known projection pursuit methods , but makes efficient use of low-overhead random search, and employs Thornton's separability index as a novel interest measure. It has also been proposed that the method can be extended to unlabelled datasets by inferring a partial labelling using a spectral method, and applying the mapping so found transductively to the dataset as a whole. The method was implemented in broad principle and shown to be effective on a single dataset, but much work remains to be done solve some outstanding problems and to show that the extension is both robust and generally applicable.

5. References

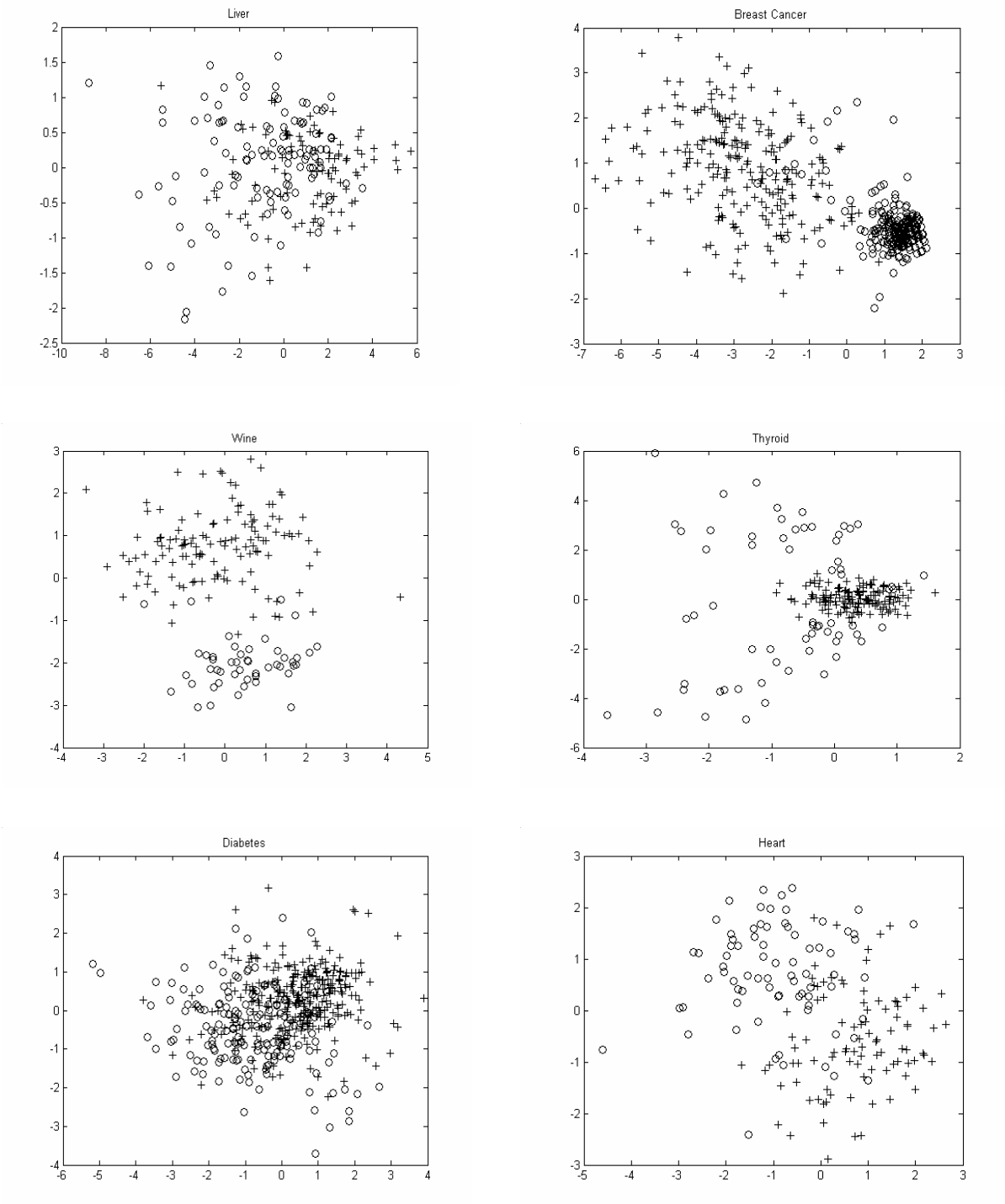
1. Y Koren and L Carmel, "Visualization of Labelled Data Using Linear Transformations". *Proceedings of IEEE Information Visualization Conference InfoVis03*, (2003), IEEE pp121–128, 2003.

2. J Greene, "Feature subset selection using Thornton's separability index and its applicability to a number of sparse proximity-based classifiers". *Proceedings of the Pattern Recognition Association of South Africa*, PRASA, Franschhoek, 2001.

3. D Verma and M Meila, "A Comparison of Spectral Clustering". *Technical Report 2003*.UW CSE Technical Report 03-05-01

4. C.J.C. Burges. "Geometric methods for feature extraction and dimensional reduction: A guided tour". Microsoft research Technical report, 2004.

Appendix 1. Result of the visualisation method for labelled data



In each case the visualization provides useful information on the distributional structure. In the case of the Breast Cancer, Wine, Thyroid and Heart datasets, dichotomous clustering was very clearly visible, with visual separability indices in the range 0.5-1. It is clear that the Breast Cancer, Wine and Heart datasets are linearly separable, while the Thyroid data is readily separable, but requires a non-linear discriminant. In all cases 50 iterations sufficed to find an optimal projection.

Survey of JPDA algorithms for possible Real-Time implementation

M.J. Goosen, B.J. van Wyk, M.A. van Wyk

Rand Afrikaans University, Johannesburg, South Africa.

mjgoosen@ing.rau.ac.za

French South African Technical Institute in Electronics at the
Tshwane University of Technology, Pretoria, South Africa

ben.van.wyk@fsatie.ac.za mavw@fsatie.ac.za

Abstract

In this paper different Data Association Algorithms for multiple target tracking systems are discussed and compared. The aim is to find the best algorithm suitable for real-time hardware implementation. Simulations for the various algorithms were done using MATLAB. The comparison between the algorithms was based on CPU time and graphical display for the tracked values.

1. Introduction

An essential part of any surveillance system is the ability to efficiently track objects within the system's range. The most important components of such a system are sensors and computer subsystems. The data obtained from the sensors are used in the computerized subsystem to confirm the target tracks.

One of the most significant tracking problems is that of tracking the states of unknown multiple maneuvering targets in the presence of severe clutter. There are numerous approaches and algorithms to address this problem. The tracking performance of such a tracking algorithm is mainly dependent on the accuracy of the target state estimator used. The most widely used state estimator is the Kalman filter. Other state estimators like the interacting multiple model (IMM) estimator is also used. The IMM estimator is effective for instances of tracking maneuvering targets.

The expected locations of the tracked targets are computed using the prediction models. These are the tracking filters referred to in the previous paragraph. The estimated positions of the targets will usually not coincide with the new set of measurements received from the sensors. The probability that a certain measurement represents a certain target can then be calculated. These probabilities are joint probabilities that are computed in the presence of the other measurements and targets. In some instances this can lead to great amounts of computations and memory use, justifying real-time hardware implementations.

In a Multi-Target Tracking (MTT) system the data received will consist of true targets, background clutter and false signals. The MTT sensor requires a broader field of view than in a Single Target Tracking (STT) system. The broader coverage ensures higher likelihood of target detection. Most modern surveillance systems make use of MTT systems. The most important element in the MTT system is the data processing stage, where the complex data associations and prediction are computed.

There are three main Data Association Methodologies of interest in a MTT system. These methodologies are Global Nearest Neighbor (GNN), Joint Probabilistic Data Association (JPDA) and Multiple Hypothesis Tracking (MHT). These methods increase in efficiency from GNN to JPDA to MHT but also increase in complexity and number of computations [1]. Taking into account the current development in computer capabilities it is possible to successfully implement these methods on a computerized system.

The MHT method creates multiple track hypotheses from the association of feasible observations and tracks. The final observation-to-track decisions can be deferred until more data becomes available from future scans. When new data is received the probable hypotheses are confirmed and improbable hypotheses are pruned. The JPDA methods update the track with a weighted sum of the feasible observations. All feasible observation-to-track associations are taken into consideration when calculating the weights (probability values). The state estimate of the track therefore does not depend on a single observation but on all observations falling inside the track gate. Multiple observations in a gate occur when gates overlap or clutter is detected inside a gate. Great amounts of processor power are needed when processing a large number of tracks in the above mentioned methods.

The JPDA method shows an exponential increase in computational time with an increase in targets. When constrained by the limited processor power available and a large number of tracks, the algorithms are still not real-time implementable. Several sub-optimal JPDA algorithms were proposed such as those of Roecker [4] and Ding *et al*. [12].

The focus of this paper will be on different JPDA algorithms for Data Association in a MTT system. They will be evaluated according to complexity and computation towards possible real-time hardware implementation.

The historic optimal JPDA algorithm, as in [10], has five basic steps, namely

1. Generation of a validation matrix. Binary matrix representing all feasible observation- track pairing (result of Gating).

2. Generation of feasibility matrices from the validation matrix. This represents all the non-competing events in which an observation j is associated with only one track t except for $t = 0$. The $t = 0$ track is the clutter track (measurement originated from clutter). The number of feasibility matrices rapidly explodes as the number of tracks and observations increase.
3. Calculation of the probabilities β_{jt} for each track-observation pair at time step k from the feasibility matrices.
4. Updating the state estimate error covariance of the Kalman tracking filter.
5. Updating the state estimate vector for each track using all plausible measurements, each multiplied by the appropriate scalar weighting coefficient β_{jt}.

The last two steps as mentioned above are standard optimal linear filter operations, and will therefore not be discussed in this paper.

There are a few approaches of the JPDA algorithm that is of importance for this paper. These approaches are those of Van Wyk *et al.* [3], Fisher and Casasent [9] and Zhou & Bose [7]. The aim of this paper is to determine which of these algorithms are best suited for possible real-time implementation. A comparison between the algorithms will be based upon the CPU time needed to complete the algorithm, together with an analysis of a graphical display for the tracked simulation data. Factors like memory usage, computational complexity and number of computations will play a significant role in a real-time implementation. Therefore the above-mentioned algorithms will also be analysed for their complexity.

The conventional JPDA algorithm as well as the JPDA algorithms of Van Wyk *et al.* [3], Fisher and Casasent [9] and Zhou & Bose [7], will be discussed in section 2. The simulation is discussed in section 3 with the concluding remarks presented in section 5.

2. The JPDA algorithm

The following presents a brief explanation of the conventional JPDA Algorithm [7], [8], [9], [10]. The differences in the other JPDA algorithms used are also explained. Assume there are m measurements and n targets being tracked. The first step is to construct the binary validation matrix. The Validation matrix is an $m \times (n + 1)$ rectangular matrix,

$$\Omega = \left[\omega_{jt}\right] = \begin{matrix} \overbrace{\begin{matrix} 0 & 1 & 2 & \cdots & n \end{matrix}}^{t} \\ \begin{pmatrix} 1 & \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ 1 & \omega_{21} & \omega_{22} & \cdots & \omega_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega_{m1} & \omega_{m2} & \cdots & \omega_{mn} \end{pmatrix} \end{matrix} \begin{matrix} 1 \\ 2 \\ \vdots \\ m \end{matrix} \Bigg\} j \qquad (1)$$

In the above matrix the value $\omega_{j0} = 1$ implies the measurement originated from clutter where

$$\omega_{jt} = \begin{cases} 1 & \text{if measurement } j \text{ is in gate of target } t \\ 0 & \text{if measurement } j \text{ is not in gate of target } t \end{cases} \qquad (2)$$

for $j = 1,2,...,m$, and $t = 1,2,\ldots,n$.

The feasibility matrices are now generated from the validation matrix. These feasibility matrices are subject to the following two restrictions:

1. Each measurement can have only one origin, whether it is a real target or clutter.
2. No more than one measurement can originate from any given target.

Thus only one element per row may be chosen from the validation matrix so that there is at most one value per column.

The number of feasibility matrices increases exponentially with increases in m and n. These feasibility matrices provide a format in which to examine every possible observation-track combinations. This is also known as the individual events, χ. All of these events may be represented by a feasible matrix,

$$\hat{\Omega}(\chi) = \left[\hat{\omega}_{jt}(\chi)\right] \qquad (3)$$

This consist of the unit values in the validation matrix, Ω, corresponding with the associations assumed in the event χ. $\hat{\Omega}$ is the same size as the validation matrix Ω with $\hat{\omega}_{jt} = 1$ only if measurement j is hypothesized to be either from clutter ($t = 0$) or from a target t ($t \neq 0$).

For each $\hat{\Omega}$ the conditional probability for the data association hypothesis, or the feasible event χ, needs to be calculated. The simplified version of the formula from [10] is

$$P\left(\chi\left(\hat{\Omega}\right)\middle| Z\right) = \frac{1}{c}(P_0)^{\min(n,m)-m_d} \prod_{j:\omega_{jt}=1} P_{jt} \qquad (4)$$

where Z is the set of all measurements received up to time index k, c is a normalization constant, m_d the number of detected targets in the event χ, and $\hat{\omega}_{jt} = 1$ indicates that the measurement j is associated with the track t. The normalization constant c is obtained by summing the conditional probabilities, $P\left(\chi\left(\hat{\Omega}\right)\middle| Z\right)$, over all the events $\chi\left(\hat{\Omega}\right)$. The values for P_{jt} are as follows

$$P_{jt} = \begin{cases} N(\tilde{z}_j^t; 0, S^t) P_D & \text{,if } \omega_{jt} = 1 \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

for $j = 1,2,\ldots,m$, and $t = 1,2,\ldots,n$.

Here λ is the clutter density, P_D is the probability of detection and $N(\tilde{z}_j^t; 0, S^t)$ is a normal distribution density function with zero mean and covariance matrix S^t.This

covariance matrix is obtained from the tracking filter, usually a Kalman Filter.

The next step is to calculate the *a posteriori* probability β_{jt} from the conditional probabilities, $P\left(\chi\left(\hat{\Omega}\right)\middle|Z\right)$ by the following equation,

$$\beta_{jt} = \sum_{\chi(\hat{\Omega})} P\left(\chi\left(\hat{\Omega}\right)\middle|Z\right)\hat{\omega}_{jt}$$
$$\beta_{0t} = 1 - \sum_{j=1}^{m} \beta_{jt} \tag{6}$$

for $j = 1,2,\ldots,m$, and $t = 1,2,\ldots,n$.

These values for the *a posteriori* probability β_{jt} are then used to update the innovation vector as well as the covariance matrix of the Kalman filter.

The biggest computational burden of the JPDA algorithm is clearly the generation of the feasibility matrices and the evaluation thereof. Some approaches completely circumvent the formation of feasibility matrices. These algorithms is more likely to be real-time implementable. Two such algorithms were developed by Van Wyk *et al.* [3] and Fisher and Casasent [9]. The algorithm of Van Wyk *et al.* [3] was developed using a Projection Onto Convex Sets (POCS) method. Fisher and Casasent [9] also derived a new method for calculating the β_{jt} coefficients. Another approach is to derive a better algorithm for evaluating all the feasible event matrices. Such an algorithm was developed by Zhou and Bose [7]. Their algorithm efficiently visits all the nodes in the hypothesis tree by using the *exhaustive search with certain constraints* model to solve this combinatorial problem. The PJPDA algorithm of Van Wyk *et al.* [3] is a totally new approach and is therefore discussed in more detail in the following section.

2.1. The Projection-Based JPDA (PJPDA)

The POCS JPDA, of Van Wyk *et al.* [3], can be described as a solution for the following constrained optimization problem: Determine the probabilities (weights) β_{jt} from the quantities

$$p_{jt} = \begin{cases} (1 - P_{D,t}) & \text{if } t = n' \\ f_t\left[z_j\right] P_{D,t} & \text{otherwise} \end{cases} \tag{7}$$

for $j = 1,2,\ldots,m$, and $t = 1,2,\ldots,n'$, such that the row and column constraints $\sum_{t=1}^{n'} \beta_{jt} = 1$, $\sum_{j=1}^{m} \beta_{jt,t\neq n'} \leq 1$ and $0 \leq \beta_{jt} \leq 1$ are satisfied and $\sum_{jt} (p_{jt} - \beta_{jt})^2$ is minimized. $P_{D,t}$ is the probability of detection, $f_t[z_j]$ is the probability density of measurement j associated with target t. This is taken to be scaled normal distribution with zero mean and covariance matrix S^t of the Kalman Filter. The assumptions are that there are $t = 1,..,n'$ tracks, with $t = n'$ denoting the clutter track, and $j = 1,..,m$ observations. The use of $t = n'$ for the clutter track is merely for convenience. The algorithm is executed at every time step k. For simplicity, time dependency is omitted.

For applying POCS to the JPDA optimization algorithm, the row constraint matrices are represented by the set,

$$C_r = \left\{ \beta_r \in \mathbb{R}^{mn'} : \begin{array}{c} \beta_r = [\beta_1,\ldots,\beta_m], \\ \bar{\beta}_j = \left[\beta_{j1},\ldots,\beta_{jn'}\right], \\ \sum_{t=1}^{n'} \beta_{jt} = 1, \\ 0 \leq \beta_{jt} \leq 1 \end{array} \right\} \tag{8}$$

which can be shown to be closed and convex. The column constraint matrices are represented by the set,

$$C_c = \left\{ \beta_c \in \mathbb{R}^{m(n'-1)} : \begin{array}{c} \beta_c = \left[\beta_1,\ldots,\beta_{(n'-1)}\right], \\ \bar{\beta}_t = [\beta_{1t},\ldots,\beta_{mt}], \\ \sum_{j=1}^{m} \beta_{jt,t\neq n'} \leq 1, \\ 0 \leq \bar{P}_{jt} \leq 1 \end{array} \right\} \tag{9}$$

which is also closed and convex. $C_0 = C_r \bigcap C_c$ is the intersection of the above convex sets, and is non-empty. The closedness, convexity and non-empty C_0 allow the use of a POCS methodology to solve the JPDA problem. The main idea is to calculate $T_0(\mathbf{p})$, the projection of

$$\begin{aligned} \mathrm{p}^{\mathrm{T}} &= [\bar{p}_1,\ldots,\bar{p}_m], \\ \bar{p}_j^{\,T} &= [p_{j1},\ldots,p_{jn'}], \end{aligned} \tag{10}$$

onto $C_0 = C_r \bigcap C_c$. The projection $T_0(\mathbf{p})$, will be the solution. The set C_0 is considerably more complex in structure than C_r and C_c, and a direct realization of $T_0(\mathbf{p})$ was avoided. The projections $T_r(\mathbf{p})$ and $T_c(\mathbf{p})$ onto the sets C_r and C_c is easily realizable. The algorithms for these projections are explained further in Van Wyk *et al.* [3]. This algorithm is easily implementable and less complex in comparison to the conventional JPDA, which makes it ideal for real-time implementation.

2.2. DFS Approach

Zhou & Bose [7] proposed a method based on a depth-first search (DFS) approach. This approach was specifically aimed at developing a fast method of generating the data association hypotheses and computing the conditional probabilities. This is a *track-oriented* approach where each track is hypothesized to be either: undetected, terminated, associated with an observation, or linked with a maneuvre. The approach is exactly like the classical JPDA algorithm, except for the use of their DFS algorithm in generating the feasibility matrices and calculating the conditional probabilities.

They approach the JPDA as a combinatorial problem called an *exhaustive search with certain constraints*. A general description is as follows. There are m variables X_j ($j = 1,2,\ldots,m$). These values, X_j, belong to a finite and linearly ordered set Z_j. A candidate solution is a m-tuple, (X_1, X_2,..., X_m). The object is to find one solution or all solutions under the above-mentioned

constraints. For this, Zhou & Bose developed the DFS procedure to solve the problem mentioned above. In the context of multiple-target tracking the problem can be described as follows. Assume there is m measurements and n targets. Let X_j (j = 1,2,…,m) denote the measurement j. The individual values of X_j identifies the target ($X_j = t$, t = 1,2,…,n) or clutter (X_j = 0, t = 0) associated with measurement j. The set Zj are defined by

$$Z_j = \{t \mid \omega_{jt} = 1\}, \text{ for } j = 1,2,..,m, \text{ and } t = 1,2,..,n. \quad (11)$$

Since $\omega_{j0} = 1$, this implies $0 \in Z_j$ for j = 1,2,…,m. The gating process determines these values where $t \in Z_j$ if measurement j falls in the validation gate of target t.

The DFS procedure is designed to visit all the nodes in the hypothesis tree to evaluate all possible feasible events. Conditional probabilities, see (4), are computed from the feasible events, χ, where the P_{jt} can be computed in advance. After each $P\left(\chi\left(\hat{\Omega}\right)\middle| Z\right)$ is computed the β_{jt} values as well as the normalization constant c can be updated.

2.3. Fast JPDA

This scheme was proposed by Fisher and Casasent [9], and also avoids the generation of feasibility matrices. They adopt the same modelling assumptions as the conventional JPDA, but introduce the notion of an analogue modified validation matrix in calculating the β_{jt}'s. The analogue values for the modified validation matrix are the individual probabilities P_{jt}. These values are the same as used in the above mentioned JPDA methods. Their method calculates only T^2 vectors of single observation-target pair probabilities, where T is the maximum number of targets. All possible sums of the last column are formed and the process is repeated for the rest of the columns. A measurement-used binary data word is constructed and used to obtain all possible combinations of track-measurement pairs. The β_{jt} values are then computed by a simple matrix multiplication. For details of this method refer to [9].

3. Simulation Results

A simulation to compare the JPDA algorithms was done in MATLAB. At this stage only non-maneuvering simulated targets were used. This was done for the sake of simplicity. A set of data, similar to that used by Zhou & Bose [7], was created to simulate the targets in a specific window. The tracking was done by using the Cartesian coordinates of the target positions as the observations. It is very easy to convert from these Cartesian values to polar coordinates, used in the North-East-Down (NED) coordinate system. The NED coordinate system is more suitable for use in airborne platforms. A detailed explanation of this coordinate system can be found in Chapter 3 of [2].

A standard Kalman Filter was implemented as the tracking filter. The two states used in the Kalman Filter was position and velocity for the X and Y coordinates. The gate threshold was chosen to be $g^2 = 10$, for simplicity. A probability of detection $P_D = 0.9$ was chosen for each target in all of the simulations. Clutter was assumed to be uniformly distributed over the surveillance region. For the scope of this paper only non-maneuvering target data was generated Minor adjustments to the simulation will enable the ability to include maneuvering targets. For maneuvering targets an extra state for acceleration is needed in the Kalman Filter and the simulation data must include accelerations in certain time intervals. Each simulation run was completed over 50 time steps, k, with the sampling period, T, set to one second. In all of the simulations a specific number of existing target tracks were assumed that was then tracked using the received measurements.

The CPU times in the tables below are the time in seconds for the execution of the various algorithms in the tracking of four and ten targets in limited amounts of clutter.

The following two tables shows the CPU times as a measure of comparison between the three algorithms. The minimum time recorded points towards the case where there is close to only one observation in each validation gate of the individual targets. This greatly reduces the amount of time needed and the total complexity of computing the desired values needed in the algorithm. Maximum time points towards the opposite where much more than one observation falls within the validation gates of the individual targets. The worst case scenario is when all of the measurements fall within every target gate, leading to the maximum amount of computation. The average time used is a good measure of comparison for the three methods explained above. A very effective combinatorial approach of evaluating the feasible events for the conventional JPDA algorithm was implemented for this simulation.

Table 1: Values of the CPU time used in the Data Association algorithms for 4 targets at each time step k. CPU times are in seconds

CPU Time	DFS	Conventional JPDA	PJPDA
Minimum	0.0310	0	0
Maximum	0.0780	0.1250	0.0780
Average	0.0393	0.0147	0.0097

From the values in Table 1 it can clearly be seen that the PJPDA algorithm of Van Wyk *et al.* [3], performs very well in comparison to the other methods. The total time used in this algorithm is almost 4 times faster than the time of the DFS algorithm of Zhou & Bose [7].

In Table 2 the there is a significant difference between the times recorded for the JPDA methods listed in the table. The PJPDA algorithm clearly outperforms the other two proposed methods. Although some problems still exists in the PJPDA algorithm it is clear that solving the problems will yield great advances towards real-time implementation.

The graphical display of the simulations for the two cases is presented on the next page. The Conventional JPDA algorithm and the DFS Algorithm are very similar. The only difference between the two techniques is the method of calculating the feasibility matrices. In the light of the mentioned

similarity, it comes as no surprise that the simulation results for the two methods were found to be very similar. Therefore only the figure for the DFS algorithm is displayed. The main focus of this paper is the CPU times of the algorithms. Each figure displays the values for the real track data (dotted line), the predicted position (solid line), the gate size in relation to the window (circle) as well as clutter points (+).

Table 2: Values of the CPU time used in the Data Association algorithms for 10 targets at each time step *k*. CPU times are in seconds

CPU Time	DFS	Conventional JPDA	PJPDA
Minimum	0.1250	0.0940	0.0150
Maximum	90.75	123.0630	0.0940
Average	18.665	12.6762	0.0373

Listed below are the Graphical display of the JPDA methods when tracking the cases of 4 targets and 10 targets in limited clutter. Figure 1 to 2 is for the case of 4 targets.

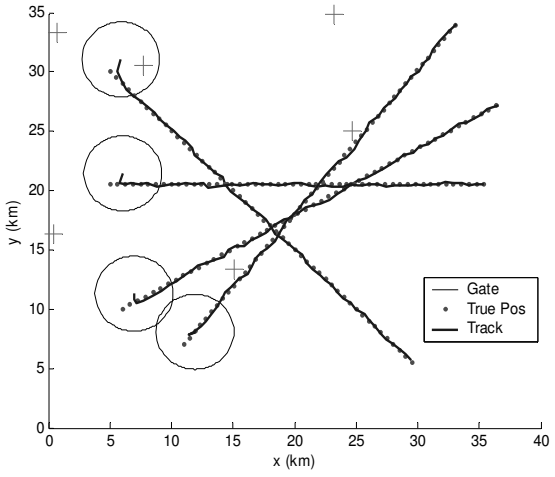


Figure 1: DFS method for 4 targets

In the case of tracking the 4 targets, the results from the three methods closely resemble each other. There is also not even a great difference in the CPU times recorded in Table 1 for the algorithms.

The main difference between the algorithms is apparent when tracking the 10 targets. This is illustrated in Figure 3 to Figure 4. It is clear from Figure 4 that there is a missed association of some sorts when using the PJPDA method. This however is in the process of being solved. From the figures above it is clear that the DFS algorithm yields the best tracking results although having the least favourable CPU times in Table 2.

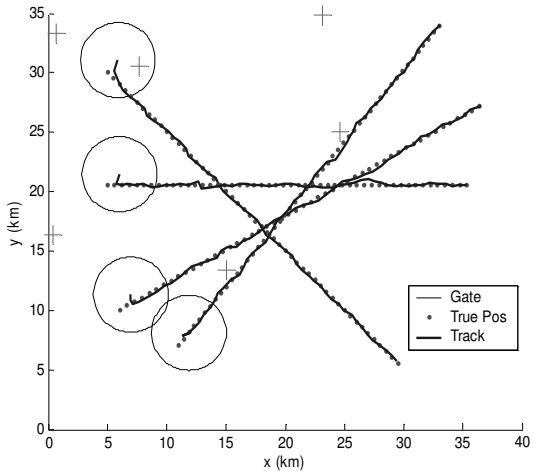


Figure 2: PJPDA method for 4 targets

Figure 3 to 4 is for the case of 10 targets.

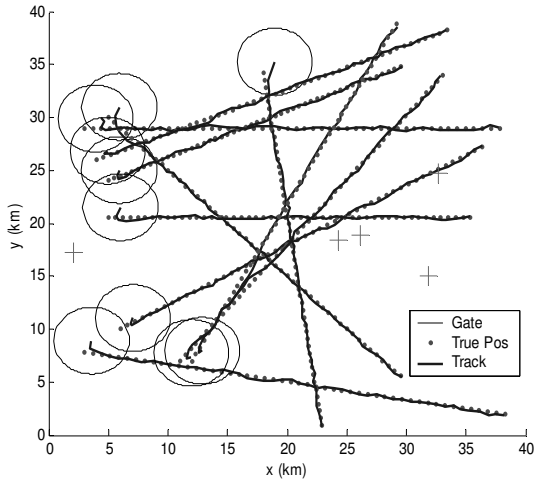


Figure 3: DFS method for 10 targets

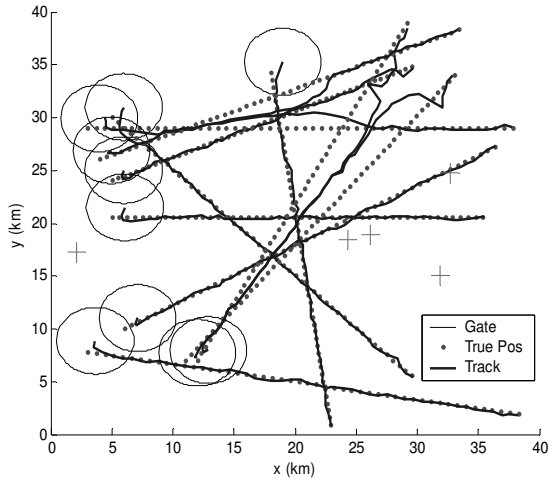


Figure 4: PJPDA method for 10 targets

4. Conclusion

The three algorithms mentioned earlier where compared to each other in a MATLAB simulation. As a measure of effectiveness, CPU time was used to compare them. The PJPDA algorithm, although not entirely sorted out, produced attractive results. In instances of limited clutter and few targets the PJPDA algorithm produced the same results as the DFS algorithm, but at a very low CPU time. It is therefore an attractive option for real-time implementation. Although work is still in progress to sort out some problems with the PJPDA algorithm for dense clutter and large numbers of targets, the algorithm is expected to outperform the other algorithms. With this in mind it is clear that the PJPDA algorithm is a very attractive option for real-time implementation.

5. Acknowledgements

We acknowledge the financial support of Armscor and Denel Aerospace Systems, provided for this project.

6. References

[1] Blackman, S, and Popoli, R, *Design and analysis of Modern Tracking Systems*. Artech House, 1999.

[2] Blackman, S.S., *Multiple-Target Tracking with Radar Applications*, Artech House, 1986.

[3] Van Wyk, B.J., Van Wyk, M.A., Noel, G., "A Projection-Based Joint Probabilistic Data Association Algorithm," *Proceedings of IEEE AFRICON 2004*, Gabarone, Botswana, 15-17 September, pp 13-17, 2004.

[3] Roecker, J.A.; Phillis, G.L., "Suboptimal joint probabilistic data association," *Aerospace and Electronic Systems, IEEE Transactions on*, Volume: 29 Issue: 2, April, pp 510 –517, 1993

[4] Roecker, J.A., "A class of near optimal JPDA algorithms," *Aerospace and Electronic Systems, IEEE Transactions on*, Volume: 30 Issue: 2, April, pp 504 –510, 1994

[5] Rose, K.; Gurewitz, E.; Fox, G., "A nonconvex cost optimization approach to tracking multiple targets," *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications'*, Proceedings, IROS '90, IEEE International Workshop on ,Vol.1, 3-6 July, pp 25 -31, 1990.

[6] Wang Ming-Hui, Peng Ying-Ning, You Zhi-Sheng, "Improved joint probabilistic data association algorithm," *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, Volume: 2, 8-11 July, pp 1602 -1604, 2002.

[7] Zhou, B.; Bose, N.K., "Multitarget tracking in clutter: fast algorithms for data association," *Aerospace and Electronic Systems*, IEEE Transactions on, Volume: 29 Issue: 2, April, pp 352 –363, 1993.

[8] Zhou, B.; Bose, N.K., "An efficient algorithm for data association in multitarget tracking," *Aerospace and Electronic Systems*, IEEE Transactions on, Volume: 31 Issue: 1, January, pp 458 –468, 1995.

[9] Fisher, J. L., and Casasent, D. P., "Fast JPDA multi-target tracking algorithm," *Applied Optics,* Vol. 28, January, pp 371-376, 1989.

[10] Fortmann, T.; Bar-Shalom, Y.; Scheffe, M, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, Vol. 8, pp 173- 184, 1983.

[11] Reid, D., "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, December, pp 843- 854, 1979.

[12] Ding, Z., Leung, H., Hong, L., "Decoupling Joint Probabilistic Data Association Algorithm for Multiple Target Tracking," *IEE Proceedings on Radar, Sonar and Navigation*, Vol. 146, No. 5, pp 251- 254, 1999.

A combining strategy for ill-defined problems

Thomas Landgrebe, David M.J. Tax, Pavel Paclík, Robert P.W. Duin, and Colin Andrew

Elect. Eng., Maths and Comp. Sc.,
Delft University of Technology, The Netherlands
{t.c.w.landgrebe, d.m.j.tax, p.paclik, r.p.w.duin}@ewi.tudelft.nl
colin.andrew@ieee.org

Abstract

In this paper we present a combining strategy to cope with the problem of classification in ill-defined domains. In these cases, even though a particular *target* class may be sampled in a representative manner, an *outlier* class may be poorly sampled, or new *outlier* classes may occur that have not been considered during training. This may have a considerable impact on classification performance. The objective of a classifier in this situation is to utilise all known information in discriminating, and to remain as robust as possible to changing conditions. A classification scheme is presented that deals with this problem, consisting of a sequential combination of a one-class and multi-class classifier. We show that it can outperform the traditional classifier with reject-option scheme, locally selecting/training models for the purpose of optimising the classification and rejection performance.

1. Introduction

Consider a problem in which a *target* class is to be discriminated with respect to an *outlier* class. In many applications, both classes are sampled as a set of measurements in order to construct a training set that represents the class. A classifier can then be designed, for example, by estimating the class conditional densities for both classes. Good estimates allow for an optimal tradeoff to be made between the classes. However in some applications some classes may not be well-defined. In this paper we assume that the *target* class is well represented, but the *outlier* class is not. This may be due to a variation in *outlier* class distribution, such as *sensor drift* [1], or new *outlier* classes may be present that were not represented during training. Examples of this phenomenon include:

- Diagnostic problems in which the objective of the classifier is to identify abnormal operation (*outlier* class) from normal operation (*target* class) [2]. It is often the case that a representative training set can be gathered for the *target* class, but due to the nature of the problem the *outlier* class cannot be sampled in a representative manner. For example in machine fault diagnosis [3] a destructive test for all possible abnormal states may not be feasible.
- Recognition systems often involve a detection and classification stage. An example is road sign classification, in which a classifier needs not only to discriminate between examples of road sign classes, but must also reject non-sign class examples [4]. Gathering a representative set of non-signs may not be possible. Similarly face detection [5], where a classifier must deal with well-defined face classes, and an ill-defined non-face class, as well as handwritten digit recognition [6], where non-digit examples are a serious issue.

The goal of a classifier in these cases is to obtain a high true positive rate and low false positive rate, with respect to the *target* class. Even though the *outlier* class is poorly defined, we would still like to make use of all knowledge that exists for the problem (to account for known class overlaps). Thus the objective is to obtain a high *classification performance*, and robustness to changes in the *outlier* class (referred to as *rejection performance*). Formalisation and consequences of this problem are given in Section 2.

Previous work in this area has typically been the *classifier with reject-option*, first proposed by Chow in [7], often called the *ambiguity reject-option*. In this reject-option, when the cost of misclassification is higher than the cost of rejection, the example in question should be rejected, based on thresholding of the posterior probability. This reject-option is applicable for handling ambiguity between classes (examples close to the *target* class), which is not of interest here. In this paper we are interested in rejecting examples occurring far away from the *target* class. Dubuisson and Masson proposed the *distance reject-option* in [2]. This rejection scheme was designed to cope with the condition in which new classes are present that are not represented during training, introducing a different type of *reject* class ω_r. New examples situated a particular distance (based on a reject threshold t_d) from known class centroids are rejected. A similar procedure can be applied to density-based classifiers, except here the class conditional density is thresholded. In this way a *closed* decision surface is obtained, providing protection against new unseen classes[1]. New classes will be rejected if they fall outside the class description. Thus to minimise the probability of accepting examples from class ω_r, assuming they are uniformly distributed in feature space, the volume of the description should be minimised. The reject-option is discussed further in Section 3.

The limitation of the reject-option approach is that a model chosen for good classification performance does not necessarily imply good rejection performance. The opposite is also true. Improved performance may result from a practitioner viewpoint if an adequate evaluation methodology is used. However as will be discussed later, since the same model is used for classification and rejection, we may have to sacrifice the performance of one for the other. In this paper we present a classification strategy that can in some cases alleviate this situation, consisting of a sequential combination of one-class and multi-class classifiers (called *SOCMC*). The proposed 2-stage scheme allows both rejection and classification performance to be explicitly modified by varying the respective models and representations. Thus a classifier model can be designed to obtain good performance

[1] This thresholding of a single class model is equivalent to one-class classifi cation [8].

on known classes, and a separate classifier model to improve robustness with respect to unknown classes. The SOCMC is discussed in Section 4.

A number of experiments are performed to investigate the SOCMC approach in Section 5. All experiments benchmark SOCMC results with the distance-based reject option, as well as with traditional discriminant-based approaches. Experiments are performed on a number of real datasets, showing the applicability of the new approach. Finally, conclusions are given in Section 6.

2. Ill-defined problems

To formalise this problem, we assume that there is a well defined *target* class ω_t, and the *outlier* class is composed of two classes ω_o and ω_r, where the former consists of known class information, and the latter of unknown information (called the *reject* class). Note that in this setup, we classify examples considered to be either ω_o and ω_r as *outlier*. Examples of each class are composed of vectors of measurements $\mathbf{x}$ with dimensionality d. It is assumed that $\mathbf{x}$ is represented by a feature space χ (later we discuss classifiers that operate on the data in new feature spaces, consisting of various mappings of the original space χ). The unconditional density $p(\mathbf{x})$ can then be written as in Equation 1.

$$p(\mathbf{x}) = p(\omega_t)p(\mathbf{x}|\omega_t) + p(\omega_o)p(\mathbf{x}|\omega_o) + p(\omega_r)p(\mathbf{x}|\omega_r) \quad (1)$$

To evaluate classifiers in this situations, two performance measures are of interest:

1. The classification performance (performance between known classes/data), denoted perf(ω_t, ω_o).
2. The rejection performance (performance between the ω_t and ω_r), denoted perf(ω_t, ω_r).

Ideally both perf(ω_t, ω_o) and perf(ω_t, ω_r) should be high. Note that estimation of perf(ω_t, ω_r) is not straightforward, since this class is by definition absent during training. In the experimental Section 5 a methodology is given to provide some estimate of this. In Figure 1 an example of this problem is shown, demonstrating the weakness of general discrimination approaches with respect to this problem. Here a synthetic dataset has been constructed in two dimensions. In the left image, the training set consisting of ω_t and ω_o is shown, upon which a Bayes quadratic classifier is shown. In the right image the testing situation is shown, in which a new class ω_r is present. The classifier is clearly not robust to these changes in conditions.

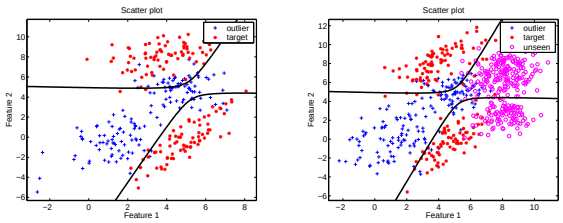


Figure 1: Illustrating a discrimination classifier applied to the problem in which a new unseen class is present in testing. The left plot shows the classifier decision boundary for the training data, and the right plot for the testing data, in which a new class ω_r is present. A Bayes quadratic classifier is used.

Two classification approaches are utilised in this paper. The first are multi-class classifiers (sometimes referred to as MCC's/discriminators). In this paper, we deal specifically with two-class discriminant classifiers, denoted D_{MCC}. A classifier trained on ω_t and ω_o can be defined as in Equation 2, with $\hat{p}(\omega|\mathbf{x})$ representing an estimate of the posterior probability of class ω. These classifiers result in an open decision boundary, since it is assumed that ω_t and ω_o are well represented.

$$D_{MCC} : \begin{cases} \textit{target} & \text{if } \hat{p}(\omega_t|\mathbf{x}) > \hat{p}(\omega_o|\mathbf{x}) \\ \textit{outlier} & \text{otherwise} \end{cases} \quad (2)$$

The second classification approach used is one-class classification (sometimes referred to as OCC's), denoted D_{OCC} [8]. These classifiers are trained on only a single class, resulting in a closed description of the class density or domain. No assumptions about other classes are made, and thus these classifiers do not make a trade-off between overlapping classes. The decision boundary is however constrained/closed, i.e. all objects situated outside the class description are rejected as outliers, providing protection against new, unseen classes. The OCC description/model is trained, with some allowance made for outliers in the training set by adjusting a decision threshold θ. The D_{OCC} can be written as in Equation 3, classifying all objects as either *target* or *outlier*.

$$D_{OCC} : \begin{cases} \textit{target} & \text{if } \hat{p}(\mathbf{x}|\omega_t) > \theta \\ \textit{outlier} & \text{otherwise} \end{cases} \quad (3)$$

3. The classifier with reject-option

As previously mentioned, the original reject option (*ambiguity* reject) [7] rejects objects that are considered to be ambiguous, based on a threshold t_d. For an incoming test object, the classifier assigns a class label. The relevant posterior of the assigned class is examined and compared to t_d. Examples are either assigned to an $ACCEPT$ region $\Re_{accept}$ or $REJECT$ region $\Re_{reject}$, as shown in Equation 4.

$$\begin{aligned} \Re_{accept} &= \{\mathbf{x}|\max_i p(\omega_i|\mathbf{x}) \geq t_d\},\ i \in \{t, o\} \\ \Re_{reject} &= \{\mathbf{x}|\max_i p(\omega_i|\mathbf{x}) < t_d\},\ i \in \{t, o\} \end{aligned} \quad (4)$$

With the *distance reject option*, the conditional density of the class of interest is thresholded, resulting in a *closed* decision boundary[2], providing protection against unseen classes. Again a two-stage procedure is undertaken. In the first stage an example is assigned to a particular class $\omega_i, i = t, o$, referring to *target* and *outlier*, using Bayes rule. In the second step, if the example has been assigned to the *target* class, the conditional probability $p(\mathbf{x}|\omega_t)$ is thresholded via a reject threshold t_d. Examples exceeding this threshold are rejected. Examples are either assigned to an $ACCEPT$ or $REJECT$ region, $\Re_{accept}$ and $\Re_{reject}$ as shown in Equation 5.

$$\begin{aligned} \Re_{accept} &= \{\mathbf{x}|p(\mathbf{x}|\omega_i) \geq t_d\},\ i \in \{t, o\} \\ \Re_{reject} &= \{\mathbf{x}|p(\mathbf{x}|\omega_i) < t_d\},\ i \in \{t, o\} \end{aligned} \quad (5)$$

The *distance* reject option is illustrated on a simple example in Figure 2. The left plot shows a model based on a linear classifier, and the right image a mixture-of-Gaussians classifier with 15 mixtures. It is clear that a closed boundary results, and the

[2]For classifiers that are not density-based such as k-Nearest Neighbour, Dubuisson and Masson proposed to reject based on the mean distance to the k nearest neighbours. In this case a meaningful threshold should be chosen based on the scale of the distances.

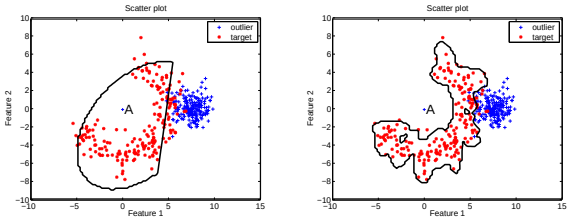


Figure 2: Illustrating the *distance* reject option classifier in a two-class 2D example, showing a linear classifier model in the left plot which results in good classification performance, but poor rejection performance. The right image depicts a more complex mixture-of-Gaussians classifier, resulting in good rejection, but poor classification. The decision boundary indicates the threshold used for class assignment. Poor models have purposefully been chosen for the sake of illustration to simulate realistic conditions.

trade-off between known classes is accounted for. We discuss two situations that could lead to sub-optimal performance.

In the first situation we discuss the practitioner. If the practitioner designs a classifier based on knowledge of the ω_t and ω_o classes only, a situation such as that depicted in the left figure may result. Here the classifier obtains near optimal classification performance, but since the model is not chosen explicitly to fit the *target* class distribution, sub-optimal rejection results. Thus we propose to evaluate classifiers in these situations based on both classification and rejection performance. This may lead to choosing more appropriate models. For example some classifiers focus on discrimination only, discarding domain information (e.g. support-vector classifier). A better choice would be to choose models modeling the distribution (e.g. mixture-of-Gaussians density estimation).

In the second situation we assume the practitioner is aware of an adequate evaluation methodology. In this case the practitioner will focus on obtaining the best-possible rejection/classification performance. In real problems, typical limitations are that the training set size is limited, the input dimensionality high, and computation time limited. In these situations, choosing a model that results in high classification performance (i.e. focus on known overlapping regions) may be at the expense of a worse performance in terms of rejection performance e.g. the class conditional density may be well estimated in the overlapping region, but poor in other areas. This is depicted in the left plot of Figure 2 where a new *outlier* example marked *A* on the plot will be incorrectly classified as *target*. Similarly, the classification performance may be compromised for the case in which a model is chosen for good rejection performance (right plot). In Section 4 a classification scheme is presented in which different models can be selected/trained explicitly for classification and rejection respectively. We argue that in some cases it is better to choose a local model suitable to perform the classification, and another for rejection. This flexibility is lacking in the reject-option case.

4. Sequential combining of a one-class and multi-class classifier

We present a classification scheme here consisting of the sequential combining of one-class and multi-class classifiers (SOCMC). The rationale is that the class model and representation used in the first stage (denoted D_{OCC}) can be explicitly chosen for the purpose of rejection i.e. between ω_t and ω_r. In a similar way the second stage classifier D_{MCC} can be chosen locally in the area of known overlap to obtain good classification performance between known classes, i.e. between ω_t and ω_o. The SOCMC classifier is depicted in the block diagram in Figure 3. In the first stage, the one-class classifier D_{OCC} attempts

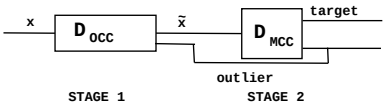


Figure 3: Block diagram of the *SOCMC* classifier. The first stage classifier D_{OCC} consists of an OCC, trained on the well defined *target* class. The second stage classifier D_{MCC} is a multi-class discriminant trained on examples considered to be *target* by the first stage.

to detect all *target* examples from $p(x)$, given a test set. A one-class classifier [8] is appropriate for this stage since it protects against unseen classes ω_r, and capitalises on the knowledge that the *target* class is well defined by the training set[3]. At this stage it is not important if examples of the class ω_o are incorrectly accepted, since we rely on this discrimination in the second stage. Thus it is assumed that the output of D_{OCC}, denoted $\tilde{\mathbf{x}}$ will consist only of examples of class ω_t and ω_o, with all ω_r having been rejected (as well as ω_o examples that do not overlap with the *target* OCC description.).

Note that both the representation and class description model can be selected/trained to improve the rejection performance perf(ω_t, ω_r). The D_{OCC} represents the input data $\mathbf{x}$, derived from the feature space χ, by a new representation χ_{OCC} (D_{OCC} consists of both a representation and classification stage). The classifier can thus be written as $D_{OCC}(\mathbf{x}_{OCC})$, defined as in Equation 3 for class ω_t. The output $\tilde{\mathbf{x}}$ is then shown in Equation 6.

$$\tilde{\mathbf{x}} = \{\mathbf{x} | D_{OCC}(\mathbf{x}_{OCC}) = \textit{target}\} \tag{6}$$

The output $\tilde{\mathbf{x}}$ is then applied to the second stage classifier D_{MCC}. Note that D_{OCC} is used to select objects for the second stage. We still have the opportunity to optimise the representation and model selection used in the second stage. Thus $\tilde{\mathbf{x}}$ is used by D_{MCC} in the original representation χ. The D_{MCC} classifier is trained on the data $\tilde{\mathbf{x}}$, which is assumed to be a mixture of data from ω_t and ω_o only, which are represented by the training set. A discriminator is thus trained, with the objective of obtaining an optimal trade-off in terms of class overlap. As with the D_{OCC}, the representation and classification model can be chosen, but in this case for the purpose of optimising the classification performance perf(ω_t, ω_o). A model is trained focused on the local region, specified by a training dataset $(\tilde{\mathbf{x}})_{tr}$. The input data $\tilde{\mathbf{x}}$ that is represented by χ is now mapped to a new representation space χ_{MCC}, resulting in the classifier $D_{MCC}(\mathbf{x}_{\tilde{MCC}})$, defined as in Equation 2 between classes ω_t and ω_o. The final *SOCMC* classifier, denoted D_{SOCMC} is de-

[3]New classes will be rejected if they fall outside the class description. Thus to minimise the probability of accepting examples from class ω_r, assuming they are uniformly distributed, the volume of the description should be minimised.

fined in Equation 7.

$$D_{SOCMC}(\mathbf{x}|D_{OCC}, D_{MCC}) = \begin{cases} \textit{outlier} \text{ if } D_{OCC}(\mathbf{x}) = \textit{outlier} \\ D_{MCC}(\tilde{\mathbf{x}}_{MCC}) \text{ otherwise} \end{cases} \quad (7)$$

We illustrate the operation of the *SOCMC* classifier in the same situation as in Section 3, in Figure 4. We noticed that the classifier D1 in the left plot of Figure 2 resulted in high classification performance, and low rejection performance. The opposite was true for the classifier D2 in the right plot. In the SOCMC classifier, we select/train specific local models for the purposes of classification and rejection respectively, illustrating that the SOCMC can in some cases improve performance. In this example the model used for D1 is chosen for the D_{MCC} stage (i.e. a linear classifier), and the D2 model is used for the D_{OCC} stage (Mixture-of-Gaussians with 6 mixtures). This classifier results in a good classification and rejection performance. A number of

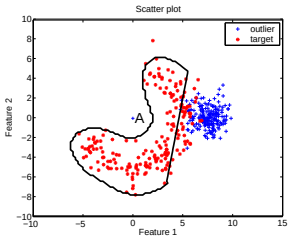


Figure 4: Illustrating the *SOCMC* classifier. A linear model has been chosen locally in the area of overlap for good classification performance, and a Gaussian-mixture model with 15 mixtures is used for rejection, showing that example *A* is correctly classified.

training considerations need to be made for the *SOCMC* classifier:

- Training set size for D_{MCC}: If a training set $\mathbf{x}_{tr}$ is used, only a subset $\tilde{\mathbf{x}}_{tr}$ will be available for the D_{OCC}. If $\mathbf{x}_{tr}$ is small, or $\tilde{\mathbf{x}}_{tr} \ll \mathbf{x}_{tr}$, there may not be sufficient samples to train D_{MCC}. This may limit the complexity of the model/representation used. Alternatively the entire $\mathbf{x}_{tr}$ could be used to train the D_{MCC}.
- Training technique: The *SOCMC* classifier is analogous to the trained combiner used in classifier combining, as discussed in [9]. If the same training set $\mathbf{x}_{tr}$ is used to train both D_{OCC} and D_{MCC}, the D_{SOCMC} could overfit to the noise in the training set. An alternative training strategy could be to split $\mathbf{x}_{tr}$ into two independent training sets $\mathbf{x}_{tr1}$ and $\mathbf{x}_{tr2}$, with the first used to train D_{OCC}, and the second used to train D_{MCC}. This may generalise better, but may actually be worse than the former strategy when the training set size is small.

5. Experiments

5.1. Evaluation

Since the exact nature of the *outlier* conditional distribution cannot be predicted in advance, estimating perf(ω_t, ω_r) is not straight forward. We propose an evaluation method to provide some confidence as to the robustness of the classifier, and to compare classifiers. The evaluation assumes a uniform *outlier* distribution. This test allows a classifier to be evaluated assuming that *outlier* examples can occur anywhere in feature space around the *target* class. It provides a measure for how well the classifier protects the *target* class (in the respective feature space) from changing conditions. However for real high dimensional problems, the number of artificial examples to be generated may be computationally prohibitive, so two methods of artificial data generation are used in real experiments to attempt to overcome this problem:

1. In the first method, called perf$_{a1}(\omega_t, \omega_r)$, a number of *outlier* examples are artificially generated uniformly in a sphere around a subspace of the *target* class [10]. Here examples are generated within a PCA (Principal Component Analysis) subspace. The original data is scaled to unit variance, and the artificial data is then generated within this space with a radius of 1.1 of the covariance of the *target* class. These can be mapped into the original space by an inverse of the PCA mapping.
2. Similar to the previous analysis, except data is generated in the original representation, following a Gaussian distribution. Here examples are generated around the *target* class, using an enlarged covariance matrix of the *target* class. The covariance matrix is enlarged by a fraction of 1.5 (this is simply a multiplication of the covariance matrix to spread the new generated examples further). The test is called perf$_{a2}(\omega_t, \omega_r)$.

The perf(ω_t, ω_o) measure relates to the known classes ω_t and ω_o. This performance is approximated using standard techniques. For all experiments a 20-fold cross-validation procedure is carried out, and the primary performance measure used is the AUC (Area under the Receiver-Operator Curve). The variance of the estimates is depicted in terms of the standard deviation. To summarise, the following performance measures are computed for each experiment:

- perf(ω_t, ω_o), estimated using cross-validation with 20-folds, computing the respective AUC.
- perf$_{a1}(\omega_t, \omega_r)$, estimated using 20-fold cross-validation procedure. In testing, for each fold an independent *target* portion of $\mathbf{x}$ is used, together with the generated artificial *outlier* data that was not used for training. Again the AUC is computed.
- perf$_{a2}(\omega_t, \omega_r)$, estimated as per perf$_{a1}(\omega_t, \omega_r)$.

5.2. Dataset description

A number of real-world datasets are used in the experimentation. These datasets have been selected based on their relevance to this problem. The following datasets are used:

1. *Face-amsterdam (Face)*: This dataset consists of a face class ω_t, and non-face class ω_o, and is described in [5], and downloaded at [11]. Each face is stored as a 20×20 image. Only the first 1000 faces from the face database, and the first 1000 non-faces from the non-face test database are used. This dataset is used because it can be argued that finding a representative set of non-face examples may be infeasible.
2. *Mfeat-Fou Digit4 (Mfeat)*: This is a dataset consisting of examples of ten handwritten digits, which can be found in [12]. In this dataset, Fourier components have been extracted from the original images, resulting in a 76-dimensional representation of each digit. 200 examples of each digit are available. In these experiments, digit *4* is used as the *target* class, and all the others as *outlier*.

3. *Geophysical (Geo)*: A multi-modal dataset, in which a *target* and *outlier* class are represented by spectra. In this problem, new *outlier* classes may appear during testing. 3982 *target* examples exist, and 3675 *outlier* examples.

5.3. Results

The results for a number of experiments on the real-world datasets are now presented. The objective of the experiments is to assess the SOCMC classifiers on the real-world problems to ascertain whether they do in fact outperform conventional discriminant-based classifiers. This paper also shows that the SOCMC classifiers can result in higher performance than the distance-based reject-option classifiers. In each experiment, SOCMC results are shown benchmarked against discriminant and reject-option classifiers. For a fair comparison, the same model and representation used for the discriminant classifier is used in the reject-option classifier, and also in the multi-class stage D_{MCC} of the SOCMC. A number of different D_{OCC} models are then chosen to attempt to improve the rejection performance perf(ω_t, ω_r), with only the best results shown for brevity (there are examples where SOCMC classifiers do not work – some optimisation is required is to select appropriate models). The reject threshold for the reject-option and one-class classifiers is fixed to reject 5.00% of *target* examples for the given training set. As a starting point for the comparison, it is important to note that the SOCMC classifier results in a similar performance to the reject-option classifier when the same model (i.e. same representation and data model) is used for both the D_{OCC} and D_{MCC} results. Small differences in results are attributed to the fact that only a subset of $\mathbf{x}$ is used to train the D_{MCC}. These results are not included due to space constraints.

In Table 1 details of each experiment are shown. The first column indicates the dataset used, and the second column the model used for the discriminant classifier **M**, the reject-option classifier **R** and the D_{MCC} stage of the SOCMC classifier **S**. The last column shows the representation and classifier used for the D_{OCC} of the SOCMC. For each classifier, three performance results are shown (in terms of mean AUC[4] over 20-folds with standard deviations shown). These consist of the perf(ω_t, ω_o), perf$_{a1}(\omega_t, \omega_o)$ and perf$_{a2}(\omega_t, \omega_r)$ measures, denoted clf, $rj1$, and $rj2$ respectively. Ideally, all three performances should approach 1.00.

First we discuss the *face* results in Figure 5. In the first experiment *face A*, it can be seen that the discriminant classifier **MA** has a rejection performance ($rj1$ and $rj2$) that is much lower than the classification performance clf. This is attributed to the fact that the *target* decision space is unconstrained, providing little protection against changing conditions. The reject-option classifier **RA** then shows a marked improvement in rejection performance in terms of test $rj1$, with a small decrease in clf. This sacrifice of classification performance for improved rejection performance alludes to a tradeoff between these two measures. The poor performance on $rj2$ was unexpected at first, but on closer inspection of the model used (QDC), which assumes unimodality, and the fact that data generated in the $rj2$ test is also distributed in a uniform manner only in the region of the *target* class, may provide an adequate explanation. These results only show marginal (but significant at times) improvements of the SOCMC classifier over the reject-option. It is suspected that this dataset is largely unimodal, and close to Gaussian-distributed (and the outliers in $rj2$ are generated in a similar fashion). In the first experiment, the **SA** performances in terms of clf and $rj1$ are slightly better than **RA**. In the second experiment *face B*, **SB** results in a much higher rejection performance than **RB**, but with some loss in classification performance. Again we observe a trade-off between classification and rejection performance. The third experiment once again shows small improvements over the reject-option with respect to **SC**.

[4] where an ideal performance in a separable problem would result in an AUC score of 1.

Dataset	Base algorithm	SOCMC D_{OCC} model
Face A	PCA 0.99 QDC	PCA 0.99 Gauss
Face B	Fisher-map QDC	PCA 0.99 MoG-8
Face C	PCA 0.99 LDC	PCA 0.99 Gauss
Mfeat A	Nearest-mean	Gauss
Geo A	PCA 0.9 QDC	PCA 0.9 MoG-5
Geo B	PCA 0.999 QDC	PCA 0.999 MoG-5
Geo C	PCA 0.9 MoG-5/class	PCA 0.999 MoG-5

Table 1: Description of experiments. The first column shows the dataset used. In the second column the algorithm used for the discriminant classifier **M**, the reject-option classifier **R** and the D_{MCC} stage of the SOCMC classifier **S** is given. The last column shows the representation and classifier used for the D_{OCC} of the SOCMC. PCA is a principal component analysis mapping, followed by the percentage of retained variance. Gauss is a Gaussian model. MoG-N is a Mixture-of-Gaussians model with N mixtures. LDC and QDC are Bayes linear and quadratic classifiers respectively.

In the left-most plot of Figure 6, the results of the *mfeat-fou digit4* experiments are shown. Here a nearest-mean classifier has been used, resulting in a 92.44% AUC classification performance for **MA**. The rejection performances are however around 50.00%. The reject-option classifier **RA** is not significantly better than **MA** at rejection. In this case a large number the outliers generated were accepted by a clearly sub-optimal rejection model, even though the classification performance is high. However the SOCMC classifier performs much better here. Even though a nearest-mean classifier is used for classification, the Gaussian model is much better at rejection. Low performances on $rj2$ suggest again that the *target* data is unimodal, with most generated *outlier* examples falling within the domain of the *target* class.

In the three right-most plots in Figure 6, the results of the *geophysical* experiments are shown, showing considerable improvements achieved by the SOCMC scheme. In *Geo A* it can be seen that both *RA* and *SA* improve in terms of $rj1$ performance. However the SOCMC is much better at $rj2$ performance. In this case, the D_{OCC} model used was a Mixture-of-Gaussians, that could model the apparent multi-modality of the *target* class, and thus provide better protection against the *outlier* examples generated in $rj2$. The reject-option rejection model was constrained to the unimodal QDC. In the second experiment *Geo B*, a good example of the SOCMC approach is shown (see **RB** and **SB**), with a clear performance improvement over the reject option. The third experiment shows that the SOCMC and reject option classifiers result in a similar performance, with a slightly better $rj2$ performance achieved by the SOCMC. We conclude that a strong classification model (fitting the data well) will result in optimal classification and reject performance. It was observed that a discriminator can indeed obtain high classification performance, but a model chosen for good classification performance can be at the expense of rejec-

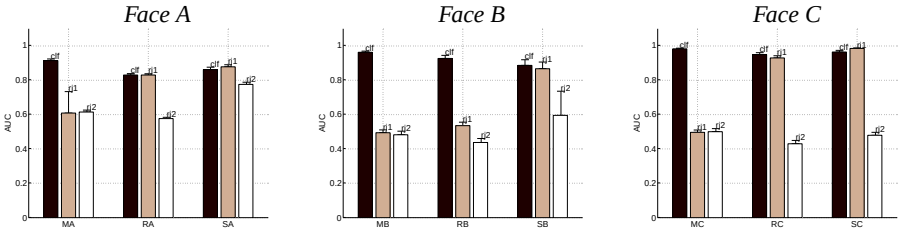


Figure 5: Summarised results of the *face-amsterdam* experiments.

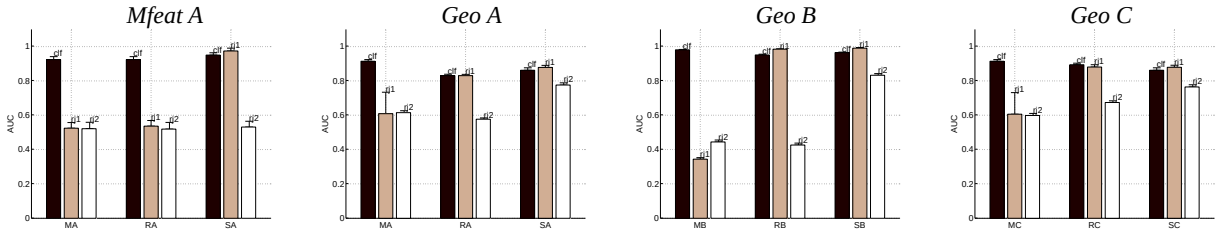


Figure 6: Summarised results of the *mfeat-fou digit4* and *geophysical* experiments.

tion performance. The SOCMC results showed that this classifier can improve upon the reject-option, with separate models trained locally for the purposes of classification and rejection respectively.

6. Conclusions

In this paper classification strategies for ill-defined problems was discussed. It was assumed that a well defined *target* class is to be discriminated from an ill-defined *outlier* class. First the implications on performance with respect to standard discrimination approaches was discussed, showing that a closed/constrained decision space around the *target* class is necessary for robustness to changing conditions. The state-of-the art classifier suited to this task is the distance-based reject option. It was pointed out that a practitioner should make use of an adequate evaluation methodology in selecting a classifier, considering both classification and rejection performance. A new classification strategy was proposed for these types of problems, involving the sequential combination of one-class and multi-class classifiers. These classifiers allow a model to be explicitly selected/trained in local regions of known overlap to emphasise either classification or rejection performance. Experimentation on a number of real-world datasets showed that in some cases the SOCMC classifier does indeed outperform the distance-based reject-option approach. An observation made during experimentation is that an inherent trade-off occurs between classification and rejection. Optimising this will be a focus of future research.

7. Acknowledgments

This research is/was supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.

8. References

[1] K. Copsey and A. Webb, "Classifier design for population and sensor drift," *Joint IAPR Workshops on Syntactical and Structural Pattern Recognition, and Statistical Pattern Recognition*, pp. 744–752, August 2004.

[2] B. Dubuisson and M. Masson, "A statistical decision rule with incomplete knowledge about classes," *Pattern Recognition*, vol. 26, no. 1, pp. 155–165, 1993.

[3] A. Ypma, D.M.J. Tax, and R.P.W. Duin, "Robust machine fault detection with independent component analysis and support vector data description," *Proceedings of the 1999 IEEE Workshop on Neural Networks for Signal Processing, Madison*, 1999.

[4] P. Paclík, "Building road sign classifiers," *PhD thesis, CTU Prague, Czech Republic*, 2004, To appear.

[5] T. V. Pham, M. Worring, and A. W. M. Smeulders, "Face detection by aggregated bayesian network classifiers," *Pattern Recognition Letters*, vol. 23, no. 4, pp. 451–461, February 2002.

[6] C.L. Liu, H. Sako, and H. Fujisawa, "Performance evaluation of pattern classifiers for handwritten character recognition," *International Journal on Document Analysis and Recognition*, pp. 191–204, 2002.

[7] C.K. Chow, "On optimum error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. It-16, no. 1, pp. 41–46, 1970.

[8] D.M.J. Tax, "One-class classification," *PhD thesis, TU Delft, The Netherlands*, June 2001.

[9] R.P.W. Duin, "The combining classifier: To train or not to train?," *ICPR16, Proceedings 16th International Conference on Pattern Recognition (Quebec City, Canada), IEEE Computer Society Press, Los Alamitos*, vol. 2, pp. 765–770, August 2002.

[10] D.M.J. Tax and R.P.W. Duin, "Uniform object generation for optimizing one-class classifiers," *Journal for Machine Learning Research*, pp. 155–173, 2001.

[11] T. V. Pham, M. Worring, and A. W. M. Smeulders, "Face database," http://carol.wins.uva.nl/ vietp/publication/list.html.

[12] "Mfeat," *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/*.

A comparison of three class separability measures

L.S Mthembu & J.Greene

Department of Electrical Engineering, University of Cape Town
Rondebosch, 7001, South Africa.

lsmthlin007@mail.uct.ac.za & jrgreene@eng.uct.ac.za

Abstract

Measures of class separability can provide valuable insights into data, and suggest promising classification algorithms and approaches in data mining. We compare three simple class separability measures used in supervised machine learning.

Their relative effectiveness is evaluated through their functional relationships and their random projections of data onto R^2 for visualization.

We conclude that the simple direct class separability measure of a dataset is an easier and more informative measure for separability than the class scatter matrices approach and it correlates well with Thornton's Separability's index.

1. Introduction

In exploratory analysis of data, simple and rapidly computable global measures such as class separability can give insights into the data and provide pointers towards the choice of classifier.

Given any dataset, one would like to know how separable the classes are before choosing a particular classifier. For low dimensional datasets (≤3) we can view the class scatter.

Unfortunately most real world datasets have more than three dimensions – furthermore we would like to automate this procedure by minimizing user-chosen free parameters in all the measures.

We compare the following three data dependent class separability measures:

1) Class Scatter Matrices (CSM)
2) Thornton's Separability index (Sepindex, SI)
3) Direct Class Separability measure (DCSM)

The class scatter matrices [1] approach is a well-known and widely used measure (particularly in the context of clustering). However this measure aggregates cluster separation into a measure based on the separation of means and thus all class distribution information is lost.

In the previous paper [2], Thornton's SI was shown to be an effective measure of class separability, well suited to feature selection in nearest neighbour and kernel classifiers.

It is possible to define a direct measure of separability in which mean distance is replaced by summation of individual pairwise distances.

The present paper examines the hypothesis that such a measure, retaining as it does distribution information may be more informative than the class scatter matrix measure. We call such a measure the direct class separability measure (DCSM).

1) **Class scatter matrices/measure** (**CSM**) for class separability is an old technique. It is defined as:

$$S_b = \sum_{i=1}^{c} (m_i - m)^t (m_i - m)$$

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{ni} (x_{ij} - m_i)^t (x_{ij} - m_i)$$

where: *c= number of classes, ni = number of instances in class i.* m_i *is the mean of instances in class i and* m *is the mean of all classes.* x_{ij} *is the jth instance in class i.* S_b *is the between class scatter matrix and* S_w *is the within class scatter matrix.*

$$J = \text{trace}(S_B) / \text{trace}(S_w)$$

J is an unbounded measure. The larger the value of *J* the smaller the within class scatter as compared to the between class scatter.

2) **Separability Index** (**SI**) as defined in [2]:

$$SI = \frac{\sum_{i=1}^{n} (f(x_i) + f(x_i') + 1) \bmod 2}{n}$$

calculates the average number of instances that share the same class label as their nearest neighbours. The performance of Thornton's separability index has been previously demonstrated in [2]. We thus report on the functional relationship of the other two separability measures versus this index.

3) We define the **Direct class separability measure, DCSM** to be:

$$S_w = \sum_{i=1}^{ni} \sum_{i=1}^{ni} \left\| x_i - x_j \right\|$$

$$S_b = \sum_{i=1}^{ni} \sum_{j=1}^{nj} \left\| x_i - x_j \right\|$$

where ni & nj = number of instances in class i & j respectively and x_i & x_j are the instances.

$$DCSM = [S_B - S_w].$$

S_B is the between class distances
S_w is the within class distances

This measure directly measures how compact each class is as compared to how far it is from the other class.

If for one dataset, $S_B < S_{w-}$ and $S_B > S_{w+}$ then the scatter of the negative class is more than the scatter between it and the positive class. Further more, the negative class overlaps the positive class.

One way of comparing correlation between separability measures is via feature selection. This is presented in section 2 of this paper.

To further explore the differences between these measures section 3 presents the all measures' random projections of a number of datasets onto two-dimensional space. Section 4 presents conclusions of the paper.

2. Functional Relationships

We calculate each measure on all 2^d-1 feature subsets of each dataset, where *d* is the number of features in a dataset. We then plot the value of each separability measure versus the value of Thornton's separability index.

We make use of the Wisconsin Breast -Cancer and Liver datasets from [3], the Ljubljana Breast-Cancer and Thyroid from [4]. We arbitrarily used realization 13 on the datasets from [5].

Comments on functional relationship graphs

The class scatter matrices (CSM) vs. SI figures show that CSM does not have a clear functional relationship with Thornton's separability index.

When the class scatter matrices measure has a feature set that produces the best class separability, the SI does not. This is different when we compare the plots of the direct class separability measure (DCSM) graphs.

It is found that there is a clearer correlation between the direct class separability measure and Thornton's separability index than there is with the class scatter matrices; furthermore one of the classes in the DCS measure can have an inverse (negative slope) relation with SI. This is additional information given by using this measure.

The definition of DCSM means we will generate two graphs of this measure for each dataset. The first graph, DCSM vs. SI S_w+, for example, shows how the measure varies, for the within class scatter distances (S_w+) of the positive class for different feature combinations.

When the slopes of the relationship between DCSM and SI for the positive and negative classes are the same, the classes are easily separable. This separability *can* be in the form of multi-clusters within each class (multimodal) and or uni-modal (each class being one compact cluster).

This results from the fact that the distances between the positive class instances are smaller than the distances between the positive class instances are from the negative class instances and the distances between the negative class instances are smaller than the distances between the negative class instances are from the positive.

When the slopes of the relationship between DCSM and SI for the positive and negative classes are different, one of the classes is overlapping the other. This is due to the fact that the one class has within class distances that are larger than its instances are from the opposing class's instances.
The class scatter matrices approach does not explicitly tell us this information.

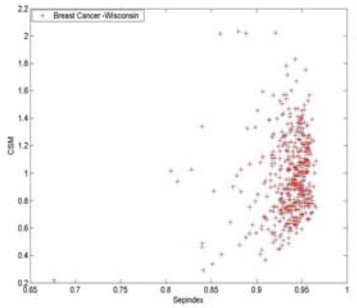


Figure 2.1 CSM vs. SI (*B-Cancer Wisconsin*)

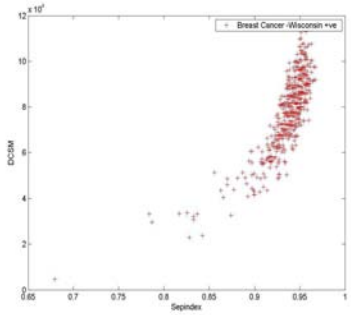


Figure 2.2. DCSM vs. SI (S_w+) (*B-Cancer Wisconsin*)

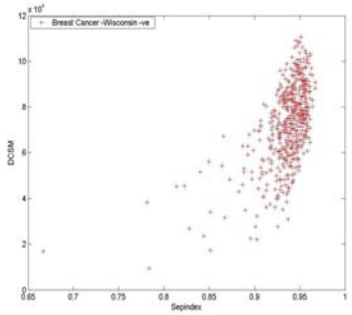


Figure 2.3 DCSM vs. SI (S_w-) (*B-Cancer Wisconsin*)

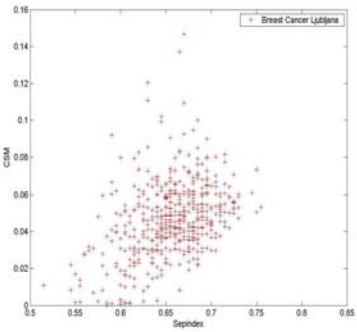


Figure 2. 4 CSM vs. SI (*B-Cancer Ljubljana)*

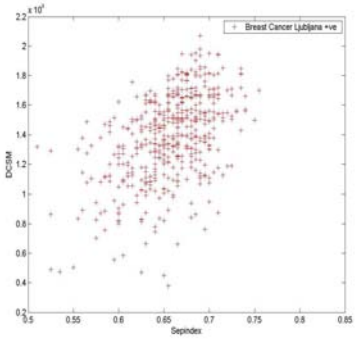


Figure 2.5. DCSM vs. SI (S_w+) (*B-Cancer -Ljubljana*)

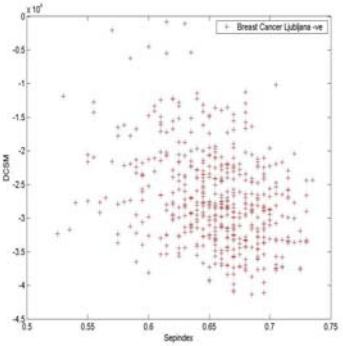


Figure 2.6. DCSM vs. SI (S_w-) (*B-Cancer Ljubljana*)

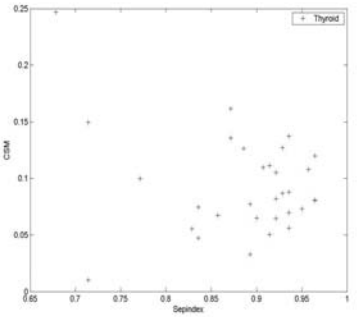


Figure 2.7. CSM vs. SI (*Thyroid*)

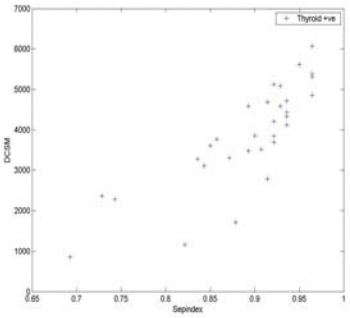


Figure 2.8. DCSM vs. SI (S_w+) (*Thyroid*)

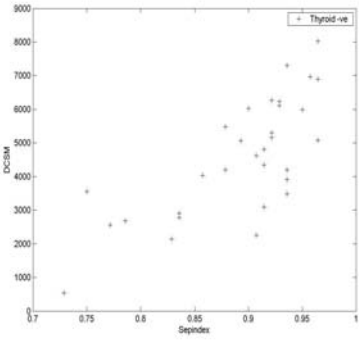


Figure 2.9 DCSM vs. SI (S_w-) (*Thyroid*)

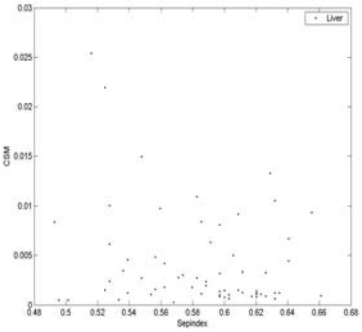


Figure 2.10. CSM vs.SI (*Liver*)

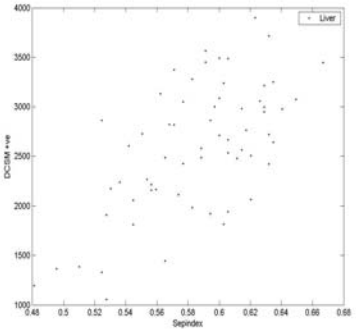


Figure 2.11. DCSM vs. SI (S_w+) (*Liver*)

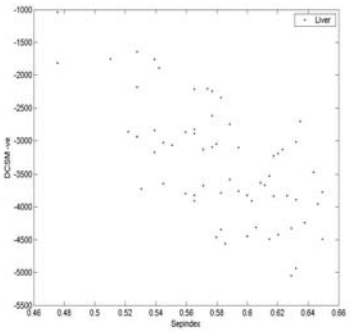


Figure 2.12.DCSM vs. SI (S_w-) (*Liver*)

3. 2D Projections of Datasets using the above measures to maximize separability

We project the full dataset onto a 2 dimensional space by 2 random vectors: Given an X = [*pxd*] observation matrix, we multiply it by a random matrix R= [*dx2*]. This projects the *d* dimensional data onto 2 dimensional space.

We generate 100 random (dx2) vectors and plot the graph that maximizes the separability measure in question.

Comments on random projection figures

The projection graphs confirm the results from the functional relationship graphs of the previous section.

When the classes in the dataset are distinct from each other the DCSM vs. SI functional relationship slopes remain the same for both classes (e.g. Wisconsin B-Cancer and Thyroid). When the slopes differ the classes in the dataset are not easily separable due to classes overlapping (e.g. Liver and Ljubljana B-Cancer).

This measure in effect tells us before hand how our classes are possibly distributed in relation to one another.

Thornton's separability index also tells us about how much class overlap there is; the more overlapping between the classes the more instances will have nearest neighbours of a different class, resulting in a low SI.

Interestingly both the above measures do not explicitly tell us whether the classes are multimodal or just uni-modal but only tell us the degree of overlap in the classes.

It is thus not surprising then that the direct class separability measure's projections are similar to Thornton's Separability index's projections of each dataset

Figures 3.8 to 3.9 show the multimodality of the Thyroid data while figure 3.7 does not clearly show this structure in the data. This is because the CSM aggregates the instances and their classes thus the information of the diversity of the data structure is lost.
All three separability measures are not able to produce projections of separable classes for the Liver and the Ljubljana breast cancer datasets. This was alluded to by the low SI index and the change in slope on the DCSM vs. SI graphs in the previous section, meaning the above mentioned datasets are not easily separable.

4. Conclusions

We have compared three class separability measures used in machine learning; class scatter matrices (CSM), direct class separability (DCSM) and Thornton's separability measure (SI).

We have shown that the CSM measure does not have a clear functional relationship with the SI while the direct class separability measure does.

The lack of good correlation between CSM and SI is due to the loss of structural information (due to the averaging of instances and classes) in the evaluation of the class scatter matrices measure. This measure is biased to Gaussian compactly clustered classes. It does not work well with multi-clustered classes.

DCSM on the other hand gives further information on the structure of the classes, i.e. their compactness and whether one class overlaps the other or not, by the inverse or direct relationship with the SI measure.

The more separable the classes are, the more direct (i.e. $S_B > S_w+$ & $S_B > S_w-$) the relationship between DCSM and SI as opposed to the inverse relationship (i.e. $S_B > S_w+$ or S_w-) for a non-easily separable dataset.

Direct class separability (DCSM) as opposed to the class scatter matrices (CSM) is a quick and more informative method of extracting information about the class scatter of a dataset.

References

[1] Pattern Classification and Scene Analysis. Richard .O Duda *John Wiley and Sons* 1973.

[2] Feature Subset Selection using Thornton's separability index and its applicability to a number of sparse proximity-based classifiers. J.Greene 2001 *PRASA 2001*.

[3] University of California, Irvine Machine Learning Database Repository at: www.ics.uci.edu/~mlearn/mlrepository /html

[4] http://www.first.fraunhofer.de/~raetsch/ by G.Rätsch.

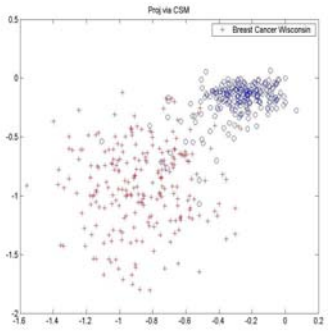


Figure 3.1 Projecting B-Cancer Wisconsin via CSM

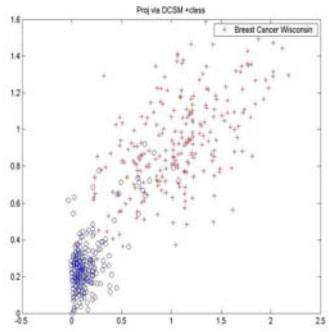


Figure 3.2 Projecting B-Cancer Wisconsin via DCSM

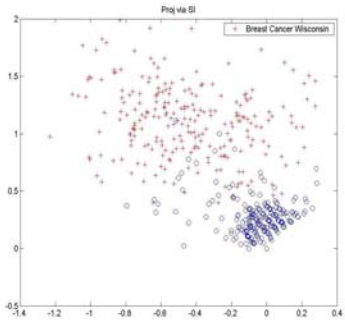


Figure 3.3. Projecting B-Cancer Wisconsin via SI

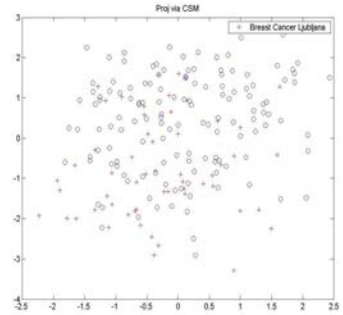


Figure 3.4 Projecting B-Cancer (Ljubljana) via CSM

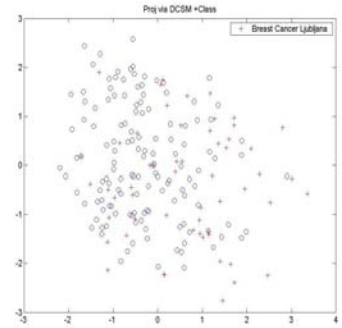


Figure 3.5 Projecting B-Cancer (Ljubljana) via DCSM

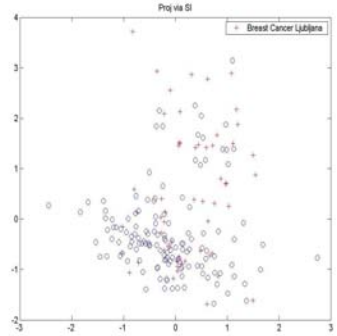


Figure 3.6 Projecting B-Cancer (Ljubljana) via SI

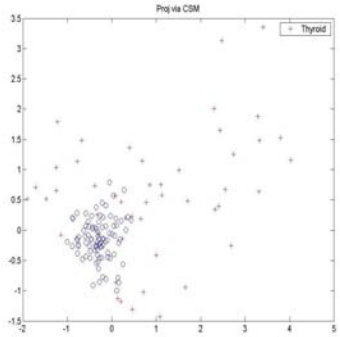


Figure 3.7 Projecting Thyroid via CSM

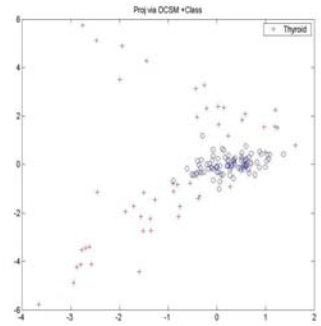


Figure 3.8 Projecting Thyroid via DCSM

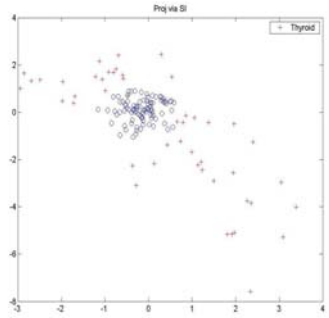


Figure 3.9 Projecting Thyroid via SI

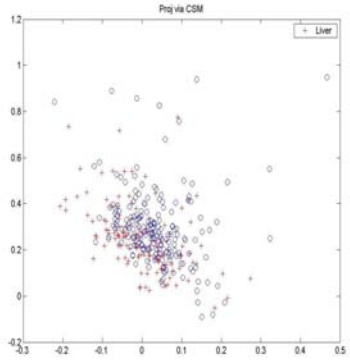


Figure 3.10 Projecting Liver via CSM

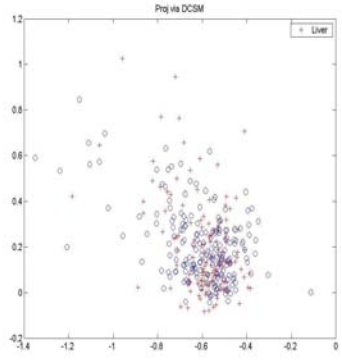


Figure 3.11 Projecting Liver via DCSM

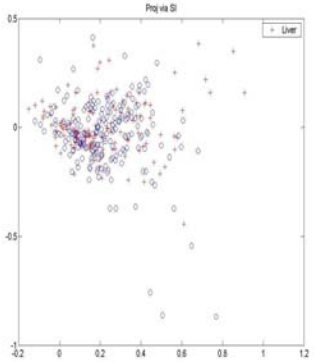


Figure 3.12 Projecting Liver via SI

Using artificial intelligence for data reduction in mechanical engineering

L. Mdlazi[1], C.J. Stander[1], P.S. Heyns [1], T. Marwala[2]

[1] Dynamic Systems Group
Department of Mechanical and Aeronautical Engineering, University of Pretoria
Pretoria, 0002, South Africa
E-mail: Lungile.Mdlazi@eskom.co.za

[2] School of Electrical and Information Engineering,
University of the Witwatersrand, Private Bag 3, Wits, 2050, South Africa
E-mail: t.marwala@ee.wits.ac.za

Abstract

In this paper artificial neural networks and support vector machines are used to reduce the amount of vibration data that is required to estimate the Time Domain Average of a gear vibration signal. Two models for estimating the time domain average of a gear vibration signal are proposed. The models are tested on data from an accelerated gear life test rig. Experimental results indicate that the required data for calculating the Time Domain Average of a gear vibration signal can be reduced by up to 75% when the proposed models are implemented.

1. Introduction

Calculating the Time Domain Average (TDA) of a gear vibration signal by direct averaging using digital computers requires large amounts of data [1-7]. This requirement makes it difficult to develop online gearbox condition monitoring systems that utilize time domain averaging calculated by direct averaging to enhance diagnostic capability. This study presents a novel approach to estimating the TDA of a gear vibration signal, using less data than would be used when calculating the TDA by direct averaging.

Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) are used for estimating the TDA of a gear vibration signal. Two models are presented. The input data comprises rotation-synchronized gear vibration signals and the output is the TDA of the gear vibration signal. When Model 1 is used, the results indicate that the amount of gear vibration data required for calculating the TDA is reduced to 25 percent of the amount of data required when calculating the TDA by direct averaging. When Model 2 is used, the amount of data to be stored in the data acquisition system is reduced to less than 20 percent of the data that would be stored when calculating the TDA by direct averaging. The ANNs that are implemented are Multi-Layer Perceptrons (MLPs) and Radial Basis Functions (RBFs) [8]. Two parameters were selected to verify whether the TDA estimated by the models retains the original diagnostic capability of the TDA. These parameters are the kurtosis for impulses and the peak value for overall vibration. The computational time is also compared to determine the suitability of the proposed models in real-time analysis.

2. ANNs and SVMs

ANNs and SVMs may be viewed as parameterized non-linear mapping of input data to the output data. Learning algorithms are viewed as methods for finding parameter values that look probable in the light of the data. The learning process takes place by training the ANNs or SVMs through supervised learning. Supervised learning is the case where the input data set and the output data set are both known, and ANNs or SVMs are used to approximate the functional mapping between the two data sets.

2.1 Multi-layer Peceptron

A two-layered MLP architecture was used. This selection was made because of the universal approximation theorem, which states that a two-layered architecture is adequate for the MLP. The MLP provides a distributed representation with respect to the input space due to the cross-coupling between hidden units. The output of a two-layer perceptron can be expressed as the following equation:

$$y_k = f_{outer}\left(\sum_{j=1}^{M} w_{kj}^{(2)} f_{inner}\left(\sum w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \quad (1)$$

where f_{outer} and f_{inner} are activation functions, $w_{ji}^{(1)}$ denotes a weight in the first layer, going from input i to hidden unit j, $w_{k0}^{(2)}$ is the bias for the hidden unit k and $w_{kj}^{(2)}$ denotes a weight in the second layer. The parameter f_{inner} was selected as hyperbolic tangent function "tanh" and f_{outer} was selected as a linear function. The hyperbolic tangent function maps the interval [-∞,∞] onto the interval [-1,1] and the linear activation function maps the interval [-∞,∞] onto the interval [-∞,∞]. The maximum-likelihood approach was used for training the MLP network. The sum-of-squares-of-error and the weight decay regularization was used as cost functions [8, 9]. The weight decay penalizes large weights and ensures that the mapping function is smooth, avoiding an over-fitted mapping between the input data and the output data [9]. In this study, a regularization coefficient of 1.5 was found most suitable. The weights w_i and biases in the hidden layers were varied using Scaled Conjugate Gradient (SCG) optimization until the cost function was minimized [10]. It was determined empirically that a two-layer MLP network with five hidden units was best suited to this application.

2.2 Radial basis functions

The RBF network approximates functions by a combination of radial basis functions and a linear output layer. The RBF neural networks provide a smooth interpolating function for which the number of basis functions is determined by the complexity of the mapping to be represented, rather than by the data set. The RBF neural network mapping is given by

$$y_k(\mathbf{x}) = \sum_{j=1}^{M} \varpi_{kj}\phi_j(\mathbf{x}) + \varpi_{k0} \quad (2)$$

where ω_{k0} are the biases, ω_{kj} are the output layer weights, $\mathbf{x}$ is the d-dimensional input vector and $\phi_j(\cdot)$ is the j[th] basis function. The thin plate-spline basis function was used in this study [8]. The radial basis function is trained in two stages. In the first stage the input data set $\mathbf{x}^n$ alone is used to determine the basis function parameters. After the first training stage, the basis functions are kept fixed and the second layer of weights is determined in the second training phase. Since the basis functions are considered fixed, the network is equivalent to a single-layer network that can be optimized by minimizing a suitable error function. The sum-of-square error function is also used to train RBFs. The error function is a quadratic function of the weights and its minimum can therefore be found in terms of the solution of a set of linear equations. For regression the basis function parameters are found by treating the basis function centers and widths, along with the second-layer weights, as adaptive parameters to be determined by minimizing the error function. In this study it was determined empirically that a RBF network with five basis functions was most suitable.

2.3 Support vector machines

SVMs were developed by Vapnik [9] and have gained much popularity in recent years. The SVM formulation embodies the Structural Risk Minimization (SRM) principle, which has been shown to be superior to the traditional Empirical Risk Minimization (ERM) principle employed by conventional neural networks [9, 11, 12]. SRM minimizes an upper limit on the expected risk, as opposed to the ERM that minimizes the error on the training data. It is this difference that gives SVMs a greater ability to generalize. When SVMs are applied to regression problems, loss functions that include a distance measure are used. The ε-insensitive loss function [9] was selected for this study. The ε-insensitive loss function is defined by:

$$L_e(y) = \begin{cases} 0 & \text{for } |f(\mathrm{x}) - y| \\ |f(\mathrm{x}) - y| - \varepsilon & \text{Otherwise} \end{cases}. \quad (3)$$

In non-linear regression, non-linear mapping is used to map the data to a higher dimensional feature space where linear regression is performed. The kernel approach is employed to address the curse of dimensionality. The non-linear support vector regression solution, using the ε-insensitive loss function, is given by:

$$\max_{\alpha,\alpha^*}(\alpha,\alpha^*) = \max_{\alpha\alpha^*} \sum_{i=1}^{l}\sum_{j=1}^{j} \alpha^*(y_i-\epsilon) - \alpha_i(y_i-\epsilon) \quad (4)$$

$$-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j)$$

with constraints,

$$0 \le \alpha_i, \alpha_i^* \le C, \qquad i = 1, K, l \quad (5)$$

$$\sum_{il=1}^{l}(\alpha_i - \alpha_i^*) = 0$$

Solving Equation (3) with the constraints in Equation (5) determines the Lagrange multipliers, α and α^* and the regression function is given by

$$f(\mathrm{x}) = \sum_{i=1}^{i}(\overline{\alpha}_i - \overline{\alpha}_i^*)K(\mathrm{x}_i,\mathrm{x}) + \overline{b} \quad (6)$$

where

$$\langle \overline{w}, x\rangle = \sum_{SVs}(\alpha_i - \alpha_i^*)K(x_i,x_j) \quad (7)$$

$$\overline{b} = -\frac{1}{2}\sum_{i=1}(\alpha_i-\alpha_i^*)(K(x_i,x_r)+K(x_i,x_s)).$$

Different kernels were investigated for mapping the data to a higher dimensional feature space where linear regression was performed. The exponential radial basis function kernel [11] with an order of 10 was found most suitable for this application.

3. Proposed models

Two different models are proposed. Model 1 maps the input space to the target using simple feed-forward network configuration. The size of the input space is systematically reduced to find the optimal number of input vectors that can be used to estimate the target vector correctly. When the ANNs and SVMs are properly trained, Model 1 is capable of mapping the input space to the target, using less data than would otherwise be used when calculating the TDA by direct averaging. It was determined empirically that 40 rotation-synchronized gear vibration signals were suitable for predicting the TDA with Model 1. Consequently, the amount of data required for calculating the TDA is reduced to 25 percent when using 40 rotation-synchronized gear vibration signals, since 160 rotation-synchronized gear vibration signals were used in calculating the TDA by direct averaging. Figure 2 shows a schematic diagram of the proposed methodology.

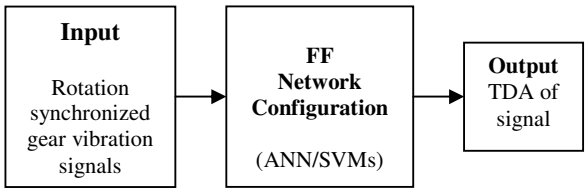


Figure 1. Schematic of proposed methodology

Model 2 estimates the TDA of the input space in small sequential steps, analogous to taking a running average of the input space. This model consists of a number of feed-forward networks. Instead of using the network to estimate the TDA of the entire input data, Model 2 first sequentially estimates the average of subsections of the input data. The output of the first stage is used as input into the second network that estimates the TDA of the entire input data. The feed-forward networks are trained off-line to reduce computation time. In Model 2, data can be discarded immediately after use. This means that the model does not require large amounts of data to be stored in the data logger, even though all the data should be collected. In this study, 10 rotation-synchronized gear vibration signals were found most suitable for estimating the instantaneous average in the first stage of estimation. As a result, the amount of vibration data that is stored in the data logger was reduced to less than 20 percent of the amount of data that is stored in the data logger when calculating the TDA by direct averaging.

4. Estimation results

Model 1 and Model 2 were used for estimating the TDA of the gear vibration signal so that the TDA estimated by the proposed models could be compared with the TDA calculated by direct averaging. The data that was used was from the accelerated gear life test rig [13, 14]. Comparisons were made in the time and the frequency domains. To quantify the accuracy of the TDA estimated by the models, the 'fit' parameter η_{sim} [15] defined by

$$\eta_{sim} = 100\frac{\sum_{k=1}^{N}|e(k)|}{\sum_{k=1}^{N}|y_{desired}(k)|} \quad [\%], \quad (8)$$

was used. In Equation 8 *e(k)* is the simulation

accuracy defined by Equation (9) and N is the number of data points used.

$$e(k) = y_{desired}(k) - y_{achieved}(k) \quad (9)$$

$y_{desired}$ is the TDA signal calculated by direct averaging and $y_{achieved}$ is the TDA signal estimated by the models. The 'fit' parameter η_{sim} gives a single value for each simulation, therefore can be used to compare the performance of the different formulations over the entire gear life. A high value for the 'fit' parameter η_{sim} implies a bad fit whereas a low value implies a good fit. Through trial and error it was established that $\eta_{sim} = 40\%$ is a suitable upper cut-off point for simulation accuracy.

For Model 1, 40 rotation-synchronized gear vibration signals from the first test with 1024 points per revolution were used for training the ANNs and 256 points per revolution were used to train SVMs. This resulted in training sets of dimensions 1×40×1024 and 1×40×256 respectively. Fifteen test data sets were measured through the life of the gear and used as validation sets. This resulted in validation data sets of dimensions 15×160×1024 15×160×256 for ANNs and SVMs respectively. For Model 2, the whole data set of 160 rotation-synchronized gear vibration signals from the first test was used for training the ANNs and SVMs. This resulted in a training set of dimensions 1×160×1024 for ANNs and 1×160×256 for SVMs. The rest of the data of dimensions 15×160×1024 and 15×160×256 were used as validation data for ANNs and SVMs respectively.

Figure 2 shows the estimation results obtained when Model 1 with an MLP network was simulated using unseen validation data sets of 40 input signals during the running-in stages of gear life. The dotted line is the TDA estimated by Model 1, and the solid line is the TDA calculated by direct averaging. The first plot in Figure 2 is the time domain representation of the results; the second plot is the frequency domain representation. It is observed that both the time and frequency domain representations are almost exact fits. This shows that Model 1 with MLP networks retains the time and frequency domain properties of the original time domain averaging process when using gear vibration data from the accelerated gear life test rig.

Similar performances were obtained throughout the life of the gear using both models with RBFs and SVMs, even though there were significant changes in the vibration signatures as the condition of the monitored gear deteriorated. The changes in the vibration signatures were due to changes in the meshing stiffness caused by cracks in the gear teeth. The good performance is due to the mapping and generalization capabilities of ANNs and SVMs.

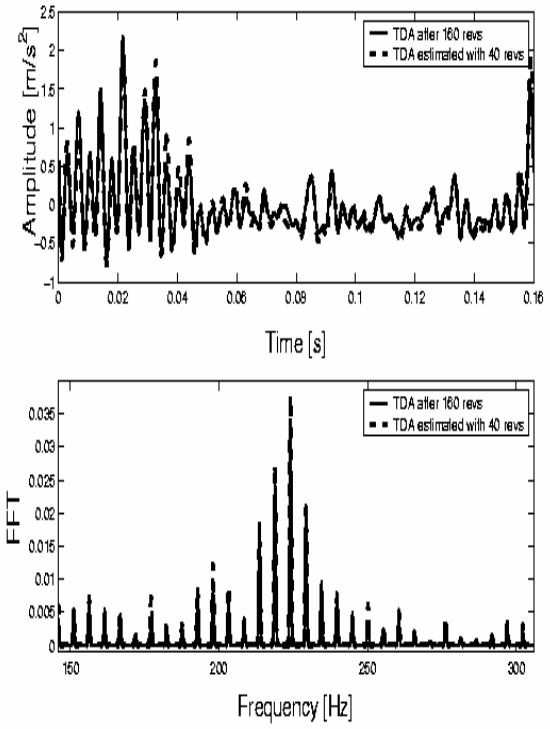


Figure 2. Model 1 with MLP estimation using data from a test conducted under constant load conditions

Figure 3 shows the simulation accuracy η_{sim} plotted against the gear life for Model 1. It is observed that Model 1 with RBF network and Model 1 with SVMs give similarly performance. Their performance was slightly better than that of Model 1 with MLP networks. The performances of all three formulations are acceptable because η_{sim} is less than the cut-off value for all the formulations.

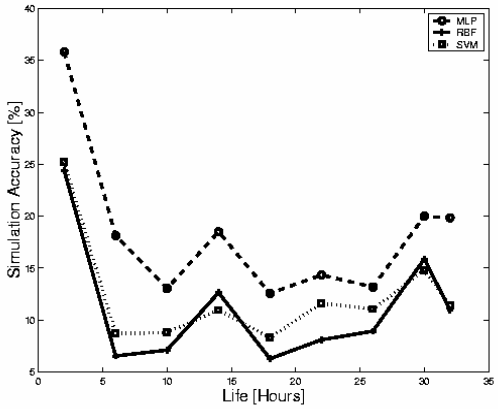


Figure 3. Model 1 Simulation accuracy η_{sim} vs. gear life under constant load conditions

In addition to simply considering the goodness of fit, the diagnostic capabilities of the TDA estimated by the models was assessed using the peak value of the vibration X_{max} during a given interval T and the Kurtosis. The peak value is used to monitor the overall magnitude of the vibration to distinguish between acceptable and unacceptable vibration levels and the kurtosis is useful for detecting the presence of an impulse within the vibration signal [16]. The peak value of the vibration X_{max} and the kurtosis of the TDA calculated using direct averaging, were compared to the TDA estimate from the proposed models.

Figure 4 is a plot of X_{max} and kurtosis calculated from the TDA estimated by Model 1, superimposed on the X_{max} and kurtosis calculated from the TDA obtained by using direct averaging. Figure 4 indicates that the kurtosis is an exact fit for all three formulations. This implies that the TDA predicted by Model 1 can be used to monitor the presence of impulses in the measured gear vibration signal. It is also observed that the kurtosis is very high during the early stages of gear life. This is characteristic of the running-in stages of the gear life, during which the vibration signature tends to be random. A similar trend is observed during the wear-out stage in which strong impulses are caused by the reduction in stiffness in cracked or broken gear teeth. Only the peak values obtained from Model 1 with MLP and SVM are close fits and can be used to monitor the amplitude of the overall vibration. Model 1 with RBF achieved an unsatisfactory performance because the RBF network selected in this simulation was not optimal; consequently it did not generalize well to changes in the measured vibration as gear failure progressed. Similar results were obtained using Model 2.

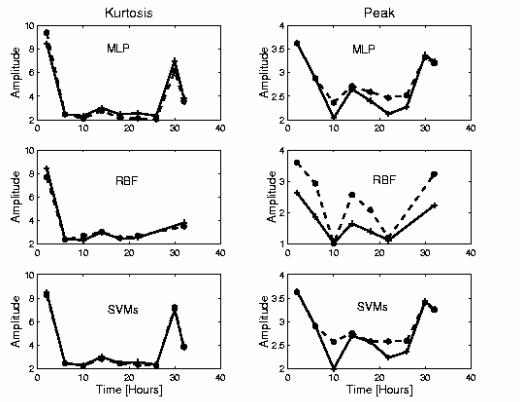


Figure 4. Comparison of kurtosis and peak values for the TDA calculated by direct averaging (+) and the TDA predicted by Model 1 (*) with MLP, RBF and SVMs

To put the proposed models into perspective, their computation time was compared to that of the existing time domain averaging method using a Pentium 4 computer with a 1.60 GHz processor. The computation times are listed in Table 1.

Table 1. Computation time in seconds

Time parameter [s]	**TDA**	**MLP**	**RBF**	**SVM**
Pre-processing Model 1	1.011	0.703	0.703	0.703
Training Model 1	-	22.24	2.219	497.0
Simulating Model 1	0.75	0.016	0.047	5.500
Pre-processing Model 2	1.011	1.011	1.011	1.011
Training Model 2	-	1.14	1.015	963.8
Simulating Model 2	-	0.08	0.078	83.76

It is clear that Model 1 requires less pre-processing time than the TDA calculated by direct averaging. This is because Model 1 uses 25 percent of the vibration data while the original TDA process uses all of the vibration data. The required pre-processing time for Model 2 is equal to that of calculating the TDA by direct averaging because both models use the same amount of vibration data. When Model 1 and Model 2 are used, RBF and MLP give the best performance in terms of simulating time and SVMs give the poorest performance. The models are trained off-line therefore the training time would not influence the performance in real-time applications. When the models are used with SVMs their performance is

poor. The poor performance with SVMs is due to the optimization problem. In SVMs optimization is a quadratic problem with 2N variables, where N is the number of data training points. Longer training times are needed because more operations are required in the process. This is much slower than the MLP and RBF neural networks in which only the weights and biases or the basis centers are obtained by minimizing the error functions.

5. Conclusion

In this paper a novel approach that uses artificial neural networks and support vector machines to reduce the amount of data that is required to calculate the time domain average of a gear vibration signal is presented. Two models are proposed. Using Model 1 the data for calculating the Time Domain Average was reduced to 25 percent of the data required to calculate the Time Domain Average by direct averaging. Model 2 was found to be excellent at estimating the Time Domain Average but had the disadvantage of requiring more operations to execute. When using Model 2, less than 20 percent of the data are needed to be stored in the data logger at any given time. Furthermore, the suitability of the developed models for diagnostic purposes was assessed. It was observed that the performances of the Model 1 and Model 2 were similar over the entire life of the gear. The good performance of Model 2 can be attributed to the fact that Model 2 uses the whole data set for training and simulation, whereas Model 1 uses only a section of the data set. Using the whole data set during training and simulation exposes the formulations in the model to all the transient effects in the data, resulting in a more accurate estimate of the Time Domain Average. The performance of Model 1 relies on the generalization capabilities of the formulation used.

6. References

[1] L. Hongxing, Z. Hongfu, J. Chengyu and Q. Liangheng, 2000 *Mechanical Systems and Signal Processing* 14(2), pp. 279-285, An improved algorithm for direct time domain averaging.

[2] C.R. Trimble, 1968 *Hewlett-Packard Journal* 19(8), pp. 2-7. What is signal averaging?

[3] S. Braun, 1975 *Acustica* (32), pp. 69-77, The extraction of periodic waveforms by time domain averaging.

[4] S. Braun and B. Seth, 1980 *Journal of Sound and Vibration* 70(4), pp. 513-526 Analysis of repetitive mechanism signatures.

[5] P. D. McFadden, 1987 *Mechanical Systems and Signal Processing* (1), pp. 83-95, A revised model for the extraction of periodic waveforms by time domain averaging.

[6] P. D. McFadden, 1989 *Mechanical Systems and Signal Processing* (3), pp. 87-97, Interpolation techniques for time domain averaging of gear vibration.

[7] B. Samanta, 2004 *Mechanical Systems and Signal Processing* (18), pp. 625-644, Gear fault detection using artificial neural networks and support vector machines.

[8] C.M. Bishop, 1995 *Neural networks for pattern recognition*, Oxford: Clarendon Press.

[9] S.R. Gunn, 1998 *Technical report*, University of Southampton, Department of Electrical and Computer Science, UK, Support vector machines for classification and regression.

[10] S. Haykin, 1999 *Neural networks*, 2nd edition, New Jersey, USA: Prentice-Hall Inc.

[11] V.N. Vapnik, 1999 *IEEE Transactions on Neural Networks* 10, pp. 988-1000, An overview of learning theory.

[12] V.N. Vapnik, 1995 *The nature of statistical learning theory*, New York, USA: Springer-Verlag.

[13] C.J. Stander and P.S Heyns, 2002 Proceedings of the 15th International Congress on Condition Monitoring and Diagnostic Engineering Management, Birmingham UK, 2-4 September 2002, pp. 220-230. Instantaneous shaft speed monitoring of gearboxes under fluctuating load conditions.

[14] C.J. Stander and P.S Heyns, 2002 *Mechanical Systems and Signal Processing* 16(6) pp. 1005-1024. Using vibration monitoring for local fault detection on gears operating under fluctuating load conditions.

[15] A.D. Raath, 1992 *Structural dynamic response reconstruction in the time domain*, Ph.D. thesis, Department of Mechanical and Aeronautical Engineering, University of Pretoria.

[16] M. P. Norton, 1989 *Fundamentals of noise and vibration analysis for engineers*, New York: Cambridge University Press.

Pattern Recognition in Service of People with Disabilities

L. Coetzee and E. Barnard

Information Society Technologies Centre (ISTC),
CSIR, P.O. Box 395, Pretoria, 0001
{louis.coetzee}/{etienne.barnard}@csir.co.za

Abstract

South Africa has a large community of people living with a variety of disabilities. Very often they can be productive members of society, but are excluded due to the high cost of assistive technologies. These assistive technologies are based on a variety of pattern-recognition techniques and algorithms. This paper analyses a number of disabilities, their assistive devices and the associated technologies in order to highlight the role pattern recognition plays in enabling accessibility and improving human computer interaction for people living with disabilities. In addition we point out some areas where pattern-recognition research can beneficially be adapted to address the needs of people with disabilities, and argue for the use of open-source technologies to improve accessibility for a larger part of the disabled community in South Africa.

Keywords: pattern recognition; disability; accessibility; information; human computer interface; open source

1. Introduction

According to Statistics South Africa, at least 5.9% of the people living in South Africa live with one or more disabilities[1]. People with disabilities often cannot contribute to the economy, even though many have the desire and ability to; similarly, they are often prevented from participation in other spheres of society to the extent of their abilities and desire[2]. The South African government has recognised this fact and has introduced legislation in an attempt to increase the fraction of people living with disabilities that are economically active, and to ensure that increased support is made available to improve the quality of life experienced by people with disabilities[3].

Depending on the specific disability, various assistive devices and technologies are required to empower a person living with a disability to become a productive member of society. These devices vary tremendously with respect to factors such as technological sophistication and user friendliness. However, a few common trends characterise the majority of such devices:

- They tend to be imported and thus expensive, placing them out of reach of most disabled people.
- These devices and applications are proprietary and closed.
- These devices and applications are not localised to allow for the cultural and linguistic variety that characterises South African society.
- Finally, almost all of these devices utilise a variety of pattern-recognition techniques and algorithms.

The current paper is an overview of the National Accessibility Portal (NAP), an initiative between the CSIR, the Office of the Status of Disabled Persons (OSDP), the Independent Living Centre (ILC), the SA National Council for the Blind (SANCB), the Deaf Federation of South Africa (DEAFSA) and the National Council for Persons with Physical Disabilities in SA (NCPPDSA). NAP's primary aim is to provide a national networking and communication system based on Internet technologies for people with disabilities; and to improve accessibility to information about or for people with disabilities in a cost effective way.

We analyse a number of disabilities and their associated assistive technologies (with the focus on technologies enhancing accessibility to information

using personal computers) in order to identify the utilised pattern-recognition algorithms. It is envisioned that this understanding will ultimately direct us to develop localised South African alternatives and allow us to contribute to the few open-source alternatives.

In the next section, we present technologies associated with low vision and blindness. In Section 3 we analyse technologies associated with deafness. Section 4 contains a discussion regarding physical disabilities, and is followed by a discussion of the required and actual characteristics of pattern-recognition applications aimed at people with disabilities, and an open-source perspective on technical applications (Section 5). A conclusion is presented in Section 6.

2. Low Vision and Blindness

A number of different categories classifying visual impairments exists. They include:

- *Low Vision, severe* – where visual tasks are performed at a reduced level.
- *Low Vision, profound* – gross visual tasks are performed with difficulty.
- *Near Blind – vision* is classified as being unreliable, and
- *Blind* – the person is totally without sight.

A Web browser installed on a standard personal computer has become one of the best tools to access information via the Internet for people who are not visually impaired. For a person suffering from low vision a screen magnifier is an important tool used to magnify regions of the user's desktop. Using a screen magnifier the user has access to the standard applications such as web browsers and email clients, thus providing the person with unlimited access to information. People suffering from colour blindness can benefit from using high contrast themes.

For a person with severe visual limitations (e.g. blindness) the situation is more complex. To enable access to the personal computer, specialised hardware (such as braille keyboards and displays) or software (such as screen readers) is required. Screen readers normally plug into the operating system and receive events and other information from the desktop. The screen reader interprets the received information and provides audible prompts of what is happening on the desktop. The audible prompts are generated using text-to-speech engines.

Commercial screen readers for Windows-based machines are quite common. They interface with Microsoft's SAPI and provide good text-to-speech prompts for languages such as English . These readers interact well with applications, allowing for a manageable desktop. Unfortunately, they are expensive (very often more than R10 000 for a single user license), and do not allow for the use of indigenous languages.

3. Deafness

The most common forms of deafness can be categorised into 3 broad categories:

- *Conductive hearing loss* – caused by damage to the outer or middle ear. Sufferers may benefit from the use of hearing aids.
- *Sensorineural hearing loss* – caused by damage in the hair cells of the inner ear or nerves. Sufferers do not benefit from the use of hearing aids.
- *Combination* of conductive and sensorineural hearing loss.

Sign language is commonly used as communication method by deaf people. Due to the nature of the disability deaf people have extreme difficulty communicating with other people, or absorbing information contained in multimedia (e.g. TV). Literate deaf people have few problems obtaining information or interacting with computers. However, a large percentage of deaf people are also illiterate, thus severely limiting their ability to interact and share information.

An exciting prospect is the application of virtual reality avatars, used in conjunction with human language technologies, to create sign language from interpreted text. Another exciting prospect is the use of speech recognition to automatically generate subscripts on multimedia (video) footage.

4. Physical disabilities

A vast number of physical disabilities exists. They include:

- *Paraplegia* – complete paralysis of the lower half of the body including both legs, usually caused by damage to the spinal cord.
- *Quadriplegia* – complete paralysis of the body from the neck down.
- *Cerebral palsy* – a disorder usually caused by brain damage occurring at or before birth and marked by muscular impairment. Often accompanied by poor coordination, it sometimes involves speech and learning difficulties.
- *Muscular dystrophy* – a group of progressive muscle disorders caused by a defect in one or more genes that control muscle function and characterised by gradual irreversible wasting of skeletal muscle.

Access to computers using the standard input devices requires a certain amount of mobility, the very element a physical disability impacts the most. Paraplegics typically have full mobility in the upper halves of their bodies, thus empowering them to use standard pointing and input devices, provided that the environment is ergonomically appropriate.

Quadriplegia has a far greater impact on mobility, preventing usage of normal character and pointing devices. In this domain, the use of head-mounted pointing devices (requiring sophisticated target tracking algorithms and image processing capabilities), in conjunction with on-screen keyboards (in combination with predictive text) and a variety of switches (mouse-click simulation devices), improves accessibility significantly. If the user also suffers from a speech disability, the above scenario is extended with the addition of a text-to-speech output device – thus allowing the person to actively communicate. Alternate *command-and-control* mechanisms use speech recognition as input device, allowing interaction with applications installed on the computer.

Sufferers of cerebral palsy and other disorders have extreme mobility limitations. Standard input devices are impractical. The use of large keys on keyboards in conjunction with applications providing synthesised voice output has proven to be beneficial as learning aids, and for information access.

5. Technological tools: characteristics, challenges and the open-source approach

The tools described in the previous section along with similar tools aimed at assisting people with disabilities, make extensive use of pattern-recognition algorithms to compensate for physical or sensory disabilities. Examples of such algorithms include:

- *Visual recognition and tracking* algorithms for gaze-tracking systems, and for use in gesture recognition[4, 5].
- *Speech recognition* to provide deaf users with transcriptions of spoken material, and to respond to commands of people with disabilities[6].
- *Language-processing and speech-synthesis* algorithms to generate spoken output for blind people – for example, in screen readers[7].
- Algorithms for *context interpretation*, for use in virtual-reality avatars and in command-and-control interfaces for various applications.

Together these are exciting and challenging applications of pattern recognition; we now discuss some of the characteristics of this domain from a pattern-recognition perspective and highlight some of the limitations of current approaches.

Some of the most salient aspects of these applications of pattern recognition are:

- Users of such systems are highly cooperative repeat users – thus, the pattern-recognition algorithm is not required to perform with perfect accuracy, but it does need to be highly robust and predictable. Users of an eye-tracking system, for example, will be tolerant of failures in tracking performance, as long as the failures do not generate false key presses.

  This also implies that user-adaptive systems are of great importance in this domain. Since a user of known identity will repeatedly interact with a given system, it is possible to refine the performance of the system for that particular user,

both during an initial training phase and during on-going usage; this should produce significantly more accurate behaviour.

- Since these systems invariably function with a human in the loop, it is possible to combine human and machine intelligence in ways that leverage the strengths of both. Specifically, humans are very good at extracting and processing semantically relevant information, whereas pattern recognition performs well at the syntactic (structural) and lower levels. This implies, for example, that an optimal speech recogniser for command-and-control applications can operate with a relatively small number of keywords, and rely on the user of the system to combine those keywords to operate the system in an acceptable fashion. (Therefore, sophisticated attempts at natural-language processing are not likely to be of great value in such applications.)

- People from all language groups suffer from disabilities, and to assist them in their own languages is of crucial importance in many applications (e.g. for monolingual users, or when the cognitive load of operating in something other than the user's first language is not tolerable). Consider, for example, a text-to-speech device that is used to assist a speech-impaired person in communicating with her family: even if family members are conversant with several languages, support for their home language is required for acceptance of the device.

  It is therefore imperative that all these systems be designed within a multi-lingual framework from the outset, and that this framework be populated with as many languages as possible.

Off-the-shelf pattern-recognition algorithms are generally not optimal for these conditions, which implies that there is much scope for algorithm development or refinement[8, 9]. Current speech-recognition algorithms, for example, are strongly biased towards "normative" or "standard" speech. Highly adaptive algorithms, which are able to handle idiosyncratic pronunciations produced by users with various speech defects, would be of great practical value. Similarly, sophisticated eye-tracking algorithms should adapt themselves to user characteristics and behaviour in a transparent fashion.

In addition, general design paradigms are required in order to assist developers in creating systems with an appropriate assignment of responsibilities to the user and system, respectively. To this end, abstract models of task domains, user characteristics and system characteristics must be developed.

Although much research and development in this area remain to be done, a substantial range of assistive devices have already been developed. Current commercial products in this domain are often of great use, but are usually prohibitively expensive for the majority of potential users in a developing country. Fortunately, a number of open-source alternatives exist. At the forefront of current development is the *Gnopernicus* suite of applications[10]. It consists of a screen reader, a screen magnifier and braille input and output interfaces. Gnopernicus interacts with the Gnome desktop through the AT-SPI (assistive technology service provider interface)[11]. The screen reader utilises the gnome-speech API to synthesise voice. Gnome-speech abstracts a number of different TTS engines, which includes Java FreeTTS[12] and Festival[13].

GOK (the gnome on-screen keyboard) is used as an alternative input mechanism[14]. GOK provides complete control over the gnome desktop, again by interacting with the AT-SPI. Using only a pointing device, the end user is thus able to generate general-purpose keyboard input. GOK utilises predictive text for faster text generation. Figure 1 depicts a screen grab of GOK with predictive text providing choices for *pattern* or *patterns* based on the input *patt*.

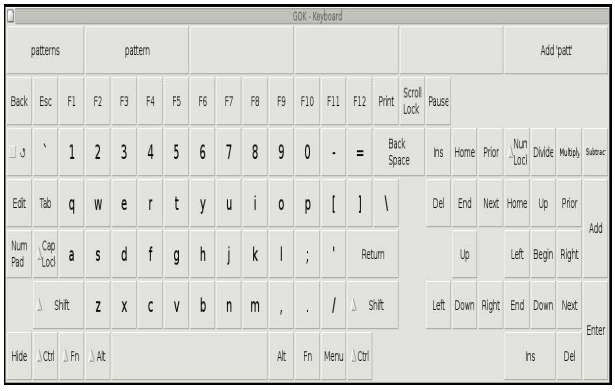


Figure 1: GOK – Gnome on-screen keyboard

For people with physical disabilities usage of normal pointing devices is problematic. Utilities to control the NaturalPoint trackIR and SmartNAV (as presented in Figure 2) are currently under development[15]. These utilities use advanced target tracking algorithms to track a specific reflective element in continuous frames. The tracked object is then translated into on-screen mouse movement.

Figure 2: TrackIR

Perlbox[16] uses a combination of Festival[13] and Sphinx[17] to provide a front end providing command-and-control capability to the Linux desktop.

The above mentioned technologies provide open API's allowing developers to customise and localise based on the needs of the disabled person. In addition up-front costs are mostly limited to hardware (computer and other devices) with software and associated device drivers based on open source applications.

6. Conclusion

In this paper, we presented a view into the world of disabilities, the assistive devices and technologies normally associated with improving accessibility, with the aim of highlighting the underlying pattern-recognition algorithms. A vast number of disabilities exist, each with their own specialised needs. We focused on three disabilities: visual, hearing and physical. It is clear that human language technologies (more specifically speech recognition and speech synthesis) as well as a variety of image processing algorithms play an important part; however, the specific characteristics of users with disabilities imply that there is much scope to improve the operation of these algorithms by careful consideration of the strengths and limitations of the target users.

Acknowledgements

We would like to thank Marelie Davel as well as the NAP team for their research, inputs and suggestions.

7. References

[1] Statistics South Africa, Census 2001. Primary tables South Africa. Census '96 and 2001 compared, 2004. Report No. 03/02/04 (2001). `http://www.statssa.gov.za/census01/html/c2001primtables.asp`.

[2] Disabled People South Africa – Pocket Guide On Disability Equity. An Empowerment Tool. Published by the DPSA Parliamentary Office on behalf of DPSA. P. O. Box 15, Cape Town 8000.

[3] Integrated National Disability Strategy White Paper. Office of the Deputy President–South Africa, November 1997. `http://www.polity.org.za/html/govdocs/white_papers/disability1.html`.

[4] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Academic Press, San Diego, California, USA, 2nd edition, 1997.

[5] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[6] S.Young. Large Vocabulary Continuous Speech Recognition. *IEEE Signal Processing Magazine*, 13(5):45–57, 1996.

[7] T.Dutoit. *An Introduction to Text-to-Speech Synthesis*. Kluwer Academic Publishers, Dordrecht, 1997.

[8] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. Wiley, New York, 1973.

[9] C.M.Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, 1995.

[10] Gnopernicus Assistive Technology. http://www.baum.ro/gnopernicus.html.

[11] The GNOME Accessibility Project. http://developer.gnome.org/projects/gap/.

[12] FreeTTS. http://freetts.sourceforge.net/.

[13] The Festival Speech Synthesis System. http://www.cstr.ed.ac.uk/projects/festival/.

[14] GOK – Gnome On Screen Keyboard. http://www.gok.ca/.

[15] trackIR for Linux. http://trackir.superlucidity.net/.

[16] Perlbox.org – Linux Speech Control and Voice Recognition. http://perlbox.sourceforge.net/.

[17] The CMU Sphinx Group Open Source Speech Recognition Engine. http://cmusphinx.sourceforge.net/html/cmusphinx.php.

Predicting Global Internet Instability Caused by Worms using Neural Networks

Elbert Marais, Tshilidzi Marwala

School of Electrical and Information Engineering
University of the Witwatersrand
Private Bag 3, WITS, 2050, South Africa

e.marais@ee.wits.ac.za
t.marwala@ee.wits.ac.za

Abstract

Internet *worms* are capable of quickly propagating themselves by exploiting vulnerabilities of software running on hosts that have access to the Internet. Once these worms have infected a computer, they have access to sensitive information on the computer, and are able to corrupt or retransmit this information. This paper describes a method of predicting Internet instability due to the presence of a worm on the Internet, based on information currently available from global Internet routers. A neural network is trained to predict anomalies in the number of router messages received by a router. The output from the network can help to provide an early warning for the presence of a worm.

1. Introduction

The rapid spread of worms leads to a sudden increase in Internet network traffic, which can have a direct impact on the stability of the Internet as a whole. Due to the widespread use and reliance by many businesses on the Internet for generating revenue, a worm is able to cause extensive downtime of infected hosts, as well as making the Internet inaccessible even for those hosts that are not infected.

Many tools are available to prevent the spread of viruses and worms. This project assumes that worms will continue to be written and effective regardless of these tools, and deals with the early detection of a worm that is already causing Internet instability.

The Border Gateway Protocol (BGP) is the current Internet standard for routers to exchange routing information, which relies on the TCP protocol for network transport of these routing messages. From existing research [1, 2, 3, 4], it has been shown that this routing information can be used to distinguish traffic caused by a worm from that caused by large-scale network outages (such as the World Trade Centre attacks on 11 September 2001).

A neural network has been trained to predict the presence of a worm on the Internet, by observing the behaviour of a global Internet router. By training the network to recognise normal routing behaviour, it is able to distinguish unusual behaviour, specifically that of an active worm. This forewarning could significantly limit the impact of a worm, reducing the impact to productivity caused by Internet instability.

This document provides background on the BGP protocol, worms and novelty detection using neural networks. This is followed by the method taken to train the network, and the results that have been obtained.

2. Worms

A worm is a program that can run independently and can propagate a fully working version of itself to other machines [5 p.3]. The main difference between a virus and a worm is that a virus attaches itself to a file or program, whereas a worm is capable of running independently. A worm spreads complete but possibly mutated versions of itself to other computers. It achieves this by exploiting holes in an operating system [6], or relies on users to infect the computers (such as email worms which rely on users to open attachments).

Even with countermeasures such as anti-virus and intrusion detection software in place, viruses continue to exist. New versions of operating systems and software are continuously being released, each possibly providing new ways for a virus or worm to spread. Additionally, worms and viruses that rely on a user for propagation will have a certain level of success. Even though anti-viruses do control the spread of worms, they are only effective after some spread and possible damage has been done. Not all computers have anti-virus software installed, or have the latest virus patterns. In the case of the *CodeRed II* worm, it took sixteen weeks for most hosts to fix the vulnerability, and thousands of systems were not patched six weeks later [7].

The cost effect of worms and viruses is difficult to estimate, since it is a combination of the time spent analysing and patching the problem, as well as the loss of productivity by end-users who are unable to use their computers or the Internet. The effect of worms in the public and private sector has been estimated to cost millions of dollars. The *CodeRed* worm which spread on 19 July 2001, together with subsequent strains, caused approximately $2.6 billion [8]. The estimated cost of SQL Slammer was estimated at $1 billion [9].

The rate at which worms spread is largely dependant on their means of propagation and the number of possible hosts that are susceptible to the worm. A worm relying on users to open attachments will spread slower than a worm that immediately replicates itself once it is present on a computer. Also, since the worms exploit vulnerabilities in specific software or operating systems, only these systems are susceptible to being infected by the worms. Other systems that can't be infected are still able to transmit the worm, as in the case of a worm that affects only Windows systems using email, but can be relayed by other operating systems which forward the emails. The following is an indication of the speed at which worms are able to spread:

- The *CodeRed II* worm required less than 14 hours to reach its saturation of more than 359,000 infections, with the worms spread peaking at just over 2,000 infections per minute [7].
- The *SQL Slammer* worm targeted a buffer overflow vulnerability in computers running Microsoft's SQL Server. The worm infected at least 75,000 hosts over a period of approximately 10 minutes [10].

3. Routing and BGP

Routing provides a means of transporting data packets from a source to a destination. A router is a physical device with many network connections, used for transferring data packets as well as determining which path to use for sending these packets. Each router has several routing tables, which are used to determine the next device to send packets to, based on each packets final destination. Routing protocols are used for inter-router communication to establish these routing tables. This project is focussed on global Internet routers, which relay traffic for the entire Internet.

Due to the dynamic state of the Internet (routers are continuously added, routers go offline temporarily), the routes between endpoints within the network are continuously changing. These route changes are propagated using control messages sent between the routers, so the more IP addresses available the more the overhead load on the network. For these reasons, the Border Gateway Protocol (BGP) is

used as the standard for routing throughout the Internet [11].

It is infeasible for a router to store the routes between all source and destination and IP addresses for the Internet, due to the number of IP addresses available on the Internet. Instead, BGP stores only the routes between Autonomous Systems.

Autonomous Systems are made up of a network or group of networks that have a common administration and common routing policies [12]. This grouping allows for several IP addresses to be represented by a single *Autonomous System*, which greatly simplifies the number of routing entries that each router must hold, since groups of IP addresses are stored together rather than individual values.

Four types of messages are used by BGP for exchanging information between routers [12]:

- *Open*
- *Update*
- *Notification*
- *Keep-Alive*

These messages are relevant only for routing status and update information, and are independent of the data being sent between the routers. The message that is most relevant for this research is the *Update* message. BGP routers send *Update* messages to exchange network accessibility information, such as when a new router is present that causes the router to change its routing paths. These changes are what are propagated to the surrounding networks. Two specific types of *Update* messages are important for the neural network:

- *Announcement* messages which inform a router that a new route is available.
- *Withdrawal* messages indicate that a previously available route has become unavailable.

Routers use these messages combined with several configuration parameters to determine the routing table entry used between the *Autonomous Systems*.

4. Neural Network for Novelty Detection

Artificial neural networks are expert systems based on modelling of the human brain [13]. These networks consist of an input and output layer which is specific to the system, as well as a hidden layer consisting of several *neurons* that are trained using a set of training data. Once a network has been trained it can be used to generate outputs for new input datasets. All data applied to the input layer of a neural network should be normalized to the range of either [-1:1] or [0:1], to ensure that the network uses data within the same range.

One method of distinguishing normal behaviour from novel behaviour is the use of an *autoencoder* [14]. An *autoencoder* consists of an input and output layer with the same number of inputs and outputs, combined with a narrow hidden layer. The network is trained using only positive examples (i.e. standard behaviour), with the training data using the same inputs and outputs. The hidden layer attempts to reconstruct the inputs to match the outputs, which minimizes the error between the inputs and the outputs when non-novel data is presented to the system. The network uses a narrow hidden layer, which forces the network to reduce any redundancies, but still allows the network to detect non-redundant data. An example autoencoder is shown in Figure 1.

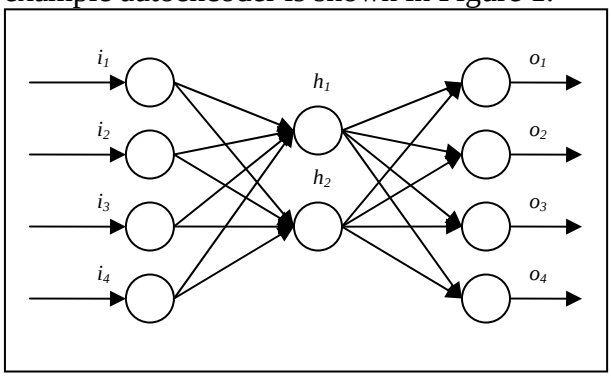


Figure 1. Example of an autoencoder neural network, with four input and output nodes, and two hidden nodes. The inputs and outputs used for training are identical.

The test data that is to be searched for novel behaviour is presented to the trained network.

The measure of novelty for each input set is then given by the mean-square error:

$$e = \frac{1}{n}\sum_{1}^{n}\left((t_n - i_n)^2\right) \quad (1)$$

where n is the number of inputs and outputs, t_n is the n^{th} output and i_n is the n^{th} input.

The network gives a low error for input datasets similar to the training data, but gives a high error for data that is significantly different to the original training data. It is then possible to set a threshold for the maximum error before the data can be classified as novel.

5. Method

Existing research on the effect of worms on global Internet routers has shown that worms cause Internet routing instability [1, 2]. Based on the analysis in [1, 3, 4], it is evident that worms have a distinguishable effect on the number of *Update* messages sent between BGP routers. Figure 2 and Figure 3 shows a marked increase in the number of announcements (BGP Update messages) caused by the *Nimda* virus. It is interesting to note that the Baltimore train wreck which severed several large fibre links on 18 July 2001, and the wide scale internet outages from the World Trade Centre attack on 11 September 2001 don't feature on this graph.

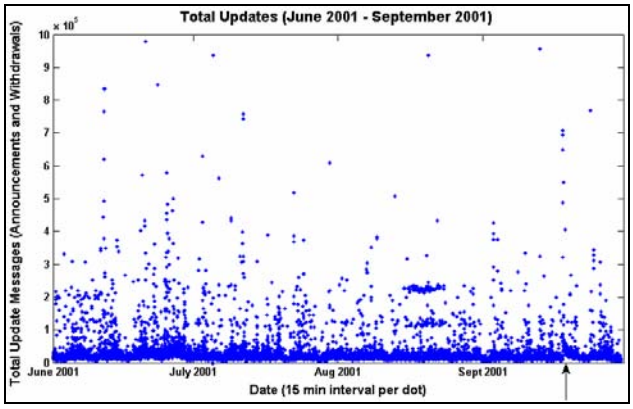


Figure 2. Number of BGP *Update* messages received by a router from June – September 2001. The arrow the increased router messages caused by the *Nimda* worm (September 2001). This data has been extracted from archives of a router based in Amsterdam, and each dot represents the total number of *Update* messages (announcements and withdrawals) in 15 minute time period.

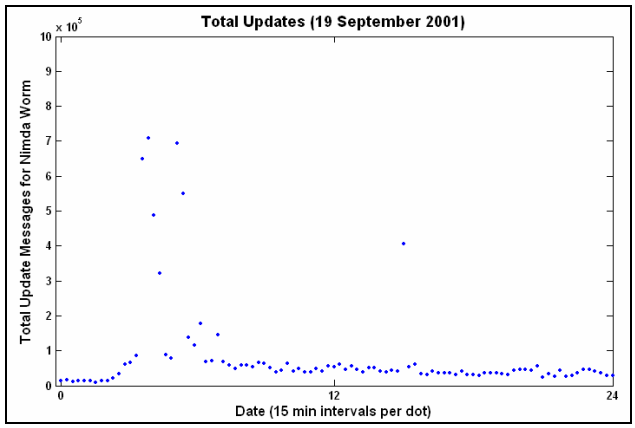


Figure 3. Number of BGP UPDATE messages received on 19 September 2001 (GMT). The sudden increase is due to Internet instability caused by the *Nimda* worm.

The neural network inputs were based on the data shown in Figure 2, but using the total number of *announcement* and *withdrawals* for each fifteen minute interval separately. Fifty inputs were used for the number of *announcement* messages, starting with the number of router announcements for the current time period, and then each subsequent input for the number of announcements for the fifteen minutes prior to the previous time period. This provides the network with the trend of the total a*nnouncement* messages for the current and previous time periods. Another fifty inputs are used for the number of *withdrawals* for each fifteen minute time period, in the same manner as for the *announcement* inputs.

The network was trained using a period which showed no significant worm activity in the dataset shown in Figure 2. The network was configured for the novelty detection method described previously, using the *Scaled Conjugate Gradient* method to optimize the network weights, which was used due to the method's computational efficiency and expediency [15]. As expected for the *autoencoder* network configuration, the mean-square error between the training inputs and the actual network outputs was minimal.

The mean-square error between the network inputs and outputs for September 2001 are shown in Figure 4, for which it is known that the *Nimba* worm significantly impacted the

Internet. The graph clearly indicates when there were increases in network instability, with the largest spike corresponding to the instability caused by *Nimda*.

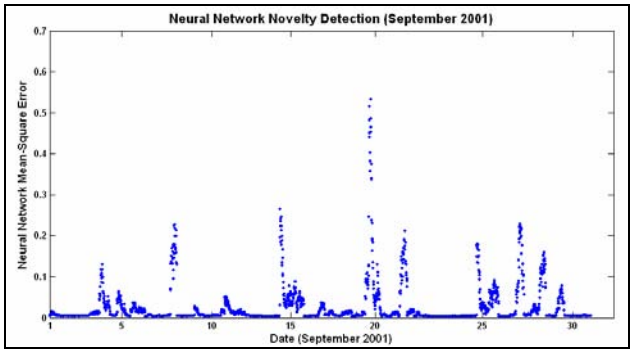


Figure 4. Graph of mean-square error for the trained neural network for September 2001. The largest spike corresponds to the effect of the *Nimda* worm, which appeared on the Internet on 19 September 2001.

6. Conclusion

This project extends from previous analysis which found that the additional load caused by worms corresponds to the number of BGP messages sent between routers. Based on initial findings, it appears that a single router can be used to train a neural network to predict the presence of Internet instability caused by a worm.

From previous incidents of worms, it has been seen that worms are capable of causing a large amount of damage simply through their propagation. Antivirus software and intrusion detection will continue to improve, but are limited in their ability to detect novel worms.

By providing an earlier warning of a potential worm, virus experts will be able to start analysing exploited system vulnerability earlier. This allows for reducing the effects of a worm, by making patches available earlier and alerting uninfected network administrators about the threat.

7. References

[1] I. Dubrawsky, "Effects of worms on Internet routing stability," *Security Focus Article*, June 2003. http://www.securityfocus.net/infocus/1702, Last accessed 30 September 2004.

[2] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S.F. Wu, L. Zhang, "Observation and analysis of BGP behavior under stress," in Proc. *ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[3] J. Cowie, A. Ogielski, B.J. Premore, Y. Yuan, "Global Routing Instabilities during Code Red II and Nimda Worm Propagation," Renesys Corporation, September 2001. http://www.renesys.com/projects/bgp_inst-ability/, Last accessed 30 September 2004.

[4] M. Lad, X. Zhao, B. Zhang, D. Massey and L. Zhang, "An analysis of BGP update burst during slammer attack," in *Proceedings of the 5th International Workshop on Distributed Computing (IWDC)*, December 2003.

[5] E. Spafford, "The Internet worm incident," Tech. Rep. CSD-TR-933, Department of Computer Sciences, Purdue University, September 19, 1991.

[6] S. Staniford, V. Paxson, and N. Weaver. "How to Own the Internet in your spare time," in *USENIX Security Symposium*, August 2002.

[7] D. Moore and C. Shannon, "Code-Red: a case study on the spread and Victims of an Internet worm," in proceedings of the *2002 ACM SICGOMM Internet Measurement Workshop*, Marseille, France, pp. 273–284, November 2002.

[8] D. Moore, C. Shannon, G. Voelker and S. Savage, "Internet quarantine: requirements for containing self-propagating code," in *Proceedings of the 2003 IEEE Infocom Conference*, San Francisco, CA, April 2003.

[9] P. Boutin. "Slammed!," Wired Online Article, Issue 11.07, July 2003,

http://www.wired.com/wired/archive-/11.07/slammer_pr.html, Last accessed 30 September 2004.

[10] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," in *IEEE Security and Privacy*, vol. 1, issue 4, pp. 33-39, Aug 2003.

[11] C. Labovitz, G.R. Malan and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515-528, 1998.

[12] Y. Rekhter, Y. Li. "A border gateway protocol 4 (BGP-4)," Tech. Rep. RFC-1771, March 1995.

[13] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, London, 1998.

[14] N. Japkowicz, C. Myers, M. A. Gluck, "A Novelty Detection Approach to Classification" in IJCAI, pp. 518-52, 1995.

[15] T. T. Jervis and W. J. Fitzgerald. "Optimization Schemes for Neural Networks". Technical Report CUED/F-INFENG/TR 144, Cambridge University Engineering Department, Cambridge, England, 1993.

Eukaryotic RNA Polymerase II Promoter detection by means of an ANN

Gerbert Myburgh, Etienne Barnard

Department of Electrical, Electronic and Computer Engineering
University of Pretoria, 0002, South Africa.

gerbert.myburgh@up.ac.za

Abstract

An automated detection process for Eukaryotic ribonucleic acid (RNA) Polymerase II Promoter is presented. This process employs an artificial neural network (ANN), in conjunction with features that are selected using an information-theoretic approach. An improvement of at least 10% in positive prediction value (PPV) in comparison with current state-of-the-art solutions was obtained.

1. Introduction

The field of pattern recognition is expanding rapidly, with fertile applications appearing at the interfaces with various other disciplines. Bio-informatics is an area where pattern recognition comes into contact with disciplines such as genetics and biochemistry. The resulting study of the principles of biological function and organization is currently one of the most exciting domains of scientific enquiry; although work in this field dates back to before 1860, current knowledge is still very sketchy. Given the vast quantities of data that are being generated, it is crucial that pattern-recognition algorithms be developed to assist in the analysis of genetic data. The specific problem that we study is gene expression [5] – that is, how and why particular genes in a cell are activated.

One of the problems with gene expression that currently receives much attention within the pattern-recognition community is that of promoter and transcription start site (TSS) detection. The process of transcription is described in more detail in section 2 below. Bajic et al. [1] as well as Knudsen [2] have proposed models for promoter detection, but both have limited functionality (which is understandable, given that the transcription process is quite complicated and that satisfactory fundamental models for this process have not been developed). We investigated whether more sophisticated features and an ANN classifier can improve on the models in [1] and [2].

An ANN was therefore developed that can read and process a section of deoxyribonucleic acid (DNA) data and detect whether the section represents a promoter or not. An ANN was chosen because it has the capacity to do classification where complex, non-linear relationships between features and desired outputs exist.

The organization of this paper is as follows: Section 2 describes the DNA transcription process and the complexity of the problem. Our ANN and feature set are described in section 3, while section 4 covers the experiments we have done and the results obtained. Our results are summarised, and a conclusion is drawn in section 5.

2. DNA transcription

Before attempting to explain the work we have done, we provide a brief background on the transcription process, and its role in the functioning of living cells.

2.1. The function of DNA

As explained by Lodish [3] Deoxyribonucleic acid (DNA) is the storehouse required to build a cell or an organism. DNA contains all the information that regulates how a specific cell or organism functions. DNA is formed by sequences of very specific molecules called nucleotides. Every known living organism has a DNA sequence that consists entirely of these nucleotides. There are only four different nucleotide types, or base types, which are conventionally indicated by the letters A, C, G or T. The biochemical properties of these nucleotides are not of interest here – we will simply consider DNA sequences as strings consisting of various combinations of these letters.

A typical DNA sequence will thus be something like
…AACGTAGATTGACACC...

2.2. DNA → RNA → Protein

As mentioned in the previous section DNA stores all the information that regulates the functionality of a cell; the DNA sequence of nucleotides, however, is not directly used without being processed first. The information stored in DNA is first transferred to ribonucleic acid (RNA), which is then in turn transferred to proteins. These

proteins directly control the cell functionality, as well as catalysing the reproduction process for DNA and RNA. This circular process - as shown in Figure 1 - is known as the central dogma of genetics [4].

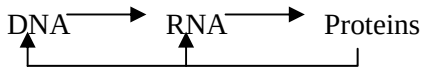


Figure 1: DNA, RNA and Protein relationship.

The process where the genetic information is transferred from DNA to RNA is called transcription.

2.3. Transcription and its complexities

DNA sequences are typically several millions of bases long, of which only certain parts will be transcribed into RNA. The sections that will be transcribed are called exons, while the remaining, un-transcribed sections are called introns. In eukaryotic organisms (organisms with a true nucleus) a single gene can be physically separated in the DNA sequence or, in other words, a single gene can consist of multiple exon sections, separated by introns as shown in Figure 2.

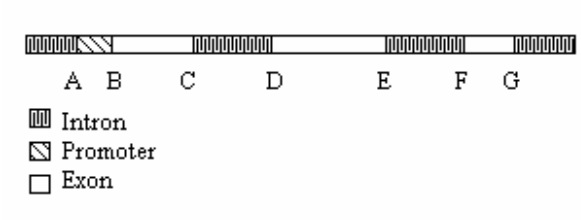


Figure 2: DNA exon and intron sections.

In the figure above the exon areas are everything between points B-C, D-E and F-G, while the introns are everything before A, between C-D, and E-F, as well as everything after G. The area A to B is the promoter region, and the main aim of the work presented in this paper.

The actual biological process of transcription incorporates means for transcribing the correct sections of the DNA, but no known computer or mathematical model exists that describes the process. What is known, however, is that each gene contains at least one promoter somewhere upstream (direction of A) of the actual start site (point B in the figure).

There is a general consensus that a sequence, known as a promoter precedes and marks every new gene or protein section. However, different sources (e.g. [5], [6], [7] and [8]) provide different "consensus" sequences, all of them called the "TATA-Box"; clearly, these sequences are in fact quite variable. A simple promoter detection approach based on the TATA-Box can therefore not be used to detect promoters with high accuracy.

2.4. Experimental data

As described in the previous section consensus sequences from the literature cannot be used to develop a reliable detection algorithm. Instead 1872 experimentally determined promoter sequences were obtained from the eukaryotic promoter database (EPD) [9]. All the work done and all the conclusions made were based on these sequences rather than the literature. The EPD was also the source used by Bajic [1], and the source of their model.

The sequences for the non-promoter regions, including introns, exons and splice sites were extracted from the raw human chromosome data found on the NCBI download section [10]. A program was written to read through the raw files and to use the given annotation data to extract sequences containing all the required DNA sequences.

3. Features and a classifier

3.1. Breaking up the DNA sequence into windows

The fist thing to note is that a DNA sequence cannot be processed as a whole, since they can be millions of bases long. The first step is to break the sequence up into smaller sections, or windows. We have chosen to use windows that are 256 bases long, based on the 250 length windows used by Bajic [1], and due to the fact that promoters of this length could easily be obtained from the EPD [9]. Typical sizes suggested in the literature ([5], [6], [7], [8]) for the TATA-Box range between 8 and 12 bases. A promoter (like the TATA-Box) is typically found a few bases before the actual TSS. Once again the consulted literature disagrees on this distance, with ranges given between 20 and 40 bases. Furthermore the TATA-Box is only one consensus promoter, others such as the CAAT-Box is said to be up to 100 bases upstream of the TSS. The 256 bases should be sufficient to compensate for all these possible promoters and varying lengths. The windows are selected so that there are 206 bases before the actual TSS, and 50 bases downstream of it to capture all the possible features.

For the rest of the paper each window of 256 bases will be referred to as a single sample. Each sample can be one of 5 different classes: Promoter, Intron, Exon, Intron-to-exon splice (I2E) or Exon-to-intron splice (E2I).

3.2. The previous model

As mentioned in the introduction we based our method on the one Bajic [1] used for their system. They have a model that, in short, uses statistics to first determine whether a given sample is likely to be either a promoter, intron or exon section. These are then combined as the three inputs of an ANN to do the prediction.

The first thing to note is that this model doesn't consider the splice sections. We have found that there are

strong features that appear in the splice sites that also occur around the TSS. Thus, a splice site can easily be detected as a TSS.

The second, and probably most important thing to note, is that this model takes into consideration each of the 250 bases in the sample window. The reason for the window length is given in the previous section, but note that this length is only used to make sure that all the possible promoter boxes are included in the window. All 250 bases do not contribute to the detection process.

3.3. Breaking up the window into n-tuples

We want to break up a single sample into smaller windows - firstly, to address the second problem mentioned in the previous section and, secondly, to search for specific identifying features within each sample of 256 bases. A process of base pairing was used to attempt to get more meaningful information from the samples. These base combinations were called n-tuples, where "n" is the number of bases that were grouped together. Thus a 2-tuple simply means that two consecutive bases are considered. With 2-tuples we have 4^2 or 16 possible combinations. N-tuple lengths of 3, 4 and 5 were used for experiments given below. With each increment of n the number of possible types grows exponentially so that there are always 4^n types. With this increased number of possible n-types it was possible to extract statistically useful information from the samples.

3.4. The statistics used

Using the n-tuple method described in the previous section it was possible to gather statistical information on the samples. determined the number of times that a given n-tuple occurred in a specific sample type. The five different classes were then grouped together, and the occurrence of each n-tuple was counted for each position in the sample. This is not a useful statistic yet, as we are more interested in where these n-tuples occur frequently, than whether they actually occur. Thus the number of occurrences of each n-tuple, at each position in each of the classes was counted.

For each n-tuple there are 256-n+1 possible positions in each window, (by position we mean the position (1 to 256 in the window) where the n-tuple starts).

These numbers could the easily be compared to each other to see whether a given n-tuple type at a given position occurs more frequently in one class than in all the others. For example after data extraction it was found that the 3-tuple 'ACG', at position 198, occurs more in promoter samples than in intron or exon samples.

However, the difference in frequency of this triple between the 5 classes is not sufficient for reliable classification of promoters. We have therefore defined an entropy-based measure that allows us to select several features, which can be combined to obtain reliable classification. The entropy of each n-tuple gives one a good measure of how meaningful that n-tuple is. The entropy calculation is shown in Eq. (1).

$$Ent(X,Y) = \ln(\frac{A_{xy}}{n}) + \ln(\frac{B_{xy}}{n}) + \ln(\frac{C_{xy}}{n}) + \ldots \quad (1)$$

Ent(X,Y) is the entropy of n-tuple X at position Y. A_{xy} is the number of times n-tuple X occurs at position Y for class A, where A can be any of the 5 different classes. Similarly B_{xy} is the count for class B, and Cxy for class C. n is the total of $A_{xy} + B_{xy} + C_{xy}$. Using this equation, n-tuples can be extracted that can be used to identify a specific class, since such n-tuples will tend to have lower entropy. This process was repeated for each of the 4^n n-tuples (X), at each of the 256-n+1 positions (Y).

3.5. The new model

Our model implements a two layer ANN as the core detector. The n-tuples identified after entropy calculations are used as the inputs to the ANN. When a 256-length sample is introduced to the system for detection some pre-processing is done. A search is done for each of the selected n-tuples at the corresponding position, and a simple binary input is used, 1 indicating that the feature is present in the sample, and 0 indicating that it is not. The number of inputs depends on how many n-tuples could be identified as significant; each input corresponds directly to a single n-tuple-position feature. The output of each input neuron is connected to the second network layer. A variable number of neurons in the hidden (second) layer were used. The hidden units are then connected to a single output neuron. The output of this neuron is a simple binary value. 1 indicating that a sample is a promoter, 0 indicating that it is not a promoter. The input layer and hidden layer also contain one biasing neuron each. The layout of the network is shown in figure 3.

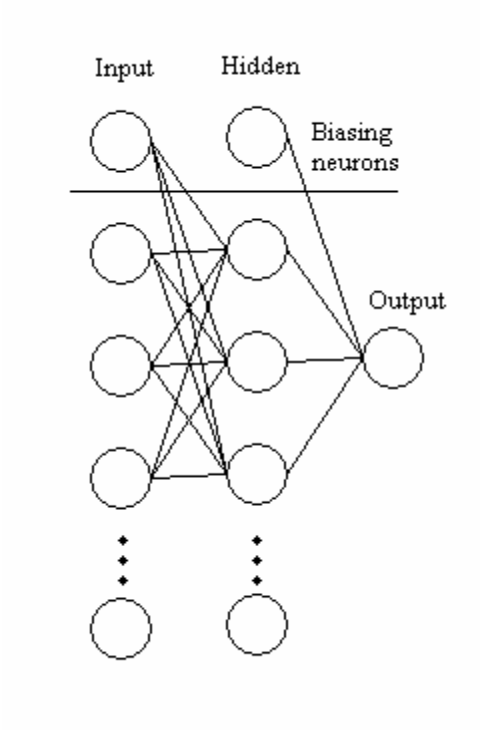


Figure 3: Network layout used.

All the neurons in the network use the sigmoid activation function. The back-propagation training algorithm was used to train the network, with adaptive learning rate and a momentum term.

3.6. Changes in detection

The main difference in the model is that the whole 256-length window is no longer used as input to the system. This is an advantage because all the DNA information that does not contribute to a sample belonging to a certain class is simply discarded. This is represented by the variable number of inputs to the ANN.

The second layer calculates the interaction between these features. This interaction is a very complex process, and is currently poorly understood -- one of the main reasons why an ANN was chosen as a possible solution.

4. Experiments

4.1. Data sets

The samples were separated into three different sets. Each of these sets contained an equal number of samples from each of the 5 different classes. A training set, with 1300 samples of each class was generated. This set was used for the entropy calculations to generate a set of reliable features, as well as to train the actual ANN. The test or tuning set containing 350 samples of each class was used to change and improve the system, for example the number of hidden units used by the ANN was tuned using this set. After training the system was tested on this set. Finally a validation set was generated containing 100 samples of each class. This was used to test the final system after all changes were made.

4.2. Entropy calculations

The next step was identifying features that can be used as inputs to the ANN. The full entropy calculations were done and 54 features were identified that can be used to identify promoters. N-tuples of length 3, 4 and 5 were used, since smaller n-tuples are less significant, and larger n-tuples are too rare to be of value in classification.

4.3. Network testing

The performance evaluation of our network was based on Bajic's [1] method. The network is tested by looking at the positive prediction value (PPV) of the network at a specific sensitivity. The equation for calculating the PPV is given in Eq. (2) and the sensitivity in Eq. (3). TD is the number of true detections, FD is the number of false detections and FR is the number of false rejections.

$$PPV = 100 * \frac{TD}{TD + FD} \qquad (2)$$

$$Sensitivity = 100 * \frac{TD}{TD + FR} \qquad (3)$$

The sensitivity measures how many of the actual promoter samples are correctly detected, while the PPV measures how accurate the total predictions were, and how many non-promoters were identified as promoters.

4.4. The network setup and results

The network was set up using 55 inputs (the 54 features plus 1 bias neuron), connected to 20 hidden neurons. The network was trained using the training set, and then tested. The PPV was measured at different sensitivity levels. The system does not function reliably at very high or very low sensitivity, but with sensitivity between 50% and 75% very good results were obtained. A second experiment was done where the number of non-promoter samples was increased significantly. Only 1300 promoters were used for the training, but the number of non-promoters was increased to 14000. This represents a more realistic relationship since actual promoters take up a small part of DNA material. The PPV drops as expected, since the number of false detections increases, but the number of true detections and false rejections stay the same giving similar sensitivity values. Figure 4 below

shows how our system performs at different sensitivity levels.

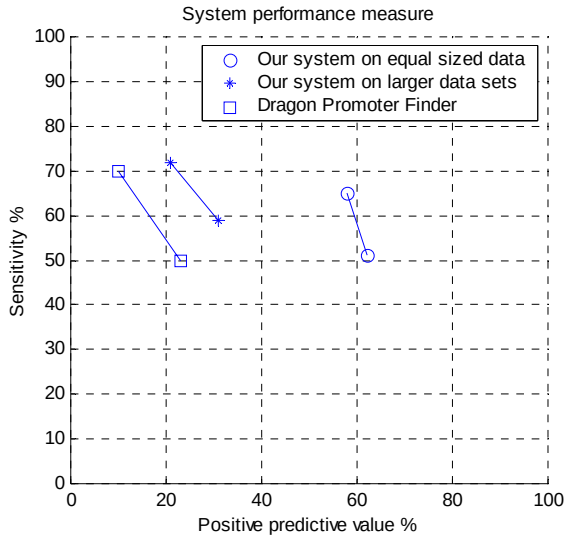


Figure 4: System performance.

The system, Dragon Promoter Finder, developed by Bajic [1] has a PPV of 24% at sensitivity 50%, and of 10% at 70% sensitivity. It can be seen that even with the increased number of non-promoters our system still shows at least a 10% improvement in the PPV. Other known systems perform at 5% PPV, 65% sensitivity, or even as poorly as 5% PPV at only 25% sensitivity. All of these systems are therefore inferior to our system.

5. Conclusion

This article extends the current detection algorithms with improved accuracy obtained by firstly identifying certain parts of the sample that are significant and using only these parts (or rather a combination of these parts) to do the detection. This method eliminates the DNA "garbage" between meaningful sections, and also determines the underlying relationship between meaningful sections that is currently still unknown or un-modelled.

6. References

[1] V.B. Bajic, S.H. Seah, A. Chong, S.P.T. Krishnan, J.L.Y. Koh & V. Brusic, *Computer model for recognition of functional transcription start sites in RNA polymerase II promoters of vertebrates*, Journal of Molecular Graphics and Modelling 21 (2003) 323–332

[2] S. Knudsen, *Promoter2.0: for the recognition of PolII promoter sequences*, Bioinformatics, Vol 15. no. 5 (1999) 356-361

[3] H.F. Lodish, J.E. Darnell & D. Balitmore, *Molecular Cell Biology*, 3rd ed, Scientific American Books, 1995

[4] R.H. Tamarin, *Principles of Genetics*, 6th ed, McGraw-Hill, 1999, 243-279

[5] D. Latchman, *Gene Regulation – A Eukaryotic perspecrive*, 3rd ed, Stanley Thornes (Publishers) Ltd, 1998

[6] T. Beebee and J. Burke, *Gene structure and transcription*, 2nd ed, IRL Press, 1992

[7] R.P. Wagner, M.P. Maguire & R.L. Stallings, *Chromosomes – A Synthesis*, WILEY, 1993

[8] A. Kornberg & T.A. Baker, *DNA Replication*, 2nd ed, WH. Freeman and Company, 1992

[9] R.C. Périer, V. Praz, T. Junier, C. Bonnard & P. Bucher, *The Eukaryotic Promoter Database (EPD)*, Nucl. Acid Res. 28 (2000) 302-303

[10] ftp://ftp.ncbi.nih.gov/genomes/H_Sapiens

Matching Feature Distributions for Robust Speaker Verification

Marshalleno Skosan, Daniel Mashao
Department of Electrical Engineering, University of Cape Town
Rondebosch, Cape Town, South Africa
mskosan@crg.ee.uct.ac.za daniel@eng.uct.ac.za

Abstract—**In this work we improve the performance of a speaker verification system by matching the feature vector distributions obtained when training and testing the system. In particular, we perform experiments using speech that has been degraded by telephone transmission. Speaker Verification experiments are performed on the NIST 2000 database. Significant improvements, above the baseline, are reported.**

***Index Terms*—Speaker verification, Histogram Equalization, Gaussian mixture models**

1. INTRODUCTION

SPEAKER verification (**SV**) is concerned with verifying that an individual is who he/she claims to be. In ideal conditions speaker verification systems perform extremely well. However, as soon as these systems are exposed to real-world conditions, their performances degrade considerably [4]. From a statistical point of view, these degradations in performance can be attributed to the mismatch between a particular speaker's training and testing data distributions caused by the exposure to real-world conditions. In this work, we improve SV performance by using a technique that has its origins in digital image processing. The technique is known as histogram equalization and is used here to optimally minimize the mismatch between training and testing distributions. Experiments are performed on the telephone degraded NIST 2000 speech database. Large improvements, above the baseline system, are reported. In addition, we show that histogram equalization outperforms two commonly used normalization techniques namely, cepstral mean normalization and mean and variance normalization.

2. AN OVERVIEW OF SPEAKER VERIFICATION

There are many papers that provide extensive overviews of speaker recognition research (eg [1, 2, 3, 4]). This section summarizes some of the concepts discussed in these papers. Fundamentally, an SV system needs to make a 2-class decision. That is, to either accept or reject the current identity claim. Figure 1 depicts a typical SV system. Here the system must decide whether the input speech signal better matches a model of the claimed speaker or a background model of non-claimant speakers (imposters). Features extracted from the front-end processing unit are compared to the claimed speaker model and to the background model.

Following this a likelihood ratio statistic $\Lambda(X)$ is computed as the ratio (or difference in the log domain) of these scores. This value is then compared to a decision threshold θ to determine whether to accept or reject the current identity claim.

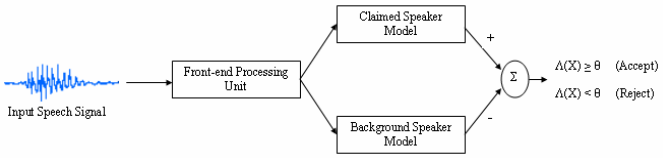


Figure 1: A typical speaker verification system

An SV system can make two types of errors, i.e. it can falsely accept imposters (**FA**) and falsely reject true identity claims (**FR**). In practice, a detection error tradeoff (**DET**) curve is used to illustrate the tradeoff between FA and FR errors as the decision threshold is adjusted. The equal error rate (**EER**) is the point on a DET curve where FA = FR and is used and as a single performance indicator for these two types of error. Another performance indicator that is often used in speaker verification research is the detection cost function (**DCF**) [2, 17]. The is DCF is the weighted arithmetic mean of the FA and FR rates and is defined as

$$DCF = C_{FR} \bullet P_{FR} \bullet P_{\text{true speaker}} + C_{FA} \bullet P_{FA} \bullet P_{\text{imposter}}$$

Cost of a false reject	- C_{FR}	= 10
Cost of a false accept	- C_{FA}	= 1
A priori probability of a true speaker	- $P_{\text{true speaker}}$	= 0.01
A priori probability of a false speaker	- P_{imposter}	= 0.99
Probability of false accept	- P_{FA}	
Probability of false reject	- P_{FR}	

The minimum value of the DCF is usually computed over all operating points (as the decision threshold is varied).

3. HISTOGRAM EQUALIZATION

In many pattern recognition tasks, improvements in performance can be expected if one reduces the mismatch between training and testing conditions. In speaker recognition (**SR**) systems this mismatch can to a large extent be attributed to varying ambient conditions, speech acquisition equipment and transmission

channels [3]. One way of reducing this mismatch is by defining transformations that normalize feature distributions obtained during the training and testing of an SR system. Two such transformations are cepstral mean normalization (**CMN**) and mean and variance normalization (**MVN**). CMN is a channel compensation technique that has successfully been used to reduce the convolutional effects of telephone channels on input speech signals [5]. CMN however, also has the dual effect of normalizing the mean of each speaker's training and test data distributions [5]. It does this by using the following transformation

$$x_{new} = x_{old} - \mu_{x_{old}} . \tag{1}$$

MVN, on the other hand, uses the transformation given in equation (2) to normalize not only the means but, the variances of these distributions as well [6]

$$x_{new} = \frac{x_{old} - \mu_{x_{old}}}{\sigma_{x_{old}}} . \tag{2}$$

In equations (1) and (2), $\mu_{x_{old}}$ is the global mean of the variable x_{old} for a particular utterance, whereas $\sigma_{x_{old}}$ is the standard deviation. However, these techniques are linear and can thus not adequately compensate for the non-linear effects caused by telephone transmission.

To this end, a technique known as Histogram Equalization (**HEQ**), which is used extensively in digital image processing [7] and, which has recently been applied to speech recognition with great success [8, 9], is applied in this research. The aim of HEQ is to completely match the distributions of the training and test data, not just the mean and/or variance (like CMN and MVN) [10]. It does this by non-linearly transforming the probability distribution of a particular speaker's feature vectors, obtained during training and testing, into a reference distribution.

The formulation of HEQ is as follows [11, 12, 13, 14]: Let x_0 be a one-dimensional variable with a probability distribution $p_0(x_0)$. Let $x_1 = T(x_0)$ be a single-valued and monotonically increasing transformation that converts the probability distribution $p_0(x_0)$ into a reference probability distribution $p_{ref}(x_1)$. In other words, it is a transformation that makes the probability of finding x_0 in a differential range dx_0 equal to the probability of finding x_1 in the corresponding range dx_1 i.e.

$$p_{ref}(x_1)dx_1 = p_0(x_0)dx_0 \tag{3}$$

Thus the transformation $x_1 = T(x_0)$ modifies the original probability distribution $p_0(x_0)$ according to the expression

$$p_{ref}(x_1) = p_0(x_0)\frac{dx_0}{dx_1} = p_0(G(x_1))\frac{dG(x_1)}{dx_1} \tag{4}$$

where $G(x_1)$ is the inverse transformation of $T(x_0)$.

Using equation (4), the relationship between the cumulative probabilities associated with $p_0(x_0)$ and $p_{ref}(x_1)$ is given by

$$\begin{aligned} C_0(x_0) &= \int_{-\infty}^{x_0} p_0(x_0')dx_0' \\ &= \int_{-\infty}^{T(x_0)} p_0(G(x_1'))\frac{dG(x_1)}{dx_1}dx_1' \\ &= \int_{-\infty}^{x_1} p_{ref}(x_1')dx_1' \\ &= C_{ref}(x_1) \\ &= C_{ref}(T(x_0)) \end{aligned} \tag{5}$$

Thus the transformation $T(x_0)$ can be obtained as

$$T(x_0) = C_{ref}^{-1}(C_0(x_0)) \tag{6}$$

where C_{ref}^{-1} is the inverse of the cumulative distribution function of the reference probability density function (**PDF**).

For practical implementations only a finite number of observations are available. As a result, cumulative histograms instead of cumulative probabilities are used. This is the reason that the transformation is called histogram equalization and not probability distribution equalization. The transformation in equation (6) cannot however be easily be applied to the multi-dimensional feature vectors obtained from the signal processing front-end of speaker recognition systems. As a result, it is assumed that the all the dimensions of the feature space are independent. Under this simplifying assumption, the transformation can be applied to each feature space dimension independently. A graphical illustration of the transformation is depicted in the figure 2. It shows how the cumulative histograms of the original variable and the transformed variable (the reference cumulative histogram) can be used to perform the transformation. Here each test/training set value x_0 is replaced the value x_1 that corresponds to the same point in the reference cumulative histogram. This illustration shows that HEQ is computationally attractive as it can be implemented by using a simple look-up table.

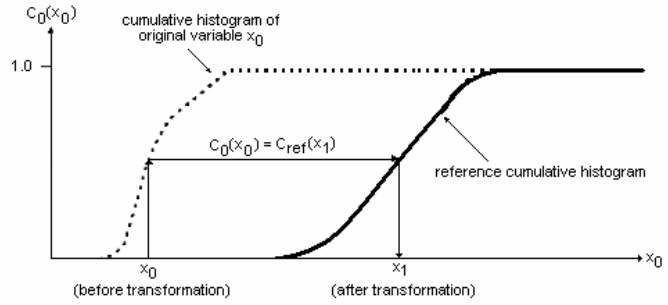


Figure 2: The histogram equalization transformation

4. THE SPEECH DATABASE

Moreno [15] states that both stationary and non-stationary noises can be encountered in a telephone network. Stationary noise appears in the form of low frequency tone-like signals, or white noise caused by thermal and other physical phenomena. He goes on to state that these single frequency noises can be produced by the harmonics power lines and by signaling tones that get transmitted by error through the telephone channel. Non-stationary noises on the other hand can be attributed to clicks and other transient phenomena caused by intermittent connections. As a result, evaluating histogram equalization on speech degraded by telephone transmission will give one a true idea of its ability to compensate for both linear and non-linear distortions.

In a previous contribution [20], we evaluated HEQ on a speaker identification task using the NTIMIT database. This database contains phonetically rich speech that was captured in a sound booth during a single session. The speech was then transmitted through a carbon-button telephone handset and recorded over local and long distance telephone loops [21]. Although HEQ was shown to outperform CMN and MVN, the effect of conversational-like speech, different telephone handsets and various periods of intersession could not be evaluated using this database.

In this work we evaluated the performance of CMN, MVN and HEQ on the NIST 2000 speaker recognition evaluation database [16, 17]. This database includes conversational telephone-quality speech taken from the Switchboard 2 corpus. The test segments are recorded from calls made from a telephone number that is different from the one used to enroll. Therefore, all test utterances may be considered to be collected using a different handset than the one used for training the speaker models. Each speaker model is trained using a single two minute session of speech, while testing utterances range between 15 and 45 seconds. This database allows one to evaluate speaker verification systems under very challenging real-world conditions as the speech, in addition to being degraded by telephone transmission, is also affected by the use of different handsets, different periods of intersession, conversational speech and different test segment lengths. We used this database to perform 1561 true speaker trials and 15501 imposter trials (all trials consisted of male speakers only).

5. EXPERIMENTAL RESULTS

5.1. The baseline system

In this work the front-end processing unit extracts mel-frequency cepstral coefficients (**MFCC**) from the input speech signal. These features are aimed at emulating the spectral compression applied by the human auditory system to an incoming speech signal [3]. MFCCs are spectrum-based features and are used here as a result of the speech spectrum having been shown to be very effective in speaker recognition (**SR**) research [2]. This is as a result of its ability to provide an adequate representation of an individual's vocal tract structure, which is one of the main speaker dependent characteristics that SR systems use to discriminate between speakers [1].

The MFCCs were generated as follows: the incoming speech signal was first multiplied by overlapping Hamming windows which divided it into a sequence of 20ms frames with an overlap of 10ms between frames. These speech frames were then Fourier transformed into the frequency domain where a sequence of log-magnitude spectra were computed. To obtain the mel-frequency cepstral coefficients, these log-magnitude spectra were filtered by a bank of mel-scaled triangular filters distributed over a bandwidth of 0Hz to 3800Hz. The outputs of the filterbank where then discrete cosine transformed into 30 dimensional feature vectors. In the subsequent experiments, CMN, MVN and HEQ were applied at this stage to modify the distributions of these feature vectors.

In order to model the distribution of feature vectors obtained for each speaker, we used Gaussian mixture models (**GMM**) [4, 18]. A GMM can be viewed as a non-parametric, multivariate PDF model that that is capable of modeling arbitrary distributions and is currently the most dominant method of modeling speakers in speaker recognition research. The GMM of the distribution of feature vectors for speaker S is a weighted linear combination of M unimodal Gaussian densities $b_i^s(\mathbf{x})$, each parameterized by a mean vector $\boldsymbol{\mu}_i^s$ and a covariance matrix Σ_i^s. These parameters are collectively represented by the notation

$$\lambda_s = \{p_i^s, \boldsymbol{\mu}_i^s, \Sigma_i^s\} \quad \text{for} \quad i = 1, \ldots, M \tag{7}$$

where p_i^s are the mixture weights satisfying the constraint

$$\sum_{i=1}^{M} p_i^s = 1 \tag{8}$$

For a feature vector $\mathbf{x}$, the mixture density for speaker S is computed as

$$p(\mathbf{x} \mid \lambda_s) = \sum_{i=1}^{M} p_i^s b_i^s(\mathbf{x}) \tag{9}$$

where

$$b_i^s(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} \mid \sum_i^s \mid^{1/2}} \exp\left(\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i^s)' \Sigma_i^s (\mathbf{x} - \boldsymbol{\mu}_i^s)\right) \tag{10}$$

Given a sequence of feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, which are assumed to be independent, the log-likelihood of a speaker model λ_s is given by

$$L_s(X) = \log p(X \mid \lambda_s) = \frac{1}{T} \sum_{t=1}^{T} \log p(\mathbf{x}_t \mid \lambda_s) \tag{11}$$

For speaker verification, equation (11) is computed for the claimed speaker model as well as for the background model of non-claimant speakers. The difference between these values is termed the likelihood ratio $\Lambda(X)$ and is subsequently compared to a threshold θ to determine whether to accept ($\Lambda(X) \geq \theta$) or reject ($\Lambda(X) < \theta$) the identity claim [4]. In this work we used GMMs with 64 mixtures to model each speaker. These GMMs were obtained from well-trained a background model with a form MAP adaptation according to the work done in [19].

5.2. The effect of CMN, MVN and HEQ

This section evaluates the performance of all the feature normalization techniques discussed in section 3. The various statistics for these techniques (such as the means, standard deviations, probability distributions and cumulative distributions) were estimated on an utterance by utterance basis.

Also, we chose a Gaussian PDF with zero mean and unity variance as the reference PDF for the HEQ technique. Table 1 displays the performance of CMN, MVN and HEQ on the male portion of NIST 2000 database.

Compensation Technique	Equal Error Rate	Relative Improvement	Minimum DCF
No compensation	31.35%	-	0.0843
CMN	24.57%	21.63%	0.0742
MVN	10.76%	65.68%	0.0403
HEQ	**10.16%**	**67.59%**	**0.0389**

Table 1: The effect of the feature normalization techniques

Table 1 clearly illustrates that HEQ performs better than both MVN and CMN but, that MVN outperforms CMN. This result is to be expected as HEQ can be viewed as an extension of MVN which, in turn, can be viewed as extension of CMN. This result emphasizes HEQ's ability to compensate for non-linear distortions of the probability distributions of the feature vectors (as discussed in section 3) which cannot be eliminated by linear methods such as MVN and CMN. However, from table 1 it can be seen that normalization of the variance of the training and testing distributions accounts for the largest improvement in performance and that normalization of other moments improves performance only slightly. The trend of the results obtained in this research corresponds to those reported in [9] and [10] which use CMN, MVN and HEQ to improve the performance of speech recognition systems in noisy environments. In figure 3 we show the significant improvements that can be obtained by minimizing the mismatch between training and testing distributions when speech is obtained in adverse environments.

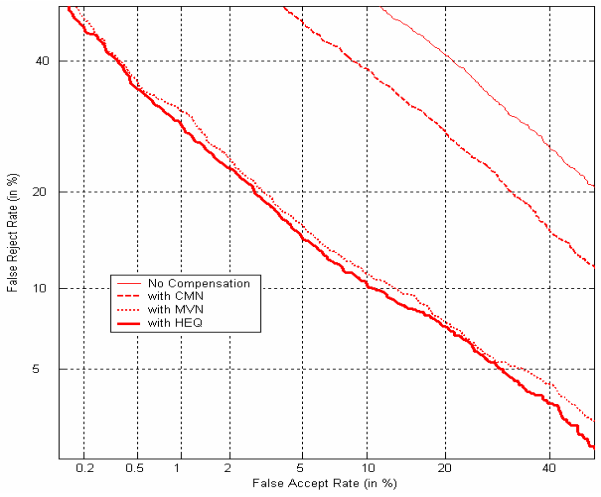


Figure 3: The improvements obtained when applying CMN, MVN and HEQ to minimize the mismatch between training and testing distributions

6. CONCLUSION

In this work we have shown that histogram equalization is very effective in compensating for both linear and non-linear effects caused by the various noise sources encountered in a telephone network. In particular, histogram equalization's ability to match training and testing distributions improved speaker verification performance above the baseline by over 67%.

7. REFERENCES

[1] D.A. Reynolds, "An overview of automatic speaker recognition technology", *Proceedings of IEEE ICASSP*, 4, pp. 4072-4075, 2002.

[2] G.R. Doddington, M.A. Przybocki, A.F. Martin and D.A. Reynolds, "The NIST speaker recognition Evaluation – overview, methodology, systems, results, perspective," *Speech Communication* 31, pp. 225-254, 2000.

[3] J. Campbell, "Speaker Recognition: A Tutorial," *Proc. IEEE*, Vol.85, No.9, pp. 1437-1462, September 1997.

[4] D.A. Reynolds, "Automatic Speaker Recognition Using Gaussian Mixture Speaker Models," *MIT Lincoln Laboratory Journal*, Vol. 8, No. 2, pp. 173-192, 1995.

[5] D.A. Reynolds and R.C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models". *IEEE transactions on Speech and Audio Processing*, vol.3, No.1, January 1995.

[6] R. Duncan, "A description and comparison of the feature sets used in speech processing", Mississippi State University, 2000.

[7] H.D. Cheng and X.J. Shi, "A simple and effective histogram equalization approach to image enhancement", Digital Signal Processing 14, pp.158–170, 2004.

[8] S. Molau, M. Pitz, and H. Ney, "Histogram Based Normalization in the Acoustic Feature Space," *Proc. of ASRU*, December 2001.

[9] A. de la Torre, J. C. Segura, M. C. Ben´itez, A. M. Peinado, and A. Rubio, "Non-linear transformations of the feature space for robust speech recognition", *Proc. ICASSP*, pp. 401–404, 2002.

[10] S. Molau, F. Hilger and H. Ney, "Feature Space Normalization in Adverse Acoustic Conditions", *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. I, pp. 656-659, Hong Kong, China, April 2003.

[11] A. de la Torre, A.M. Peinado, J.C. Segura, J.L. Perez, C. Benitez, A.J. Rubio: "Histogram equalization of the speech representation for robust speech recognition". IEEE Transactions on Speech and Audio Processing, Article In Press. 2003

[12] S. Molau "Normalization in the Acoustic Feature Space for Improved Speech Recognition". PhD Dissertation, Aachen, Germany, February 2003

[13] Cepstral domain segmental nonlinear feature transformations for robust speech recognition. J. C. Segura, C. Benítez, Á. de la Torre, A. J. Rubio, J.

Ramírez. IEEE Signal Processing Letters, Vol. 11, no. 5 may 2004, pp.517-520.

[14] S. Dharanipargda and M. Padmanabhan, "A Nonlinear Unsupervised Adaptation Technique for Speech Recognition", in Proc. ICSLP 2000 Peking, china, October 2000, pp. 556-559

[15] P.J. Moreno, “Speech recognition in Telephone Environments”, MSc dissertation, Carnegie Mellon University, Pittsburg, Pennsylvania, December 1992.

[16] http://www.itl.nist.gov/iad/894.01/tests/spk/2000/doc/spk-2000-plan-v1.0.htm Accessed: 21/09/2004

[17] M.A. Przybocki and A.F. Martin, “Odyssey Text Independent Evaluation Data,” in Proceedings of *2001: A Speaker Odyssey, A Speaker Recognition Workshop,* pp 21- 24, June 2001.

[18] D.A. Reynolds and R.C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models". *IEEE transactions on Speech and Audio Processing*, vol.3, No.1, January pp. 72-83, 1995.

[19] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, “Speaker verification using adapted Gaussian speaker mixture models,” *Digital Signal Processing*, pp. 19-41, 2000.

[20] M. Skosan and D.J. Mashao, “Improving Speaker Identification Performance for Telephone-based Applications”, *Proceedings of the South African Telecommunication Networks and Applications Conference*, September 2004

[21] J. Campbell and D.A. Reynolds, "Corpora for the Evaluation of Speaker Recognition Systems", *Proceedings of IEEE ICASSP*, pp. 2247-2250, 1999.

.

Efficient coding leads to novel features for speech recognition

Willie Smit, Etienne Barnard

Department of Electrical,
Electronic and Computer Engineering
University of Pretoria, 0002
ebarnard@up.ac.za

Abstract

The principle of efficient coding of stimuli can explain the receptive fields in the primary visual cortex and in the primary auditory cortex. When efficient coding is applied to a generative model, it forms biologically realistic basis functions and the code it produces has the spike-like property found in the cortex. We show that this representation can be used for isolated word speech recognition.

We have trained a temporal generative model on spoken single digits. The code from the model is spatiotemporal and spike-like, and we used a k-nearest neighbour classifier to classify this code. The network is able to classify 92% of test samples correctly.

1. Introduction

There is good evidence that properties of neurons in the primary visual cortex and in the primary auditory cortex can be understood in terms of efficient coding of natural stimuli [1, 2, 3, 4, 5]. Such an efficient code gives rise to natural features.

There are two approaches to find an efficient code: independent component analysis (ICA) and sparseness. Both approaches yield a similar code as both solve the same problem, but with slightly different assumptions [5].

ICA gives rise to basis functions resembling the receptive fields in the primary visual cortex [1, 2], but it does not produce a spike-like code as is seen in the cortex. Sparse coding on the other hand explains both the visual receptive fields [1, 5] and the spiking activity of neurons in the primary visual cortex [6]. *Sparse coding in time* [6] refers to a code that is sparse over both time and space; with such a code the activity of a channel over time is mostly close to zero, but occasionally it is very high, and the temporal activities of channels are statistically independent.

With a sparse code, a generative model finds the code that will reconstruct the input using a set of basis functions. When the input is temporal, the model needs to consider the temporal nature in order to find a code that is independent over time.

We developed a single word speech recognition model that is based on the efficient coding of stimuli and that uses a temporal generative model to find the code. The code is spike-like and we interpret the activity as spikes. The spike trains are then classified with a k-nearest neighbour classifier.

2. Methods

2.1. Spectrotemporal processing

Sounds are spectrotemporally processed by the primary auditory cortex. We used a 16 channel spectrotemporal representation for the sounds. We further clipped values in the spectrogram that were below a certain threshold, as these values represent inaudible sounds.

2.2. Generative model

The cortex should implement a generative model in order to explain the sensory message. Most authors use a static generative model, where the current state of the model is independent of its previous states. A stimulus X of length N is encoded by M responses a_i, $i = 1, \ldots, M$. The stimulus is a linear sum of M basis functions $\phi_i(t)$ of length N:

$$X(t) = \sum_{i=1}^{M} a_i \phi_i(t), \qquad t = 1, \ldots, N \tag{1}$$

A stimulus can be encoded efficiently with sparse coding. It requires that the responses should be independent. A set of natural stimuli is used to evaluate the independence of the responses; this way the probabilistic nature of the stimuli is also considered.

An over-complete presentation, where there are more basis functions than dimensions in the stimulus, has some advantages. It is more robust to noise and it can produce an even sparser code. When an over-complete representation is used, there are infinitely many solutions for the code. This means that other criteria can be used to find the "best" code. Sparseness is a good choice for the reasons mentioned earlier.

To encode a stimulus of length K with basis functions of length N, it is usually assumed that the stimulus is divided into $K - N + 1$ sections of length N, and each

section is encoded independently from the other sections. This leads to a redundant representation because the responses $a_i(t)$ at time t contains information about the stimulus in the window $[t-N, t-1]$, it is not necessary for the responses at time $t+1$ to duplicate the stimulus in that window. That is why Lewecki [4] and Klein et al. [3] found that the model in (1) forms similar basis functions at different delays. We use a temporal generative model [6] that eliminates this form of redundancy.

In this model, the past activity of channels also contribute to the current state of the model.

$$X(t) = \sum_i \sum_T a_i(T)\phi_i(T-t) \tag{2}$$

$$\phi_i(t) = \begin{cases} \Re & \text{if } 0 \le t < \Delta t_\phi - 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

For a multichannel input the basis functions have as many channels as the input. The generative model then becomes:

$$X_j(t) = \sum_i \sum_T a_i(T)\phi_{ij}(T-t) \tag{4}$$

with j the subscript to the input channel number.

A set of basis functions is over-complete when there are more basis functions than channels, in which case we use sparse coding to find the "best" code. The sparseness of a temporal code can be quantified as:

$$\sum_T \sum_i S(a_i(T)) \tag{5}$$

$$S(a) = \ln(1 + (a/\sigma)^2) \tag{6}$$

where σ is a constant (we used $\sigma = 0.1$). The sparseness measure will be small when the code is sparse. Model 4 aims to find a code that will reconstruct the input by using a linear combination of the basis functions. The problem of finding such a code that is also sparse can be expressed as an optimization problem with the following cost function:

$$E = \lambda \sum_T \sum_j [X_j(T) - R_j(T)]^2 + \sum_T \sum_i S(a_i(T)) \tag{7}$$

where $R(T)$ is the reconstructed signal, λ is a constant that adjusts the balance between the reconstruction error and sparseness (we used $\lambda = 33.3$). The generative model allows for the code to have positive and negative values. This property does not agree with a spike-like code as spikes signal events [7]. We can force the code to only take on positive values by means of an inequality constraint $a_i(T) \ge 0$. We solved the optimization problem using the LFOPC optimization algorithm [8] as it is robust, it quickly moves close to a minimum and it does not get stuck in small local minima. The derivative of the code with respect to the cost function is:

$$\frac{\partial a_i(t)}{\partial E} = -2\lambda \sum_T \sum_j [X_j(T) - R_j(T)]\phi_{ij}(t-T) + S'(a_i(t)) \tag{8}$$

There exists a set of basis functions that will minimize the cost function for a given set of stimuli. This set of basis functions can be found by making Φ part of the design variables of the optimization problem. Each iteration of this optimization problem is then solved in two steps; during the first step the code that minimizes the cost function given the set of basis functions is found. Then the basis functions are adapted to further minimize the cost function given the code. The second step was solved with the golden section method by searching along the direction of steepest descent. The derivative of the basis functions with respect to the cost functions is:

$$\frac{\partial E}{\partial \phi_{ij}(t)} = -2\lambda \sum_T [X_j(T) - R_j(T)]a_i(t+T) \tag{9}$$

The norm of the basis functions will grow without bounds, because the greater their norm, the smaller the values that the code require. The basis functions were rescaled after each iteration to have a norm of 0.1.

2.3. k-Nearest neighbour classifier

Once the code a has been found, it can be used as features for a classifier. We used the maximum value, A_i, of each channel as a feature, and also the time, τ_i, at which the maximum value occurs. The distance measure is the sum of two components, $d = d_A + d_\tau$, one for the maximum values and one for times at which these values occur. The components were calculated as follows:

$$d_A = \parallel A - A^{train} \parallel \tag{10}$$

and

$$d_T = \parallel \tau - \tau^{train} \parallel \tag{11}$$

The distance d_T should be independent of translations of either pattern. This can be accomplished by translating one pattern so that its temporal mean coincide with that of the other pattern:

$$\bar{D} = \tau - \tau^{train} \tag{12}$$

$$\bar{D}_{coincide} = \bar{D} - \text{mean}(\bar{D}) \tag{13}$$

Usually not all the channels are active for a sample; this means that for the inactive channels the times at which the maximum values occur are undetermined. The entries in $\bar{D}_{coincide}$ that correspond to channels that are inactive for both samples should be set to 0, and a penalty should be added for each such channel. The time component is now:

$$d_T = \parallel \bar{D}_{coincide} \parallel + \eta N_{XOR} \tag{14}$$

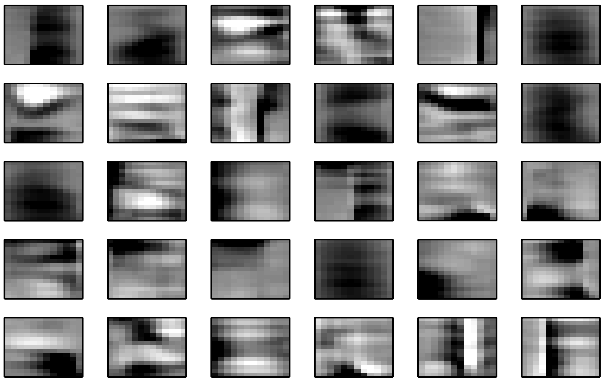


Figure 1: The basis functions after training. The y-axis gives the frequency. The x-axis gives the time. (Basis functions span 250ms).

where the penalty is η (we chose $\eta = 20$), and N_{XOR} is the number of channels that should be penalized.

Finally the distance should be normalized:

$$d = \alpha \frac{d_A}{\sum A + \sum A^{train}} + \frac{d_T}{N_{spikes} + N_{spikes}^{train}} \quad (15)$$

where N_{spikes} and N_{spikes}^{train} are the number of active channels for each pattern, α is a weight (we chose $\alpha = 100$) to set the balance between d_A and d_T.

3. Results

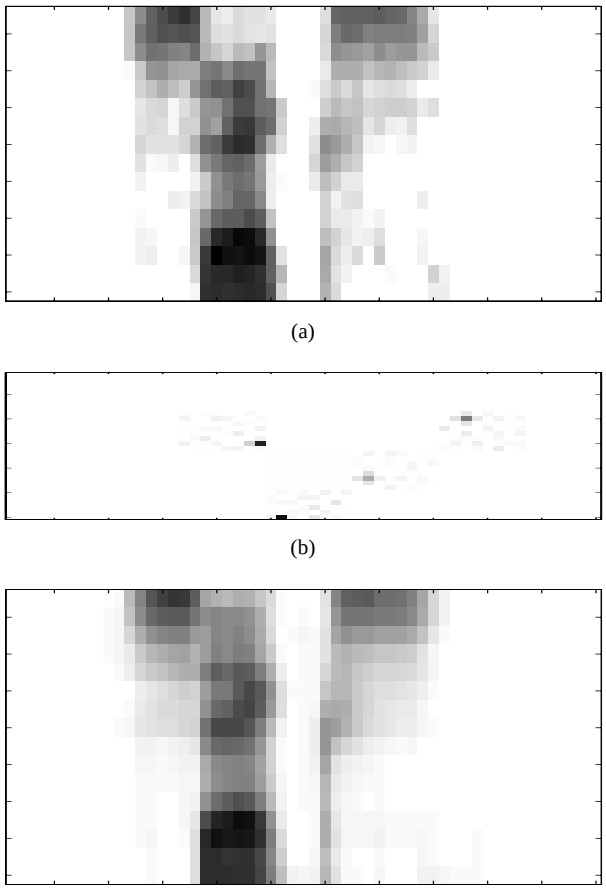
The recognition model needs to be trained in two steps. First the basis functions that will give a sparse code in time must be found and then the classification network has to be trained.

The first step is unsupervised. We used all the single digits ("oh", "zero", "one",... ,"nine") from the TIDIGITS training database for training [9]. The basis functions that were formed are given in figure 1.

With a distance measure that does not include the d_T term, we found a correct classification rate of 92% of the test samples. With the d_T term included, the classification dropped to 91% of the test samples. The d_T term does not appear to improve performance of isolated word recognition, but it may help with continuous speech recognition, especially when the same word is spoken repeatedly.

4. Conclusion

We have presented a simple speech recognition model based on recent research into the efficient coding of stimuli. From a pattern recognition perspective the model learns, without supervision which features to extract. The generative model then uses these features to reduce the dimensions of the input by transforming it into a sparse spatiotemporal code. The code gives the times at which a specific feature is present in the signal.



(a)

(b)

(c)

Figure 2: (a) A spectrogram of the word "six" ($X(t)$). (b) The sparse code ($a(t)$) for the word in figure 2(a) using the basis functions in figure 1. Notice that only four of the basis functions (1, 9, 16, 21) are used to reconstruct the spectrogram. (c) The spectrogram after it has been reconstructed ($R(t)$) from the sparse code.

We have showed that the temporal code can be used to do isolated word recognition. Since the code is temporal, it should be possible to use this code as the input to a continuous speech recognition system. The challenge is to develop a system that will be able to use the temporal code.

Sparse coding can also be applied to recognition problems in vision, or for that matter any other sensory domain. It is a promising approach to recognition problems, and the results researchers have obtained so far indicates that the cortex performs sparse coding.

5. Acknowledgements

The authors would like to thank the National Research Foundation for financial support (Grand number 2053242) and Intel for sponsoring the Ipercube.

6. References

[1] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, pp. 3327–3338, 1997.

[2] J. H. V. Hateren and D. L. Ruderman, "Independent component analysis of natural images sequences yield spatiotemporal filters similar to simple cells in primary visual cortex," *Proceedings of the Royal Society of London*, vol. 265, pp. 2315–2320, 1998.

[3] D. J. Klein, P. König, and K. P. Körding, "Sparse spectrotemporal coding of souds," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 659–667, 2003.

[4] M. S. Lewicki, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, 2002.

[5] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[6] B. A. Olshausen, *Probabilistic Models of the Brain: Perception and Neural Function*. MIT Press, 2002, ch. Sparse Codes and Spikes, pp. 257–272.

[7] P. M. Hoyer, "Modeling receptive fields with non-negative sparse coding," *Neurocomputing*, vol. 52–54, pp. 547–552, 2003.

[8] J. A. Snyman, "The LFOPC leap-frog algorithm for constrained optimization," *Computers and Mathematics with Applications*, vol. 40, pp. 1085–1096, 2000.

[9] L. D. C. (LDC), "NIST CD-ROM version of the Texas Instruments-developed studio quality speaker-independent connected-digits corpus," lDC catalog no.: LDC93S10.

Using high-level and low-level feature concatenation for speaker identification

Brodwyn L. Appanna, Marshalleno Skosan, Daniel J. Mashao

Department of Electrical Engineering
University of Cape Town, Rondebosch, 7700, South Africa

bappanna@crg.ee.uct.ac.za mskosan@crg.ee.uct.ac.za daniel@eng.uct.ac.za

Abstract

Traditional and current speaker recognition systems primarily use low-level (physiological) features of speech that model the physical dimensions of the vocal tract. The popular MFCC is such a feature vector. There is a growing trend in the literature, however, that evidently supports the idea of improved systems by fusing low-level features with high-level (psychological) features like conversational, lexical, phonemic and prosodic patterns found in speech.

In this work we investigated the performance of a speaker ID system evaluated on the NTIMIT database employing the popular MFCC feature vector concatenated with a high-level feature vector containing prosodic information, viz. voicing and pitch. The vector contains the maximum autocorrelation values of a segmented frame of speech and is accordingly named the MACV feature. This paper is an extension of the work done by Wildermoth and Paliwal [11] who reported on an improved speaker ID system that used a fused LPCC-MACV feature set instead of a LPCC-only system.

Results presented in this paper showed an improvement from 82.74% to 85.32% for the fused system, a relative improvement of over 3% for the identification rate. This result corroborated with Wildermoth and Paliwal's [11] performance (an increase from 78.4% to 86.8%) and supports literature on improved recognition systems due to high-level low-level feature fusion. The increase in performance on a popular, state-of-the-art feature vector, like the MFCC, further creates anticipation for promising results to future work on similar systems used on more challenging databases.

1. Introduction

"There are two main sources of speaker-specific characteristics of speech: physical and learned" [1]. The former is based on the alteration of an acoustic wave's frequency content as it passes through the vocal tract. The resonances of the vocal tract (formants), determined by its physical dimensions, modifies the acoustic wave's spectrum [2], [3]. On the other hand, the latter speaker-specific characteristics are psychological or habitual rather than physiological. They include features like conversational, lexical, phonemic and prosodic patterns found in speech [3]. Speaker recognition systems make use of speaker-specific characteristics by employing feature vectors extracted from the speech signal. Subsequently, two main categories of features arise, viz. physiological and psychological or low-level and high-level [4].

The vast majority of speaker recognition systems are based primarily on using low-level spectral features that model a person's vocal tract shape via Gaussian Mixture Models (GMMs) [5]. Generally these systems, especially state-of-the-art, rely on the mel-frequency cepstral coefficient (MFCC) feature extraction technique [6]. Such systems perform very good under clean conditions and acceptable under noisy matched conditions. Under mismatched conditions, however, performance significantly deteriorates [7]. One of the principal reasons for poor performance in these conditions is because of the nature of low-level features; being spectral, they are susceptible to spectral variations due to noise and channel effects [4].

Recent studies have shown that by incorporating high-level features of speech into the conventional system, the performance is improved [2-4], [8-10]. This also makes sense practically when considering the way humans use such patterns to recognize speakers, e.g. identifying impersonations.

Prosodic features are among the most common high-level feature used in such fusion-system research [8]. Prosodic features are known to carry speaker-specific information like melody, intonation and loudness. They are sometimes referred to as source features because they originate at the glottal source [11]. Melody and intonation, comprising a major segment of prosody, are parameterized by the pitch (fundamental frequency – F0) [9]. Many past efforts using stand-alone pitch features returned unimpressive results. The main reason for such performance was attributed to poor, unreliable pitch estimation methods [11].

Wildermoth and Paliwal [11] presented a technique called the Maximum Auto-Correlation Values (MACV) that extracted pitch and voicing information from the speech signal. They investigated the feature vector in a speaker identification environment using the TIMIT, NTIMIT and IISC databases. The performance of the SID system using only a MACV feature vector was poor, although it was an improvement on systems using conventional pitch features only. Results were greatly improved, though, when MACV was combined with a cepstral feature vector, viz. the LPCC feature vector. On all databases there was a significant improvement with the fusion system than with LPCC alone. Experiments on the NTIMIT database, for example, yielded an ID rate improvement of 78.4% to 86.8% (a relative improvement of 10.71%).

Sanderson and Paliwal [10], [12] extended the application of this feature-fusion technique to a speaker verification system, concatenating the MFCC vector with MACV and obtained similar improved performances.

In this paper, we further explore the work done in [11] by also working in an identification scenario but with the exception of concatenating the more common MFCC feature vector with MACV. Experiments are investigated on the NTIMIT database.

We use the same concatenation technique used in [11] with the aim of verifying its superior performance to the MFCC extraction technique alone. The primary objective is to corroborate the increase in robustness and performance of a speaker identification system by combining current, good performing low-level features with high-level features.

2. Overview of speaker identification

Speaker identification, together with speaker verification, makes up the larger discipline of speaker recognition. Speaker Identification is concerned with recognizing an individual from a group of speakers based on a sample of his/her speech, whereas speaker verification is concerned with verifying that an individual is who he/she claims to be [13].

In this paper, research is conducted into the area of text-independent speaker identification. This type of speaker identification is concerned with determining who, from a group of known speakers, is speaking, regardless of what is being spoken [14]. In literature this is referred to as closed-set identification as the system must perform a 1: *N* classification, where *N* is the number of speakers enrolled in the system [13]. The speaker identification process can be summarised as follows.

First the system needs to be trained with samples of speech collected from the speakers to be identified. Once this is complete, the system is tested (a speaker is identified) by comparing a speech sample from an unidentified speaker to the speech samples stored by the system and determining who the most likely speaker is [14].

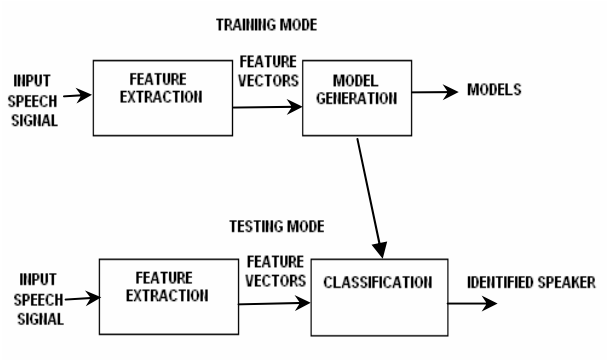


Figure 1: A typical speaker identification system

Figure 1 depicts a typical speaker identification system. As illustrated, it usually consists of three main components. These components perform the following tasks:

The *feature extraction component* is responsible for reducing the amount of data required to represent the input speech signal and minimising sources of noise. It does this while hopefully preserving distinguishing speaker-specific information. The *model generation component* is responsible for creating a model of each speaker's speech characteristics during training. During testing it makes its database of speaker models available to the classification component. The *classification component* is used during testing to compare an unidentified speaker's utterance to the speaker models produced by the model generation component. On the basis of these comparisons, it determines who the most likely speaker is.

The MACV feature set would fall under the feature extraction block and the algorithm to generate the vector follows.

3. The MACV feature vector

Given a speech frame $\{s(n), n = 0, 1,\ldots, N_s - 1\}$, the MACV features are computed as follows [6-8]:

a) Compute the autocorrelation function:

$$R(k) = \frac{1}{N_s}\sum_{n=0}^{N_s-1-k} s(n)s(n+k) \qquad k = 0,\ldots,N_s - 1 \tag{1}$$

b) Normalize $R(k)$ by its maximum value i.e.

$$\hat{R}(k) = \frac{R(k)}{R(0)} \tag{2}$$

c) Divide the higher portion of $\hat{R}(k)$ into M equal parts.

d) Find the maximum value of $\hat{R}(k)$ in each of the M divisions.

e) The M Maximum Autocorrelation Values (MACV) forms an M-dimensional feature vector.

Figure 2 conceptualises the above algorithm.

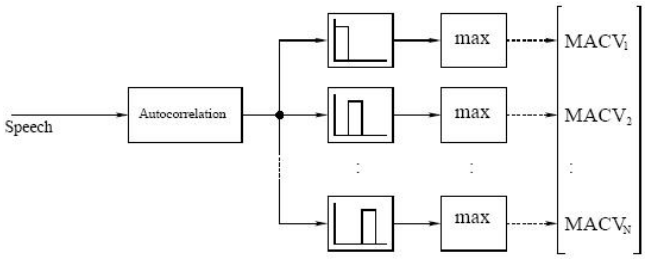


Figure 2: MACV feature extractor (after [10])

It should be noted that the lower portion of the normalised auto-correlation function is not used. It contains information from the vocal tract (system component of speech) which is already extracted by the MFCC vector to which the MACV will be concatenated. The higher portion of the normalised auto-correlation function was based on the fact that human pitch frequency is typically between 60-400Hz (males: 60-160Hz; females: 160-400Hz) which translates into a range from 2ms to 16ms [11].

4. The speech database

All experiments in this research use the NTIMIT speech database. This database contains phonetically rich speech that was captured in a sound booth. The speech was then transmitted through a carbon-button telephone handset and recorded over local and long distance telephone loops. The type of noise contaminating the speech database is thus mainly caused by telephone transmission effects [15, 16]. The NTIMIT database consists of 630 speakers each having spoken 10 utterances of about 3 seconds each. The

first two utterances, labeled as the sa# utterances, are common across all speakers. The next eight are all different and are labeled as the si# and sx# utterances. As a result, this database allows one to evaluate the performance of a text-independent speaker identification system using short testing and training times on telephone-quality speech.

Over the years, the NTIMIT database has been used extensively in speaker recognition tasks [15, 17]. Recently, however, researchers have criticised the NTIMIT database since the speech samples that it contains are actually read sentences which have been recorded in a single session [18]. As a result, effects caused by intersession, handset microphones and conversational speech cannot be examined with this database. Since the work presented in this research is in its early stages and is meant to verify observations made by other researchers, the database was deemed adequate in assessing the applicability of MACVs to speaker ID.

5. Experimental Evaluation

In this section results are presented concerning the concatenation of the *M*-dimensional MACV feature vector with the MFCC feature vector extracted from the same portion of speech.

5.1. The Baseline System

In this work the feature extraction component extracts MFCCs as well as MACVs from the input speech signal. The MFCCs were generated as follows.

The incoming speech signal was first multiplied by an overlapping Hamming window which divided it into a sequence of 20ms frames with an overlap of 10ms between frames. These speech frames were then Fourier transformed into the frequency domain where a sequence of log-magnitude spectra were computed. To obtain the mel-frequency cepstral coefficients, these log-magnitude spectra were filtered by a bank of mel-scaled triangular filters distributed over a bandwidth of 0Hz to 3800Hz. The outputs of the filter bank were then discrete cosine transformed into multi-dimensional feature vectors. The MACVs were generated using the algorithm described in *section 3*.

In order to model the distribution of feature vectors obtained for each speaker, Gaussian mixture models (GMM) were used [5], [6]. A GMM can be viewed as a non-parametric, multivariate PDF model that is capable of modelling arbitrary distributions and is currently the most dominant method of modelling speakers in speaker recognition research. The GMM of the distribution of feature vectors for speaker *S* is a weighted linear combination of *M* unimodal Gaussian densities $b_i^s(\mathbf{x})$, each parameterized by a mean vector $\boldsymbol{\mu}_i^s$ and a covariance matrix Σ_i^s. These parameters are collectively represented by the notation

$$\lambda_s = \{p_i^s, \boldsymbol{\mu}_i^s, \Sigma_i^s\} \quad \text{for} \quad i = 1, \ldots, M \tag{3}$$

where p_i^s are the mixture weights satisfying the constraint

$$\sum_{i=1}^{M} p_i^s = 1 \tag{4}$$

For a feature vector $\mathbf{x}$, the mixture density for speaker *S* is computed as

$$p(\mathbf{x} \mid \lambda_s) = \sum_{i=1}^{M} p_i^s b_i^s(\mathbf{x}) \tag{5}$$

where

$$b_i^s(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_i^s \mid^{1/2}} \exp\left(\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i^s)' \Sigma_i^s (\mathbf{x} - \boldsymbol{\mu}_i^s)\right) \tag{6}$$

Given a sequence of feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, which are assumed to be independent, the log-likelihood of a speaker model λ_s is given by

$$L_s(X) = \log p(X \mid \lambda_s) = \frac{1}{T} \sum_{t=1}^{T} \log p(\mathbf{x}_t \mid \lambda_s) \tag{7}$$

For speaker identification, equation (7) is computed for the model of each speaker enrolled in the system. The identity of the speaker associated with the highest scoring model is then returned as the identified speaker. In this work GMMs with 32 mixtures to model each speaker were utilised.

5.2 Experimental Results

Experiments were primarily performed to verify that appending MFCC feature vectors with MACV features does indeed improve speaker identification performance. Note that all experiment results were averaged over three runs.

For our experiments we used the 'test' portion of the NTIMIT database consisting of 168 speakers (112 male and 56 female). We used the first eight alpha-numerically numbered sentences of each speaker to train the GMMs and the last two sentences were used to test the system. In Figure 3 we show that by simply appending 5 MACVs to MFCC feature vectors with varying dimensions improves speaker identification performance in all cases. This figure also shows that a 20-dimensional MFCC feature vector results in the highest identification rate both with and without the addition of MACVs. However, the addition of MACVs improved the identification rate from 82.74% to 85.32% - a relative improvement of over 3%.

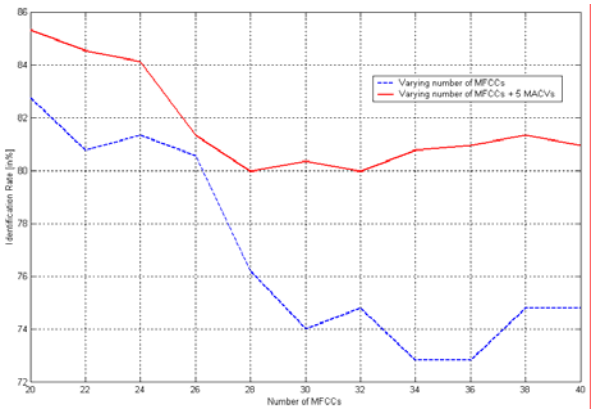


Figure 3: Speaker identification rate versus varying numbers of MFCCs (with and without the addition of 5 MACVs)

5 MACVs was initially chosen as this was the amount of MACVs used by Wildermoth and Paliwal [11]. In order to determine whether 5 MACVs is indeed the optimal number of MACVs to use, we varied the number of MACVs appended to the 20-dimensional MFCC feature vector between 0 and 10. Our results in Figure 4 show that increasing the number of MACVs from 0 to 5 leads to a consistent improvement in performance. However, increasing the number of MACVs beyond 5 degrades system perform. This observation confirms that 5 MACVs leads to the best performance when combined with MFCCs. At this stage, however, it is unclear why this trend in performance exists.

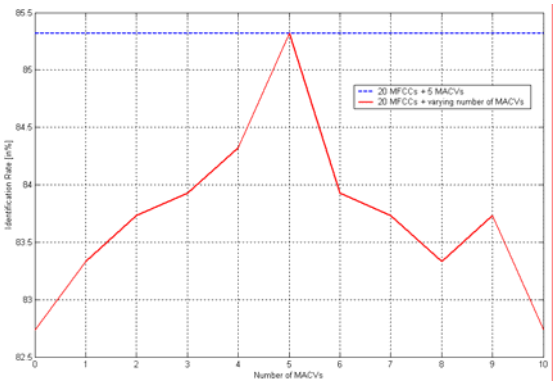


Figure 4: Speaker identification rate versus varying numbers of MACVs (with the addition of 20 MFCCs)

6. Conclusion

The primary objective of this paper was to investigate the performance of a combination of a high-level feature, viz. MACV, and a popular low-level cepstral feature opposed to using only the cepstral feature. In doing so, a secondary purpose arose to extend the work done by Wildermoth and Paliwal [11] who combined MACV and LPCC features in a SID environment; the fusion of MACV and the popular MFCC feature was investigated in this paper in a SID scope on the NTIMIT database.

In this work, a relative improvement of over 3% was observed in the identification rate when a 20-MFCC vector was concatenated with 5 MACVs. This is an improvement on using the 20-MFCC vector alone.

Compared to the work by Wildermoth and Paliwal [11] who used MACV and LPCC, the 3% relative improvement in ID rate was less than their 10%. It is worthy to note, though, that their LPCC-alone system yielded an ID rate of 78.4% whereas the MFCC-alone system investigated in this paper yielded an 82.74% ID rate. So, although the relative improvement rates were different, overall recognition performance was about the same.

The results of this paper supports existing literature that says that the combination of physiological and psychological features improve speaker recognition, specifically speaker ID in this case. The increase in performance on a popular, state-of-the-art feature vector, like the MFCC, creates anticipation for promising results to future work on other similar fusion systems performed on more challenging databases.

7. References

[1] J.P. Campbell, Jr., "Speaker Recognition: A Tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437-62, Sept. 1997.

[2] S. Kajarekar, L. Ferrer, A. Ventkataraman, K. Sonmez, E Shriberg, A. Stolcke, H. Bratt, and R.R. Gadde, "Speaker recognition using prosodic and lexical features," *Workshop on Automatic Speech Recognition and Understanding*, St. Thomas, VI, Nov. 2003.

[3] F. Farahani, P.G. Georgiou, and S.S. Narayanan, "Speaker identification using supra-segmental pitch pattern dynamics," *ICASSP '04*, Montreal, Que., Canada, May 2004.

[4] D.A. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang, "The SuperSID Project: Exploiting high-level information for high-accuracy speaker recognition," *ICASSP '03*, Hong Kong, China, April 2003.

[5] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian speaker mixture models," *Digital Signal Processing*, vol. 10, pp. 181-202, 2000.

[6] D.A. Reynolds, and R.C. Rose, "Robust Text-Independent speaker identification using Gaussian Mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, Jan. 1995.

[7] S. Krishnakumar, K.R. Prasanna Kumar, and N. Balakrishnan, "Pitch maxima for robust speaker recognition," *ICASSP '03*, Hong Kong, China, April 2003.

[8] A.G. Adami, R. Mihaescu, D.A. Reynolds, and J.J. Godfrey, "Modeling prosodic dynamics for speaker recognition," *ICASSP '03*, Hong Kong, China, April 2003.

[9] H. Ezzaidi, and R. Jean, "Pitch and MFCC dependant GMM models for speaker identification systems," *Canadian Conference on Electrical and Computer Engineering*, Niagra Falls, Ont., Canada, May 2004.

[10] C. Sanderson, and K.K. Paliwal, "Joint cohort normalization in a multi-feature speaker verification system," *Proc of 10th Annual IEEE International conference on Fuzzy System*, Melbourne, Vic., Australia, Dec. 2001.

[11] B. Wildermoth, and K.K. Paliwal, "Use of voicing and pitch information for speaker recognition," *Proc. 8th Australian Int. Conf. Speech Science and Technology*, Canberra, 2000.

[12] C. Sanderson, and K.K. Paliwal, "Information fusion for robust speaker verification," in *Proc. Eurospeech '01*, Scandinavia, 2001.

[13] D.A. Reynolds, "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification," *Ph.D. Thesis*, Georgia Institute of Technology, September, 1992.

[14] D.A. Reynolds, "An overview of automatic speaker recognition technology," *ICASSP '02*, Orlando, Florida, May 2002.

[15] H. Gish, and M. Schmidt, "Text-Independent Speaker Identification," *Proc. of IEEE Signal Processing Magazine*, 1994.

[16] D.A. Reynolds, "Large Population Speaker Identification Using Clean and Telephone Speech," *IEEE Signal Processing Letters*, 1995.

[17] P.J. Moreno, "Speech recognition in Telephone Environments," *MSc dissertation*, 1992.

[18] D.J. Mashao, "Auditory-based speaker identification system," *PRASA '01*, 2001.

Automatic intonation modeling with INTSINT

J.A. Louw and E. Barnard

Human Language Technologies Research Group
CSIR / University of Pretoria, Pretoria, 0001
jalouw@csir.co.za ebarnard@up.ac.za

Abstract

Accurate intonation modeling has become a vital part of modern day speech synthesis systems. This is especially true for tonal languages such as isiZulu, where the intonation of an utterance not only influences the perceived naturalness of the synthetic voice, but may also influence its semantics.

In this work we explore the INTSINT intonation modeling algorithm and its application to an isiZulu speech synthesiser. For fundamental frequency an algorithm, MOMEL, for the automatic derivation of a representation as a sequence of target points is applied. A symbolic coding system for fundamental frequency patterns is implemented. We show that the model's level of phonetic representation has the potential to provide an interface between abstract cognitive representations and their physical manifestations, but requires more in-depth phonological information.

1. Introduction

Prosody is the feature of text-to-speech (TTS) systems which is most in need of improvement. An important aspect of prosody is intonation. Intonation is an aspect that can be reliably extracted from the speech signal, and that can be presented in an easily understandable form. In conjunction with appropriate timing information, fundamental frequency contours furnish most of the significant prosodic information contained in speech expressions. It is thus at the center of much current interest in speech analysis and speech modeling.

The body of research into manual and automatic intonation analysis systems and techniques has been growing rapidly in the last few years [1]. Spoken language understanding systems can benefit from the structural and pragmatic information which intonation often conveys. However, current trends in speech processing have increased the need for large corpora, since stochastic speech synthesis methods (including those used for intonation modeling) require a great deal of data to be effective. Manually labeling speech databases for intonation is recognized as difficult and time consuming. Automatic labeling can decrease both time and funds spent on building the databases from which theoretical models and viable applications can be built.

Automatic analysis of intonation is a crucial step towards the long-awaited automatic transcription of intonation. The goal of the research on intonation analysis detailed in this paper is to create a system which can automatically label speech with intonation information. This contribution evaluates a method for creating intonation models from recorded speech. The goal is to predict fundamental frequency contours, given the orthography. The emphasis is on automatic data-driven techniques. Data-driven models can easily be adapted to different speakers, different text styles and different languages.

An important aspect of this work is the selection of universal (language independent) linguistic factors that are important for predicting observed intonation phenomena. These selected linguistic features (e.g. part-of-speech, type of punctuation) are then combined with prosodic features such as word boundary strength, word prominence and phone duration, which were themselves predicted by prosody models.

Phonetic models use a set of continuous parameters to describe intonation patterns observable in an F_0 contour [2]. An important goal is that the model should be capable of reconstructing F_0 contours faithfully when appropriate parameters are given. However, to make it functional, a phonetic model should also be linguistically meaningful. In fact, using certain functions, such as polynomial equations, to accurately represent F_0 contour is not a difficult task. What is more challenging is developing a model whose parameters are predictable from available linguistic information. To be more specific, the mapping from various linguistic factors, which could affect intonation, to the model parameters, or vice versa, is more critical.

2. Background

Below, we first classify intonation models into two major classes [3], and then provide details on a hybrid intonation model which is the focus of the current study.

2.1. Phonological Versus Phonetic Models

A *phonological* intonation model uses a phonological representation of F_0. Such a representation is descriptive and discrete. It uses an inventory of abstract phonological categories, with each category having its own linguistic function. An example is the tonal tier of the ToBI labeling system [4]. ToBI specifies an inventory of tones: one set is used to mark accented syllables, while another set is used to mark phrase boundaries. Each tone marks a different type of accent or boundary.

A *phonetic* model is developed from acoustic (F_0) data. It attempts to describe F_0 movements and it is usually continuous in nature. Often, the description of F_0 movements is linked in some way to the linguistic level. The Tilt intonation model [2], for example, describes pitch accents and boundary tones via rising and falling quadratic functions. A pitch accent can be composed of a rising function, a falling function, or a rising followed by a falling function. Stretches of speech between

intonational events are described by straight-line interpolations. The amplitudes and durations of the rising and falling functions, combined with the position of the pitch accent/boundary tone in the (t, F_0) plane, together constitute the basis for the Tilt description of F_0 contours (5 continuously- valued parameters per event). The Fujisaki model [5] views an F_0 contour as the sum of a base F_0 value, phrase components and accent components (in the log F_0-domain). Phrase and accent components are generated by respectively passing impulse and step functions through second-order filters. The timings and amplitudes of the impulse and step functions constitute the phonetic representation of F_0.

Traber's representation of F_0 [6] merely consists of samples of a smoothed and interpolated F_0 contour. It does keep track of the syllabic structure of the utterance, but it has no other links to the linguistic level.

2.2. MOMEL/INTSINT

The method used in this work can be viewed as a hybrid phonetic/phonological model [7]. It starts with a low-level phonetic analysis technique known as MOMEL (MOdélisation de MELodie). Then a phonological description system, INTSINT (INternational Transcription System for INTonation), is derived from the results of phonetic analysis.

2.2.1. MOMEL

The MOMEL algorithm aims to analyze and synthesize F_0 curves automatically. An F_0 curve is modeled as the superposition of two components: a micro-prosodic component caused by the characteristics of the individual phonematic segments of the utterance and a macro-prosodic component reflecting the choice of intonation pattern for the utterance [1]. The MOMEL algorithm extracts the macro-prosodic component from the F_0 curve and models it as a series of quadratic splines.

There are four basic stages:

1. preprocessing of F_0

   All values more than a given ratio higher than both their immediate neighbours are set to 0. This preprocessing has essentially the effect of eliminating erratic F_0 values.

2. estimation of target-candidates

   The following steps are followed iteratively for each instant *x*.

   (a) Within an analysis window centered on *x*, values of F_0 (including values for unvoiced zones) are neutralised if they are outside a range of thresholds and are subsequently treated as missing values.

   (b) A quadratic regression is applied within the window to all non-neutralised values.

   (c) All values of F_0 which are more than a given distance below the value of F_0 estimated by the regression are neutralised. Steps b and c are reiterated until no new values are neutralised.

   (d) For each instant *x* a target point is calculated from the regression equation. If the target is outside the current analysis window or if the target lies outside the F_0 thresholds, then the target is treated as a missing value.

   Steps b, c and d are repeated for each instant *x*, resulting in one estimated target point (or a missing value) for each original value of F_0.

3. partition of candidates

   The sequence of target candidates is partitioned by means of another moving window, in which the average value of the targets in the first half of the window is compared to the average value in the second half. The boundaries of the partition are then taken as those values which correspond to a local maximum for this distance and which is greater than the overall average value of the distances.

4. reduction of candidates

   Within each segment of the partition, outlying candidates are eliminated. The mean value of the remaining targets in each segment is then calculated as the final candidate for that segment.

2.2.2. INTSINT

INTSINT describes intonation with a limited set of abstract tonal symbols, which is designed such that separate inventories of pitch patterns for different language are not required. The input to the INTSINT system is a series of target points, which is estimated from the acoustic low-level MOMEL modeling technique.

The abstract symbols defined to represent the target points are:

- **T** – Top
- **M** – Mid
- **B** – Bottom
- **H** – Higher
- **S** – Same
- **L** – Lower
- **U** – Up-stepped
- **D** – Down-stepped

Figure 1 shows the abstract symbols used in the INTSINT labeling system.

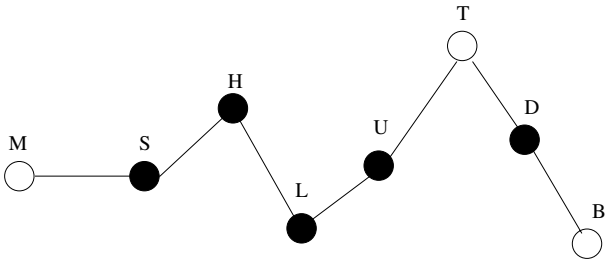


Figure 1: *INTSINT labeling system.*

Among these symbols, tones T, M, B are regarded as absolute tones, which refer to the speaker's overall pitch range.

Tones H, S, L, U, D are relative with respect to the value of the preceding target point. The relative tones are further distinguished between non-iterative H, S, L and iterative U, D tones – the latter can occur repeatedly, whereas the former cannot. An automatic coding scheme is used to relate the target points and the abstract symbols through a set of rules.

1. The highest and lowest target values in the utterance are coded as T and B, respectively.
2. The first target point, as well as any which follow a silent pause of more than a certain length in duration, is coded M (unless already coded T or B).
3. All other target points are coded with relative tones. A target which is less than a given threshold from the previous target is coded S. Otherwise it is coded H, L, U or D according to its configuration with respect to the preceding and following target points as in Figure 1. Where there is no relevant following target point the point is coded as either S, H or L depending on the previous target.
4. The statistical value of each category of target points is then calculated: for absolute tones the mean value is taken, for relative tones a linear regression on the preceding target is calculated.
5. Any target points originally coded H or L can be recoded as T, U, B or D if this improves the statistical model
6. Steps 4 and 5 are then repeated until no more points are recoded.

It is trivial to revert back to a MOMEL curve (which represents the intonation curve) from the INTSINT labels. Thus, if you have the INTSINT labels you can calculate the intonation curve.

3. Implementation

The MOMEL and INTSINT algorithms were implemented as modules into the Festival Speech Synthesis system [8]. The MOMEL algorithm was implemented straightforwardly as described in [7], but the implementation of the INTSINT labeling system required some more work.

The problem that arises with the implementation of the abstract phonological INTSINT labeling system stems from the fact that these labels are derived from a curve that was calculated from phonetic data. The temporal positions of the labels are defined by the inflection points of the quadratic splines calculated with the MOMEL algorithm. This implies that there is no direct connection between the INTSINT labels and the phonological data. Thus, the INTSINT labels must be time aligned with some form of linguistic feature that can be extracted from the orthographic data.

In this work the INTSINT labels were aligned with the syllables of the utterances. The method is as follows:

1. Add an INTSINT label to the mid point of each syllable in an utterance based on the rules as described in Section 2.2.2.
2. Calculate the resulting MOMEL curve from these labels.
3. Change the labels to minimize the difference between the MOMEL curves resulting from the syllable aligned labels and the original non-aligned labels.

The MOMEL curve calculated from the resulting phonologically aligned labels would then be the best approximation to the original MOMEL curve, which represents the intonation pattern of the utterance. From these labels *Classification and Regression Trees* (CART) [9] models were trained.

4. Experimental results

The dataset used in the experiments consisted of 152 isiZulu recordings. These recordings were phonetically transcribed by means of an automatic process and then checked by hand.

The experiments consisted of two parts:

1. Automatic intonation labeling as described in Sections 2.2 and 3.
2. Training and testing of intonation models build from the INTSINT-labeled data.

4.1. Automatic intonation labeling

The INTSINT labels were phonologically aligned to the middle vowel of the syllables in the utterances. Two experiments were conducted. In the first test all syllables were labeled with INTSINT markers, and in the second test only stressed syllables were labeled. The MOMEL curves resulting from these labels were then calculated and compared to the MOMEL curves calculated form the acoustic data. Table 1 shows the mean value and standard deviation of the root-mean-square (RMS) error calculated over the whole test set when compared to the original MOMEL curves.

	All syllables	Stressed Syllables
Mean RMS error	9.28 Hz	20.67 Hz
σ RMS error	2.01 Hz	4.54 Hz

Table 1: *INTSINT labeling accuracy obtained from labeling all syllables and only stressed syllables.*

Figure 2 shows an example of the MOMEL curve as calculated from the acoustics together with the F_0 curve, while Figures 3 and 4 give comparisons of the MOMEL curve as calculated from the acoustics versus the MOMEL curves aligned to all syllables and stressed syllables respectively.

4.2. INTSINT intonation models

Two CART intonation models were trained from the two sets of INTSINT labeled data. The *wagon* [10] CART training program was used for the training. A test set of 10% of the two labeled data sets was not included in the training data. A *stop* value of 50 and a *balance* value of 5 was chosen for the training of the trees.

Table 2 shows the intonation prediction results of a CART, trained on the all syllable labeled data, on the test set. The rows of the table gives the correct INTSINT labels and the columns gives the predicted INTSINT labels (for example, the *Same* label (S) was wrongly predicted as a *Bottom* label 18 times, and the prediction of S was correct in 22 of its 61 occurrences).

The were 301 syllables in total in the test set of which 106 were correctly predicted, giving a correct prediction rate of 35.22%.

Label	B	D	H	L	M	S	T	U	Total	Correct	% Correct
B	35	0	0	5	3	0	0	0	43	[35/43]	81.395
D	6	2	2	15	2	5	0	2	34	[2/34]	5.882
H	3	4	4	5	4	7	1	1	29	[4/29]	13.793
L	15	6	3	8	2	10	0	3	47	[8/47]	17.021
M	0	0	0	0	18	0	1	0	19	[18/19]	94.737
S	18	5	1	10	2	22	1	2	61	[22/61]	36.066
T	0	0	3	2	4	3	14	0	26	[14/26]	53.846
U	5	5	3	13	4	9	0	3	42	[3/42]	7.143
Total	82	22	16	58	39	56	17	11			

Table 2: *INTSINT labeling accuracy obtained from a CART trained on the all syllables data set.*

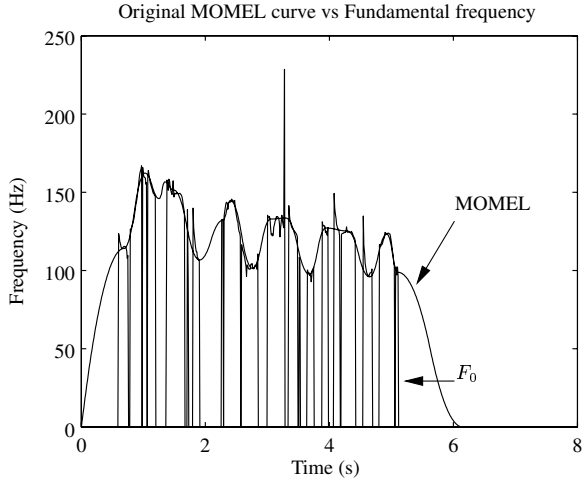


Figure 2: *A comparison between the MOMEL curve as calculated from the acoustics and the F_0 curve.*

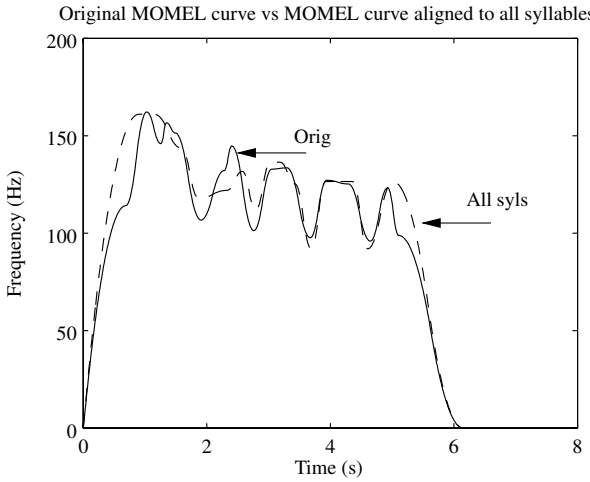


Figure 3: *A comparison between the MOMEL curve as calculated from the acoustics and the MOMEL curve aligned to all syllables.*

Table 3 shows the intonation prediction results of the CART, trained on only stressed syllables in the labeled data, as predicted on the test set. The convention is the same as in Table 2, except that – since only the stressed syllables can have an intonation event – there is an extra column for when there is no intonation label.

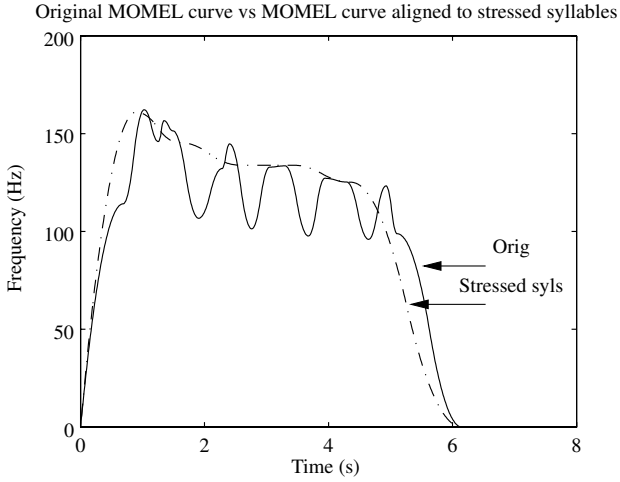


Figure 4: *A comparison between the MOMEL curve as calculated from the acoustics and the MOMEL curve aligned to stressed syllables only.*

The same test set was used as in the previous experiment, thus of the 301 syllables a total of 227 (75.42%) were predicted correctly.

5. Conclusion

From Table 1 and Figures 3 and 4 it is clear that the MOMEL curve resulting from the INTSINT labels aligned on all the syllables in an utterance gives a better approximation to the true MOMEL curve as calculated from the acoustics. From Table 3 we can see that the CART model trained from only the stressed syllables labels produces a higher correct prediction percentage. Closer inspection reveals, however, that the majority of correct predictions were for syllables with no intonation events. If we were to exclude the results of the syllables with no intonation events, the correct prediction percentage would drop from 75.42% to 19.56%. This result is not unexpected: by labeling each syllable with an intonation event, a richer model (with more variation) is produced; as long as this model can track the contour of the true intonation data, it should be able to provide a more detailed fit to that data. Encouragingly, that

Label	0	B	D	H	L	M	S	T	U	Total	Correct	% Correct
0	209	0	0	0	0	0	0	0	0	209	[209/209]	100.000
B	0	4	0	0	2	0	0	1	0	7	[4/7]	57.143
D	0	8	0	0	5	0	0	0	0	13	[0/13]	0.000
H	0	7	0	0	2	0	0	0	0	9	[0/9]	0.000
L	0	8	0	0	8	0	0	0	0	16	[8/16]	50.000
M	0	1	0	0	1	0	0	13	0	15	[0/15]	0.000
S	0	8	0	0	6	0	0	0	0	14	[0/14]	0.000
T	0	1	0	0	0	0	0	6	0	7	[6/7]	85.714
U	0	4	0	0	7	0	0	0	0	11	[0/11]	0.000
Total	209	41	0	0	31	0	0	20	0			

Table 3: *INTSINT labeling accuracy obtained from a CART trained on the stressed syllables data set.*

is the case for the models considered here.

Although the INTSINT model is phonological in nature, the actual labels are derived from a MOMEL curve, which in turn is derived from phonetic data. Unfortunately, this link is still fairly weak, as indicated by the relatively low CART classification accuracies that we have observed. This restricts the overall accuracy of the system, and the intonation produced (even for a stress language such as English) is not particularly natural. Work on additional ways of incorporating phonological and maybe even semantic data into the model is therefore required.

6. Acknowledgements

This work was supported by the CSIR *Information Society Technologies Centre*, South Africa, as well as the *Local Language Speech Technology Initiative* (LLSTI).

7. References

[1] Hirst, D. , "Automatic analysis of prosody for multilingual speech corpora," *Improvements in Speech Synthesis* (Keller, E. , G.Bailly, J.Terken, and M.Huckvale, eds.), Wiley, 2001.

[2] Taylor, P. , "Analysis and synthesis of intonation using the Tilt model," *Journal of the Acoustical Society of America*, vol. 107, no. 3, 2000, pp. 1697–1714.

[3] Garrido, J. , *Modelling Spanish intonation for text-to-speech applications*. PhD thesis, University Autònoma de Barcelona, 1996.

[4] Beckman, M. and Elam, G. , *Guidelines for ToBI labelling, version 3*. Ohio State University, March 1997. www.ling.ohio-state.edu/research/phonetics/E_ToBI.

[5] Möbius, B. , Pätzold, M. , and Hess, W. , "Analysis and synthesis of German F0 contours by means of Fujisaki's model," *Speech Communication*, vol. 13, 1993, pp. 53–61.

[6] Traber, C. , "F0 generation with a database of natural F0 patterns and with a neural network," *Talking Machines: Theories, Models and Designs* (Bailly, G. , Benoît, C. , and Sawallis, T. R. , eds.), pp. 287–304, Amsterdam, The Netherlands: Elsevier, 1992.

[7] Hirst, D. , Cristo, A. D. , and Espesser, R. , "Levels of representation and levels of analysis for intonation," *Prosody : Theory and Experiment* (Horne, M. , ed.), Dordrecht, The Netherlands: Kluwer, 2000.

[8] Black, A. , *Speech Synthesis in Festival*. Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA, 1.4.1 ed., May 2000. A practical course on making computers talk.

[9] Breiman, L. , *Classification and Regression Trees*. Pacific Grove, CA: Wadsworth and Brooks, 1984.

[10] Taylor, P. , Caley, R. , Black, A. , and King, S. , *Edinburgh Speech Tools Library*. University of Edinburgh, Edinburgh, Scotland, 1.2 ed., June 1999. System Documentation.

Increased Diphone Recognition for an Afrikaans TTS system

Francois Rousseau and Daniel Mashao
Department of Electrical Engineering, University of Cape Town, Rondebosch,
Cape Town, South Africa, frousseau@crg.ee.uct.ac.za, daniel.moshao@ebe.uct.ac.za

Abstract – In this paper we discuss the implementation of an Afrikaans TTS system that is based on diphones. Using diphones makes the system flexible but presents other challenges. A previous effort to design an Afrikaans TTS system was done by SUN. They implemented a TTS system based on full words. A full word based TTS system produces more natural sounding speech than when the system is designed using other techniques. The disadvantage of using full words is that it lacks flexibility. The baseline system was build using the Festival Speech Synthesis System. Problems occurred in the baseline due to the mislabeling of diphones and the diphone index. The system was improved by manually labeling the diphones using Wavesurfer, and by changing the diphone index. Wavelength comparison tests were done on the diphone index to show how much of the diphones are recognized during synthesis. For the diphones tested results show an average improvement of 38% in the recognition of diphones compared to the baseline. These improvements improve the overall quality of the system.

Key words**:** Festival Speech Synthesis, diphones, labels, diphone index

1. INTRODUCTION

Afrikaans is the first language to approximately six million people in South Africa. The language originates from seventeenth century Dutch and is influenced by English, Malay, German, Portuguese, French and other African Languages [2]. Together with English it first became the official language in 1925 according to the *Act of 1925*. Previous work on an Afrikaans Synthesizer was made by SUN [1]. The system is embedded within a system called AST (African Speech Technology) which is a hotel reservation booking system that works for Afrkaans, Zulu, Xhosa and English.

TTS (Text-to-speech) in the simplest words is the conversion of text to a speech output using a computerized system. It therefore allows for the communication between humans and machines through synthetic speech [5]. TTS consists of two phases. The first is called high level synthesis also known as the front-end [8]. This is where text analysis and the linguistic analysis are done on the input text. The second phase is called the low level phase, also known as the back-end. This is the phase where prosody is added to the phonetic information gained at the front and where the speech waveform is generated [4]. These two phases are shown in Figure 1.

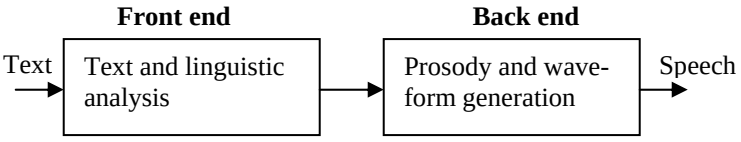


Figure 1: Two phases involved in TTS

Section 2 discusses different techniques of achieving TTS. The most flexible and natural sounding TTS systems are achieved by concatenating short prerecorded speech samples such as phones and diphones to produce synthetic speech. The Festival Speech Synthesis System is a system based on this technique [3]. The system was designed at the Centre for Speech Technology Research (CSTR), at the University of Edinburgh, Scotland [3]. It is an open source system with the ability to be a workbench for the development of new TTS systems [8]. The labeling of diphones together with the diphone index gives the system crucial on the duration of the diphones. This is where problems occur in Festival. The problems are:

(i) the automatic labeling technique label's the diphones incorrectly
(ii) the diphone index is set up to only recognize portions of the diphones.

The methods involved in solving these problems are discussed in Section 3. Section 4 discusses the results of the baseline system and the improvements made by the methods discussed in Section 3. Conclusions on the improved system are discussed in Section 5.

2. TEXT TO SPEECH SYNTHESIS

The first speech synthesis system was build by Christian Kratzenburg in 1779 [5]. The system was able to produce five long vowel sounds using resonators activated by vibrating reads. This breakthrough has lead to the various synthesis techniques available today. The three main techniques are articualtory synthesis, formant synthesis and concatenative synthesis.

The articulatory synthesis method models the human articulators and vocal cords. The advantage of this method is that the vocal tract models allow for accurate models of the transients due to abrupt area changes [4]. The disadvantage is that it is incredibly difficult to implement and hence very rarely used in practice.

Formant synthesis models the formant frequencies of human speech. The advantage is that it has an infinite number of sounds which makes its more flexible than other methods [4].

The disadvantage is the lack of natural sounding synthetic speech. This is due to the fact that usually up to five formants are required for good synthetic speech [4].

Concatenative synthesis connects prerecorded speech units derived from natural speech for synthesis [4]. The sizes of the units vary from phones to diphones to even full words. Using full words has the advantage that it produces very natural synthetic speech. Such systems are however limited to a specific database and hence not flexible, as in the case of the AST system [1]. Building full blown TTS systems based on this method is expensive and time consuming. Using diphones has the advantage that the system is very flexible. Instead of using long prerecorded speech units such as words this method uses diphones, which are the possible phone-to-phone transitions for the language. For example the word '*hello'* would be made up of the diphones '*h-e'* and '*l-o'*. By theory the amount of diphones present in a language is the square of the number of phones [8]. The disadvantage of this method is that there is no pronunciation variation in the diphones. This leads to unnatural sounding synthetic speech and information on segment duration and prosody must to be added to gain naturalness [10]. This includes information on stress levels and phone durations for the desired output.

The aim of this paper is to present a full blown Afrikaans TTS system that is flexible and natural. Therefore we implemented the concatenative synthesis technique based on diphones to build the Afrikaans synthesizer.

3. IMPLEMENTATION OF AFRIKAANS SYSTEM

Building the Afrikaans synthesizer using the Festival Speech Synthesis System is faced with the problems of automatic labeling and an undesired diphone index. The baseline system was constructed using the methods provided by Festival and was improved by manually labeling the diphones and rebuilding the diphone index.

3.1 The baseline system

The system is built using the Festival Speech Synthesis System which runs in a UNIX environment under Linux. The packages required are:

1. Festival-1.4.3
2. Festvox-2.0
3. Speech_tools-1.3

These packages are freely available for download from the CSTR website [3]. A diphone database and a lexicon database are required for building a new voice in Festival. Modules written in Scheme (a Festival specific language) are provided for these two requirements and are to be manipulated to suite the language.

Constructing the diphone database

The diphone database was constructed using *Die Groot Woorde Boek*, Afrikaans dictionary [9]. In total we found 64 phones therefore the amount of diphones were 4096. The diphone database was generated automatically by the system using the phone-to-phone transition rules for Afrikaans. These are the consonant-consonant, consonant-vowel, vowel-vowel and vowel-consonant transition rules for Afrikaans. The generated diphones are placed within non-sense words. These words are used for the extraction of the speech units for concatenation. Table 1 shows a list of diphones located within non-sense words.

Table 1: Examples of diphones located within non-sense words

Diphones	Non-sense word	Diphones with-in non-sense word
'b-a' 'a-b'	tababa	t *a-b-a-b* a
'sj-a' 'a-sj'	takasjata	t a k *a-sj-a* t a
'kn-o' 'o-kn'	takoknota	t a k *o-kn-o* t a
'tj-e' 'e-tj'	taketjeta	t a k *e-tj-e* t a

Recording the speaker

The objective of recording is to get the uniform set of diphone pronunciations. For this research my own voice was used. Recording was done using *na_record*, part of the *speech_tools-1.3* package. This recording system creates wave files of the recorded non-sense words and places them into a log file that stores them as * .wav files.

Labeling the non-sense words

The labeling of non-sense words is important because it labels the positions of the diphones within the non-sense words. At minimum the start of the preceding phone to the first phone in the diphone, the changeover and the end of the second phone should be labeled [8]. Festival provides an automatic labeler called *make_labs* to automatically label the diphones.

Building the diphone index

The diphone index is needed for the extraction of diphones from the acoustic non-sense words. During synthesis the system looks at this index to see where in the recorded non-sense words the diphone should be extracted from. The index is built by taking the diphone list and finding the occurrence of each diphone in a label [8]. By default the diphone will be extracted from the middle the first phone to the middle of the second phone. This is done by using *make_diph_index* a module provided by Festival.

Extracting the pitchmarks

Festival requires information on the pitch periods in an acoustic signal for synthesis and therefore the pitchmarks in each speech waveform must be extracted [8]. The technique used to get this information is called Residual Excited Linear-Predictive Coding (LPC). Linear prediction works on the basis that a current speech sample x(n) can be predicted from a finite number of previous *p* amount of samples x(n-1) to x(n-k) by a linear combination with an error e(n) [4]. This error term is the residual signal. And therefore

$$x(n) = e(n) + \sum^{p}_{k=1} a(k)x(n-k), \quad (1)$$

and

$$e(n) = x(n) - \sum^{p}_{k=1} a(k)x(n-k) = x(n) - \tilde{x}(n) \quad (2)$$

where $\tilde{x}(n)$ is the predicted value, p is the linear predictor order and a(k) are the linear prediction coefficients which are found by minimizing the sum of the squared errors over a speech frame [4].

The best way to find the pitchmarks in a waveform is to extract them from an EGG (electroglottograph) recording of the signal [8]. The EGG records the electrical activity in the glottis during speech, which means the pitch moments, can be found more easily and are more precise [8]. For this research no EGG was available so the pitchmarks were extracted automatically from the waveforms using methods provided by Festival.

Building the LPC parameters

Due to the natural changes in the recording environment and because of human fatigue the ideal recordings could not be realized. These factors made it impossible for all recorded diphones to be at the same power level. These fluctuations in power levels produce bad synthesis [8]. To overcome this power normalization was done on all the recorded non-sense words using a method provided by Festival. The method used finds the mean power for each vowel in each of the non-sense words and then finds the power factor with respect to the overall mean vowel power [8]. Using the calculated power factors the LPC coefficients and residuals for LPC analysis were generated.

Building lexicon support database and prosody

The lexicon database consists of the letter-to-sound rules and pronunciation guides for the system. Unpronounceable words and abbreviations are also given definition here.
Certain phones and diphones are not always as required when trying to pronounce certain words. Take the word "*Francois"* as an example. The first syllable of the word can be pronounced just by using the information of the phones. The second syllable is not pronounced correctly in the context of how the full word should be pronounced. For this reason the system needs to be told how to pronounce this syllable. Below is an example taken from the lexicon database that shows how the syllable is pronounces.

```
(lex.add.entry
 '('Francois' nil (((f r a n) 0) ((s w a) 0))))
```

This now gives the system a definition to how the word "Francois" should be pronounced.

Problem statement

Problems occur in the baseline system due to the mislabeling of diphones by the automatic labeler, and due to the basis on which the diphone index is built. The quality of a concatenative TTS system is directly related to the accuracy with which the underlying acoustic inventory is labeled [13]. Therefore because the diphones are mislabeled the performance of the system is undesired. The problem with the diphone index is that it is set up to only recognize the portion of the phone-to-phone transitions. This means that the entire transition is not used during synthesis which is also undesirable.

3.2 Improving the baseline system

By manually labeling the diphones and by changing the basis on which the diphone index is built the baseline system is improved.

The manual labeling the diphones fixes the errors made by Festival's automatic labeler by placing the labels in the correct positions. Figure 1 shows an example where the non-sense word "*a-c-i-c-a*" is labeled incorrectly.

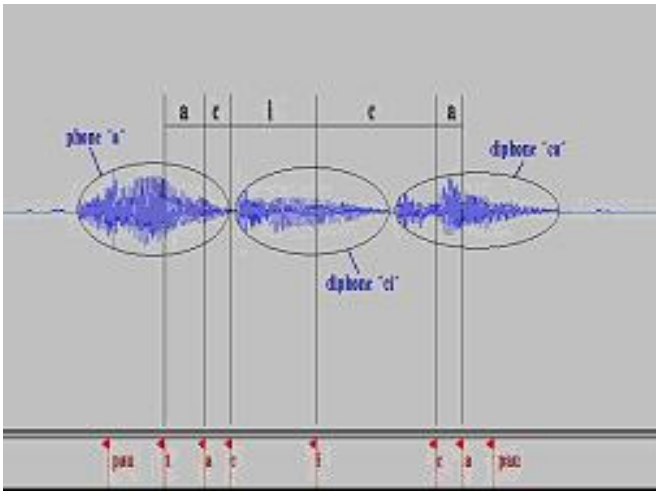


Figure 1: "*a-c-i-c-a*" labeled incorrectly

As seen from Figure 1, the diphone "*ci*" is labeled in such a way that it contains a portion of phone "*a*" and a portion of "*ci*". When the system calls "*ci"* for synthesis, it will pronounce the portion of "*a*" together with the portion of "ci", which is not desired. To solve this problem we re-label all the the non-sense words using Wavesurfer [7]. Figure 2 shows the correct labeling for the non-sense word "*a-c-i-c-a*"

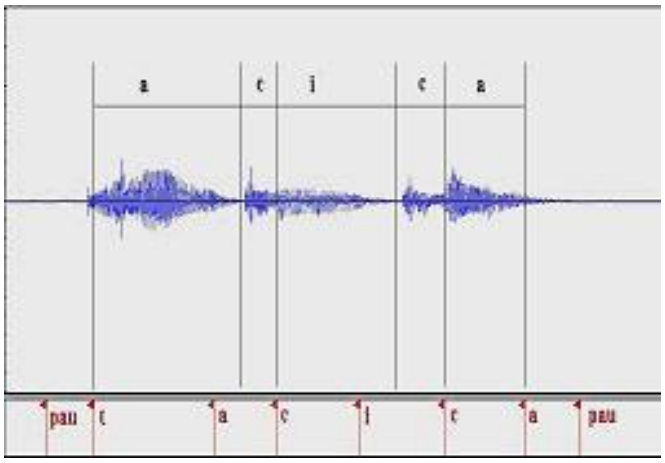
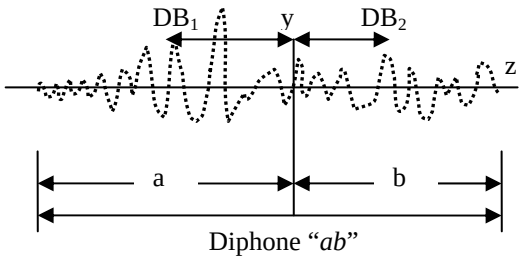


Figure 2: Correct labeling of *"a-c-i-c-a"*

Now the non-sense word is labeled such that it only contains the portion of "*ci*" that is needed and nothing else. When the system now calls on "*ci*" it will only pronounce what was labeled as "*ci*".

By default the diphone index is build in such a way that the portion from the middle of the first phone to the middle of the second phone is used for synthesis [8]. This is because diphone boundaries (DB) are positioned as shown in Figure 3. This is not desirable since full diphones are needed in order for the synthesized words that make sense.



Diphone "*ab*"

Figure 3: Diphone boundary of diphone "*ab*"

Festival's *make_diph_index* program uses the equation

$$DB_2 = (y+z)/2.0 \qquad (3)$$

where y- the mid point in diphone

z- the end point of second phone, to calculate the diphone boundary of the second phone.

To stretch the DB to the end of the diphone this line of code in *make_diph*_index was changed to:

$$\text{Let } DB_2 = z, \quad \text{instead of (3)}$$

This changes the definition the DB by placing it at the end of the diphone. The code is mirrored so the same applies for the first phone.

By applying these two methods to the baseline system, it will ensure that the full diphones are called at synthesis and not portions of it. This will increase the overall performance of the system.

4. EXPERIMENTAL RESULTS

The measures used for testing are based on how much of the diphones are recognized at synthesis. The system is tested by checking the periods of the diphones in Wavesurfer in comparison to the diphone index. For full diphone recognition these two periods should be the same. Table 2 shows the improvements made by the methods discussed in Section 3. It shows the true length of each diphone, the length recognized by the baseline and the improvements made in seconds.

Table 2: Percentage improvements on the baseline

Diphone	True wave-length (s)	Baseline wave-length (s)	Improve-ments made (s)	% Impr.
a-b	0.4432	0.3307	0.1125	34.01
b-a	0.3005	0.198	0.1025	51.76
a-c	0.5575	0.375	0.1825	48.66
c-a	0.411	0.281	0.13	46.26
a-d	0.482	0.392	0.09	22.95
d-a	0.337	0.207	0.13	62.80
a-f	0.4285	0.316	0.1125	35.60
f-a	0.3605	0.248	0.1125	45.36
a-g	0.638	0.54	0.098	18.14
g-a	0.498	0.352	0.146	41.47
a-h	0.5403	0.4303	0.11	25.56
h-a	0.395	0.285	0.11	38.59
a-j	0.6165	0.514	0.1025	19.94
j-a	0.4955	0.378	0.1175	31.08
a-k	0.414	0.294	0.12	40.81
k-a	0.362	0.212	0.15	70.75
a-l	0.5725	0.455	0.1175	25.82
l-a	0.455	0.335	0.12	35.82
a-m	0.5815	0.459	0.1225	26.68
m-a	0.432	0.317	0.115	36.27

As seen from Table 2 the improvements made are up to almost 50% in some cases. On average for these twenty diphones that were tested an improvent of 37.9% was made. Therefore more of the diphones are recognized during synthesis. This table also gives evidence to why the baseline system did not perform as desired. The majority of the portions recognized in the baseline were at the start of the diphone which means that the percentages lost at the end, held crucial information regarding the second phone in the diphone.

5. CONCLUSIONS

From the results shown is Section 4 it can be concluded that by manually labeling the diphones and changing the diphone index the overall quality of the TTS system will be improved.

Future work is to be done on completing the re-labeling process and changing the entire diphone index. This will ensure that all diphones are recognized correctly and hence should improve the overall quality of the system to such a point that it can synthesize full words and sentences accurately.

REFERENCES

[1] Prof J. Roux, Prof L. Botha, Prof J du Preez "African Speech Technology", Online Resource: www.ast.sun.ac.za, Last accessed 7 October 2004

[2] J. Oliver "Afrikaans", Online resource: www.geocities.co.za/users/~jako/lang/afrwrd.html, Last accessed 7 October 2004

[3] A. W. Black, R. Clark, K Richmond, S King "The Festival Speech Synthesis System" University of Edinburgh, Scotland www.csrt.ed.ac.uk/projects/festival, Last accessed 15 October 2004

[4] S. Lammetty, "Review of Speech Synthesis Technology", Master's Thesis, Department of Electrical Engineering, Helsinki University of Technology, March 1999, Available at http://www.acoustics.hut.fi/~slemment/dipp/index.html, Last accessed 5 August 2004

[5] A. Conkie, "Robust unit selection system for speech synthesis", Proc. Joint Meeting of ASA, EAA and DEGA, Berlin, Germany, March 1999.

[6] O. Salor, B. Pellom, M. Demirekler, "Implementation and Evaluation of a Text-To-Speech Synthesis System for Turkish", INTERSPEECH-2003/Eurospeech-2003. pp 1573-1576, Geneva, Switzerland, Sept. 2003

[7] K. Sjolander , J Beskow "Wavesurfer" , Audio Editing Software 2004, www.speech.kth.se/wavesurfer, Last accessed 25 September 2004

[8] A. W. Black, K. Lenzo "Building Synthetic Voices", unpublished document, Carnegie Mellon Universtiy, Available at http://festovx.org.bsv, Last accessed 5 November 2004

[9] Kritzenbeurg, M. S. B.(Matthys Stefanus Benjamin), Groot Woordeboek, Pretoria, Vanschaik 1972,

[10] N. Rochford, "Developing a new voice for Hiberno-English in The Festival Speech Synthesis System", Final Year Thesis Project, Trinity College Dublin. Available at http://www.cs.tcd.ie/courses/csll/projects4.html, Last accessed 7 June 2004

[11] T. Dutoit, "Introduction to Speech Synthesis Systems", Kluwer Academic Publishes, Dordrecht, 1997

[12] M. Chu, H. Peng, E. Chang, "A concatenative Mandarin TTS system without prosody model and prosody modification", Proceedings of 4th ISCA workshop on speech synthesis, Scotland, 2001.

[13] M.J Makashay, C.W Wightman, A.K Syrdal, A Conkie, "Perceptual Evaluation Of Automatic Segmentation In Text-To-Speech Synthesis", In Proc, ICSLP, volume 2, pp. 431-434, 2000

A default-and-refinement approach to pronunciation prediction

M. Davel and E. Barnard

Human Language Technologies Research Group
CSIR / University of Pretoria, Pretoria, 0001
mdavel@csir.co.za, ebarnard@up.ac.za

Abstract

We define a novel g-to-p prediction algorithm that utilises the concept of a 'default phoneme': a grapheme which is realised as a specific phoneme significantly more often than as any other phoneme. We find that this approach results in an algorithm that performs well across a range from very small to large data sets. We evaluate the algorithm on two benchmarked databases (*Fonilex* and *NETtalk*) and find highly competitive performance in asymptotic accuracy, initial learning speed, and model compactness.

1. Introduction and background

The ability to predict the pronunciation of a written word accurately is an important sub-component within many speech processing systems. This task is typically accomplished through explicit pronunciation dictionaries or grapheme-to-phoneme (g-to-p) rule sets. Both of these resources can be difficult to obtain and resource-intensive to develop when creating speech technology in a new language. The dictionary creation process can be made more efficient through the use of g-to-p rule-based bootstrapping [1]: an audio-enabled process whereby g-to-p rules are extracted from the current dictionary (however small) and used to predict additional entries. Predicted entries are subsequently presented to and verified by a human verifier and the process is repeated until a dictionary of sufficient size is obtained.

The efficiency of this process is influenced by the efficiency of the g-to-p rule extraction mechanism. G-to-p rules are typically used to generalise from existing pronunciation dictionaries when handling out-of-vocabulary words; and to compress information when requiring a pronunciation model in a memory-constrained environment. Such applications require a balance between the need for small rule sets, fast computation and optimal accuracy. During bootstrapping, a key requirement is learning speed, i.e. we are specifically interested in obtaining a high level of generalisation given a very small training set.

Various approaches to g-to-p rule extraction exist. When considering data-driven approaches, formalisms that have been used successfully for this task include neural networks [2], decision trees [3], pronunciation-by-analogy models [4]; various instance-based learning algorithms such as Dynamically Expanding Context (DEC) [5] and IB1-IG [6]; and the combination of methods and additional information sources through meta-classifiers [7].

The results when applying appropriate versions of the different formalisms mentioned above are typically comparable, with variations in performance for specific tasks. Languages with irregular spelling systems such as English and French perform well within analogy-based frameworks, while instance-based learning is well suited to languages with a more regular orthography, such as Italian or Dutch. Results for different algorithms are compared in greater detail in section 3.

In this paper, we describe a novel approach to the g-to-p rule extraction problem (section 2) and evaluate the new approach in comparison with benchmarked algorithms (section 3). We demonstrate the learning curve and asymptotic behaviour of the algorithm, and discuss the implications of our results in the concluding section.

2. Approach

Grapheme-to-phoneme prediction algorithms rely on the connection between the spoken and written form of a language. The more modern the language, the stronger this connection, the more regular the spelling system of the language, and the stronger the concept of a 'default phoneme': a grapheme that is realised as a single phoneme significantly more often than as any other phoneme. Figure 1 and Figure 2 illustrate this phenomenon for Flemish. When counting the number of times a specific grapheme is realised as a specific phoneme, most graphemes follow the trend depicted in Figure 1. For the 'most conflicted' phones (*h*,*j*,*n*,*u*), the trend is less strong, but also clearly discernable, as depicted in Figure 2. Similar trends are observable for languages with less regular spelling systems, with a larger proportion of graphemes of these languages displaying the behaviour depicted in Figure 2.

We use this information to define an algorithm that uses greedy search to find the most general rule at any given stage of the rule extraction process. When applying

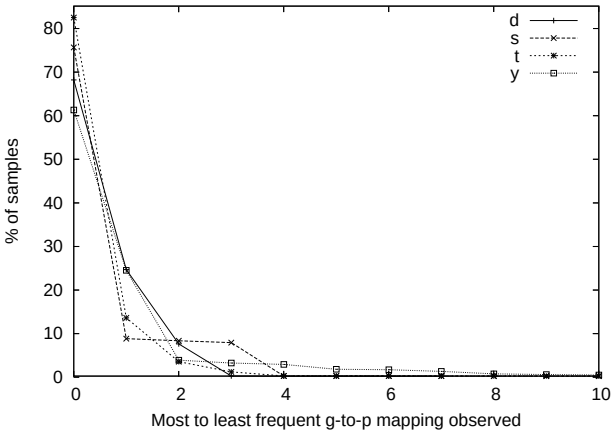


Figure 1: *Default phone behaviour of graphemes d,s,t and j in Flemish. Only the first 10 phonemic candidates are displayed.*

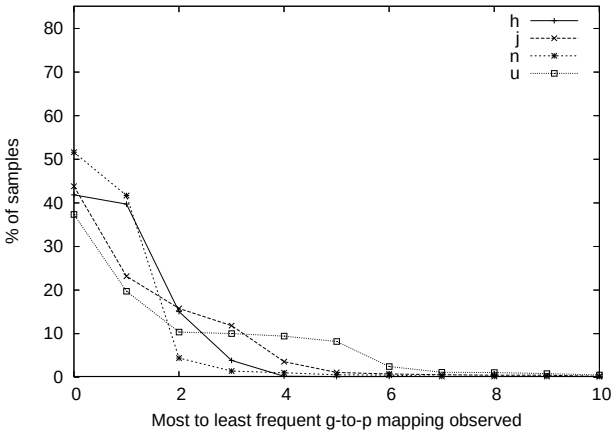


Figure 2: *Conflict phone behaviour of graphemes h,j,n,u in Flemish. Only the first 10 phonemic candidates are displayed.*

these rules during g-to-p prediction, we use the reverse rule extraction order. Explicitly ordering the rules provides flexibility during rule extraction, and ensures that the default pattern acts as a back-off for the next rule defined.

The framework we use is similar to that of most multi-level rewrite rule sets. Each g-to-p rule consists of a pattern:

$$(left\ context - g - right\ context) \rightarrow p \qquad (1)$$

The pronunciation for a word is generated one grapheme at a time. Each grapheme and its left and right context as found in the target word are compared with each rule in the ordered rule set; and the first matching rule is applied. Interestingly, while the rule application order of DEC (the algorithm closest to ours) is ordered by context size (largest rule first), our reverse rule extraction order automatically reverts to context size ordering in the case of DEC-based rule extraction.

Prior to rule extraction, grapheme-to-phoneme alignment is performed according to the Viterbi alignment process described in [8]. During rule extraction, the rule set for each grapheme is extracted separately. For any specific grapheme, applicable words are split into two sets based on whether the current rule set predicts the pronunciation of that grapheme accurately (*Completed* words) or not (*New* words). Definition of a new rule moves words from the *New* to the *Completed* set. Any words that are currently in the *Completed* set and conflict with the new rule, are moved back to the *New* set. The rule that will cause the most net words to be moved from the *New* to the *Completed* set is chosen first, irrespective of context size. Conflict is only resolved in the *Completed* set; new rules are allowed to conflict with words still in *New*. This ensures that the rule set is built for the default pattern(s) first.

Table 1: *The relationship between a word and its sub-pattern during rule extraction for grapheme e.*

Word	test
Word pattern	#t-e-st# → E
Sub-patterns	-e- → E,-e-s → E,t-e- → E,t-e-s → E t-e-st → E, #t-e-s → E,-e-st# → E #t-e-st → E,t-e-st# → E,#t-e-st# → E

In order to implement this algorithm in a computationally efficient way, the following techniques are used:

- The large word sets are used to keep track of status, but further manipulation utilises two sets of sub-patterns: the *Possible* sub-patterns, indicating all possible new rules, and consisting of all the sub-patterns of each word pattern in *New*, excluding all for which the left-hand side is an existing rule; and the *Caught* set of sub-patterns, indicating all the sub-patterns explicitly or implicitly covered by the current rule set. The relationship between a word and its sub-patterns is illustrated in Table 1. Hashes denote word boundaries.
- Words are pre-processed and the word patterns relevant to a single grapheme extracted and written to file. All further manipulation considers a single grapheme (and set of word patterns) at a time.
- The context size of the sub-patterns considered is grown systematically: only sub-patterns up to size $max + win$ are evaluated, where max indicates the current largest rule, and win is defined to ensure that any larger contexts that may be applicable are considered, without requiring all patterns to be searched.
- Both the *Possible* and *Caught* sets of sub-patterns count the number of times a matching word pattern is observed in the relevant word sets. The

next rule is chosen by finding the pattern for which the matching count in *Possible* minus the conflicting count in *Caught* is highest. (The conflicting count is the number of times a matching left-hand pattern is observed with a conflicting right-hand phoneme.)

- Whenever a sub-pattern in *Possible* or *Caught* reaches a count of zero, the sub-pattern is deleted and not considered further, unless re-added based on an inter-set move of a related word.

3. Evaluation

We use two corpora to evaluate the performance of the algorithm:

- *Fonilex*, a publicly available pronunciation dictionary of Dutch words as spoken in the Flemish part of Belgium. We use the exact pre-aligned 173,874-word dictionary as used in [7].

- *NETtalk*, a publicly available 20,008-word English pronunciation dictionary [9]. Hand-crafted grapheme-to-phoneme alignments are included in the dictionary.

In all experiments we perform 3-fold cross-validation based on a 90% training and 10% test set. We report on phoneme correctness[1], phoneme accuracy[2] and word accuracy[3]. Where there is uncertainty with regard to the measure used in the benchmark result, word accuracy provides the least ambiguous comparison.

3.1. Learning curve

As we aim to use this algorithm for the bootstrapping of pronunciation dictionaries, we are interested in the performance of the algorithm with very small training sets. We therefore evaluate word and phone accuracy for different training dictionaries of sizes smaller than 3,000 words, using subsets from *Fonilex*. Figures 3 and 4 demonstrates the learning curve for the algorithm *Default&Refine* in comparison with *DEC*. Each rule set is evaluated against the full 17,387-word test set.

The algorithm performs well, achieving 50% word accuracy ($\pm$ 90% phoneme accuracy) at 600 words. *DEC* requires an additional 1100 words before the same level of accuracy is reached. Since the correction of incorrectly predicted phonemes is the most labour-intensive aspect of bootstrapping pronunciation dictionaries, this represents a significant improvement to the process.

[1]Number of phonemes identifi ed correctly

[2]Number of correct phonemes - insertions, divided by the total number of phonemes in correct pronunciation

[3]Number of words completely correct

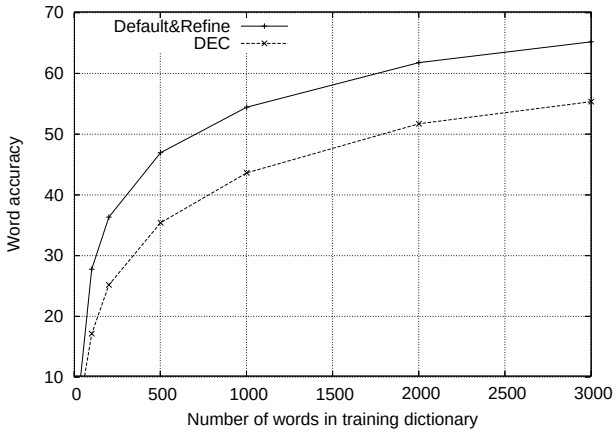


Figure 3: *Word accuracy during initial 3000 training words*

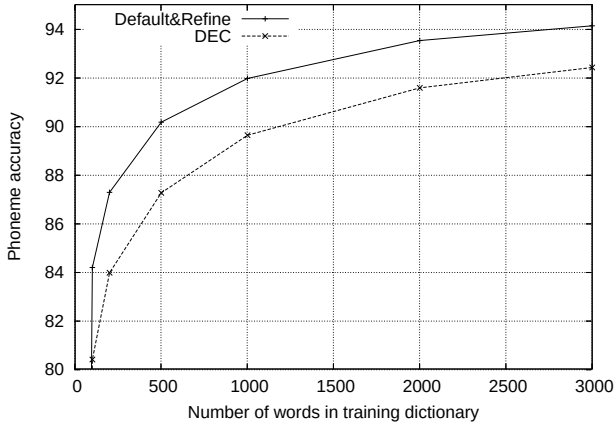


Figure 4: *Phoneme accuracy during initial 3000 training words*

3.2. Asymptotic performance

A steep initial learning curve does not imply that the algorithm will continue to perform well as the training data set increases. In order to evaluate asymptotic behaviour, we evaluate the accuracy of the algorithm when trained on the full *Fonilex* training set, and use the results reported by Hoste [7] as benchmark. Hoste compared various g-to-p approaches using the *Fonilex* corpus, including:

- Instance learning based IB1-IG as a single classifier.

- Cascading two separate IB1-IG classifiers (one trained on *Fonilex* and one on *Celex* - a Dutch variant corpus).

- Combining these classifiers using various meta classifiers including C5.0 (decision tree learning) IB1-IG, IGTREE (an optimised version of IB1-IG) and MACCENT (a maximum entropy-based algorithm).

- Using IB1-IG to create a meta-meta-classifier trained on the results of the previous meta-classifiers.

The different results obtained using a 156,487-word training subset of *Fonilex* are compared in Table 2. The *Default&Refine* single classifier performs better than the single classifier and the meta-classifier variations reported on; and achieves comparable accuracy to the meta-meta-classifier, without utilising the additional *Celex* data.

Table 2: *Accuracy comparison for different algorithms using the Fonilex corpus*

	Word accuracy	Phoneme accuracy	Phoneme correct
Single			
IB1-IG [7]	86.37	-	98.18
DEC-grow [8]	89.47	98.48	98.69
DEC-min [8]	90.44	98.53	98.75
Default&Refine	92.07	98.79	98.89
Meta		-	
MACCENT [7]	87.27	-	98.28
C5.0 [7]	88.41	-	98.48
IGTREE [7]	91.33	-	98.85
IB1-IG [7]	91.55	-	98.89
Meta-Meta			
IB1-IG [7]	92.25	-	98.99

With 3-fold cross-validation we observe a standard deviation in phone accuracy of 0.09 for *Default&Refine*.

3.3. Less regular spelling systems

We were interested whether the algorithm would fail for a language with a less regular spelling system, and evaluated the asymptotic performance of the algorithm on the *NETtalk* corpus, using results obtained by Anderson [3], Torkkola [5] and Yvon [4] as benchmarks. The algorithm performed surprisingly well, as shown in Table 3. The *SMPA* algorithm employs pronunciation by analogy, and is not suitable for training on small data sets.

Table 3: *Accuracy comparison for different algorithms using the NETtalk corpus*

	Word accuracy	Phoneme accuracy	Phoneme correct
Trie [3]	51.7	-	89.8
Decision Tree [3]	53.0	-	89.9
DEC [5]	-	-	90.8
DEC [4]	56.67	-	92.21
Default&Refine	58.45	90.39	91.31
SMPA [4]	63.96	-	93.19

With 3-fold cross-validation we observe a standard deviation in phone accuracy of 0.13 for *Default&Refine*. (In this table, the phoneme correctness reported in [4] for DEC seems anomalously high, in relation to our own experiments, those obtained in [5], and the reported word accuracy.)

3.4. Size of the rule set

While memory usage and the size of the rule set is typically not a concern during g-to-p bootstrapping, the size of the rule set does affect the speed of the g-to-p prediction algorithm. We therefore evaluate the growth in rule set size at different stages of the learning process.

In Table 4 we compare the number and size of rules for this algorithm with the rule set obtained via *DEC* and find that the rule set size is significantly smaller for *Default&Refine*. The latter algorithm provides both a more accurate and more compact prediction model.

Table 4: *Number and size of rules for DEC and Default&Refine when trained on training dictionaries of various sizes. The first column lists the number of graphemes in a rule, and subsequent columns give the number of rules of that size.*

	DEC			Default&Refine		
	1000	10000	156486	1000	10000	156486
1	26	26	26	26	26	26
2	94	107	132	151	197	227
3	539	1194	1429	314	775	1221
4	324	2032	3750	190	1394	4430
5	140	1882	8796	18	603	5293
6	33	740	7938	1	142	2510
7	4	276	7002	1	18	816
8	1	72	4218	-	6	288
9	-	25	2683	-	-	114
10	-	3	1574	-	-	74
11+	-	3	3165	-	-	71
	1161	6360	39270	701	3161	15070

4. Conclusion

The concept of a default phone suggests an interesting algorithm for g-to-p prediction, based on the extraction of a cascade of increasingly more specialized rules. This algorithm has a number of attractive properties, including rapid learning, good asymptotic accuracy, and the production of compact rule sets. We are integrating it into our bootstrapping system for dictionary creation [1], where it will be of value in our quest to develop linguistic resources for the languages of the developing world.

5. Acknowledgements

This work was supported by the CSIR *Information Society Technologies Centre*. We would like to thank Piet Mertens for providing us with access to the *Fonilex* database, and Veronique Hoste and Walter Daelemans for providing us with access to their experimental *Fonilex* data.

6. References

[1] M. Davel and E. Barnard, "Bootstrapping for language resource generation," in *Proceedings of the 14th Symposium of the Pattern Recognition Association of South Africa*, South Africa, 2003, pp. 97–100.

[2] T.J. Sejnowski and C.R. Rosenberg, "Parallel networks that learn to pronounce english text," *Complex systems*, vol. 1, pp. 145–168, 1987.

[3] O. Andersen, R. Kuhn, A. Lazarides, P. Dalsgaard, J. Haas, and E. Noth, "Comparison of two tree-structured approaches for grapheme-to-phoneme conversion.," in *Proceedings of the ICSLP*, Philadelphia, 1996, vol. 3, pp. 1700–1703.

[4] F. Yvon, "Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks," in *Proceedings of Conference on New Methods in Natural Language Processing (NeMLaP)*, Ankara, Turkey, 1996, pp. 218–228.

[5] K. Torkkola, "An efficient way to learn english grapheme-to-phoneme rules automatically," in *Proceedings of the International Conference on Acoustics and Speech Signal Processing (ICASSP)*, Minneapolis, 1993, vol. 2, pp. 199–202.

[6] Walter Daelemans, Antal van den Bosch, and Jakub Zavrel, "Forgetting exceptions is harmful in language learning," *Machine Learning*, vol. 34, no. 1-3, pp. 11–41, 1999.

[7] Erik Tjong Kim Sang Veronique Hoste, Walter Daelemans and Steven Gillis, "Meta-learning for phonemic annotation of corpora," in *Proceedings of the ICML-2000*, Stanford University, USA, 2000.

[8] M. Davel and E. Barnard, "The efficient creation of pronunciation dictionaries: Machine learning factors in bootstrapping," in *Proceedings of the ICSLP*, Jeju, Korea, 2004.

[9] T.J. Sejnowski and C.R. Rosenberg, "Parallel networks that learn to pronounce english text," *Complex Systems*, pp. 145–168, 1987.

Evaluating Microphone Arrays for a Speaker Identification Task

Nicholas Zulu, Daniel Mashao
Department of Electrical Engineering, University of Cape Town
Rondebosh, Cape Town, South Africa
pzulu@crg.ee.uct.ac.za daniel@eng.uct.ac.za

Abstract—Microphone array systems have been an area of active research for several years. The potential for high quality hands-free speech acquisition in noisy and reflecting environments makes microphone arrays an attractive alternative to conventional close-talking microphones. The signal-enhancement and source-location capabilities of microphone arrays make them applicable to a variety of tasks including teleconferencing, speaker tracking, speaker recognition and speech recognition. In this paper we evaluate techniques for setting up microphone arrays for speaker identification. We propose the use of an active noise canceling beamformer based on the generalized sidelobe canceller (GSC) beamformer. Significant improvements in identification rate are achieved using this method compared to other beamforming techniques investigated in this paper.

I. INTRODUCTION

Speaker identification systems are known to perform well when the speech signals are captured in a noise-free environment using a close-talking microphone worn near the mouth. However, many of the target applications of this technology do not take place in noise-free environments and it is often inconvenient for the user to wear a close-talking microphone. As the distance between the speaker and the microphone increases, the speech signal becomes increasingly susceptible to background noise and reverberation effects that significantly degrade speaker identification accuracy. This problem can be greatly alleviated by the use of multiple microphones to capture the speech signal.

Microphone arrays provide a means of localizing sound pickup and improving sound quality in noisy and reverberant conditions [1]. A microphone array uses multiple spatially distributed sensors to capture speech signals. The speech signals are captured simultaneously by each of the microphones and then processed jointly using one or more of a variety of methods to obtain a cleaner output signal [2]. The most important objective of a microphone array is to provide a high quality version of the desired speech signal for a specified application.

Microphone array speech enhancement techniques achieve this by beamforming, which reduces the level of localized and ambient noise signals, while minimizing distortion to speech from the desired direction. Beamforming has been applied to speaker identification as in [3], using speech signals generated by a computer model of room acoustics. This paper is aimed at contributing to research in the use of microphone arrays for speaker identification and proposes a beamforming technique based on the Generalized Sidelobe Canceller, (GSC) beamformer using real speech signals. This technique is aimed at reducing coherent and incoherent noise in speech signals acquired in an office environment, with minimal distortion to the desired speech.

Microphone array speaker identification has as one of its applications, automatic meeting transcription, where in conjunction with speech recognition, speakers in a conversation or conference can be identified. An example of such a deployment is being done at the Laboratory for Engineering Man/Machine Systems (LEMS) [4].

In exploring this topic, the principles of some basic beamforming techniques are discussed and evaluated. Thereafter, a review of current speaker recognition is given. A generalized sidelobe canceller is discussed and a slight modification to the GSC introduced. An overview of the system follows, with speaker identification results and conclusions.

II. BEAMFORMING TECHNIQUES

In this section three array processing techniques are reviewed. We present the theory behind these beamforming techniques, indicating their advantages, disadvantages and applicability to different noise conditions.

There are two classes of beamformers; data-independent (also known as fixed beamformers) or data-dependent (also known as adaptive beamformers). Data-independent beamformers are so named because their parameters are fixed during operation. Whereas, data-dependent beamformers continuously update their parameters based on the received signals.

A. Delay-and-sum Beamforming

The simple Delay-and-Sum beamformer is an example of a data independent beamformer [5]. The delay and sum beamforming algorithm adds the captured signals from the array sensors with corresponding delay in such a way that signal components originating from a desired location are combined coherently, while signals originating from other locations are combined in an incoherent fashion. This lends

the desired signal gain over undesired noise that increases as a function of the number of sensors [1]. By applying phase weights to the input channels, we can steer the main lobe of the directivity pattern to a desired direction. Phase shifts in the frequency domain can effectively be implemented by applying time delays to the sensor inputs. The delay for the n^{th} sensor is given by

$$\tau_n = \frac{(n-1)d\cos\phi'}{c} \qquad (1)$$

which is the time the plane wave takes to travel between the reference sensor and the n^{th} sensor. Where ϕ' is the direction of arrival of the wave, *c* is the speed of propagation and *d* is the inter-element spacing.

Delay-and-sum beamforming is so-named because the time domain sensor inputs are first delayed by τ_n seconds, and then summed to give a single array output. Expressing the array output as the sum of the weighted channels, we obtain in the time domain

$$y(t) = \frac{1}{N}\sum_{n=1}^{N} x_n(t-\tau_n) \qquad (2)$$

There exists a variation of delay-and-sum beamformers that combine the conventional delay-and-sum beamformer with channel filters to implement a desired shaping and steering of the beam pattern.

B. Filter-and-sum Beamforming

While the delay-and-sum beamformer is easy to understand, it offers minimal noise reduction and requires a large number of microphones to improve SNR [5]. It belongs to a more general class of beamformers known as *filter-and-sum beamformers*, where both the amplitude and phase weights are frequency dependent. In practice, most beamformers are a class of filter-and-sum beamformer.

The filter implemented in this research was a *multi-dimensional wiener filter*. The filter has as its inputs two correlation matrices: the correlation matrix of the *background noise* affecting the signal of interest and the correlation matrix of the *signal* affected by the noise. It is assumed that speech, ***s***, and affecting noise, ***n***, are statistically uncorrelated, and that noise is linearly added to speech: $\mathbf{x} = \mathbf{s} + \mathbf{n}$, where, for example, $\mathbf{x}$ is the output from the *N* channels of the microphone array for a given frame of analysis where each channel has a block of L_S samples being considered:

$$\mathbf{x} = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(L_S) \\ x_2(1) & x_2(2) & \cdots & x_2(L_S) \\ \vdots & \vdots & \ddots & \vdots \\ x_N(1) & x_N(2) & \cdots & x_N(L_S) \end{bmatrix} \qquad (3)$$

The objective is to estimate ***s*** given ***x*** and ***n*** for a defined filter order *L*. The algorithm has two correlation matrices as input, the background noise correlation matrix $\mathbf{R}_N$ and the signal correlation matrix $\mathbf{R}_X$. The optimal multi-dimensional wiener filter, $\mathbf{W}_{WF}$, is calculated as

$$\mathbf{W}_{WF} = \mathbf{R}_X^{-1}(\mathbf{R}_X - \mathbf{R}_N). \qquad (4)$$

As presented in [6], matrix $\mathbf{R}_X^{-1}$ above can be replaced by $(\mathbf{R}_X + \rho\mathbf{R}_N)^{-1}$, where $\rho \geq 0$. Increasing ρ improves the intelligibility at a cost of increasing signal distortion. The filtered signal matrix can then be computed from,

$$\mathbf{Y} = \mathbf{W}_{WF}.\mathbf{X}^T. \qquad (5)$$

The matrix $\mathbf{Y}$ comprises *N* filtered channel outputs which are separated and summed to give the beamformed output, y_W [7]. A block diagram showing the structure of a general filter-and-sum beamformer is given in Figure 1.

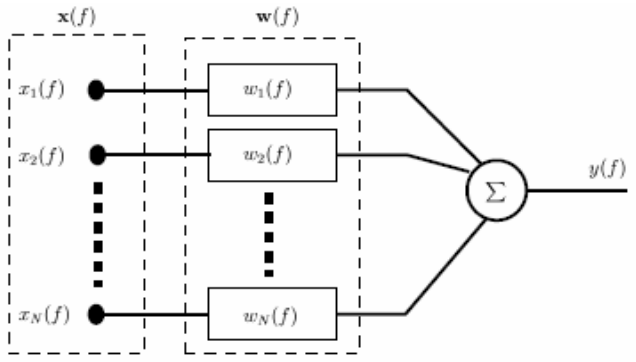


Figure 1: Filter-and-sum beamformer structure

C. Generalized Sidelobe Canceller (GSC)

A limitation of data independent beamforming techniques, such as the delay-and-sum and the filter-and-sum is their inability to adapt to changing noise conditions. Data-dependent beamforming techniques, such as the Generalized Sidelobe Canceller (GSC) [8] aim to solve this problem. The GSC separates the adaptive beamformer into two main processing paths. The first path implements a standard fixed beamformer with constraints on the desired signal. The second path is the adaptive part, which provides a set of filters that adaptively minimize the noise power in the output. The desired signal is blocked from the second path by a blocking matrix, ensuring that the noise power is minimized. Such an adaptive beamforming technique succeeds in significantly reducing the noise level for coherent noise signals emanating from localized sources [9]. Due to the blocking matrix, the lower path output only contains noise signals. The overall system output is calculated as the difference of the upper and lower path outputs

$$y(f) = y_u(f) - y_a(f) \qquad (6)$$

The GSC is a flexible structure due to the separation of the beamformer into a fixed and adaptive portion. In practice, the GSC can cause a degree of distortion to the desired signal due to what is termed signal leakage. This occurs

when the blocking matrix fails to remove all of the desired signal from the lower noise canceling path. The block structure of the generalized sidelobe canceller is shown in Figure 2.

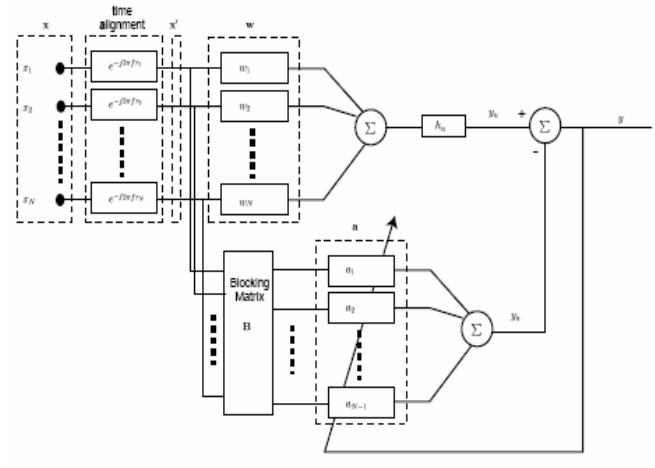


Figure 2: Generalized sidelobe canceller structure

In this section we have reviewed three common beamforming techniques. The delay-and-sum, filter-and-sum and the generalized sidelobe canceller. In the next section we discuss the speaker identification system we used to evaluate our microphone array.

III. SPEAKER IDENTIFICATION SYSTEM

Speaker recognition applications can be classified as either verification or identification tasks. Speaker verification tasks decide whether or not a speech segment was uttered by a specific speaker. On the other hand, speaker identification is concerned with recognizing an individual from a group of speakers based on a sample of his/her speech. The speaker identification system used in this research is text-independent. This type of speaker identification is concerned with determining who, from a group of known speakers, is speaking, regardless of what is being spoken. The speaker identification process can be summarized as follows: first the system needs to be trained with samples of speech collected from the speakers to be identified. Once this is complete, the system is tested (a speaker is identified) by comparing a speech sample from an unidentified speaker to the speech samples stored by the system and determining who the most likely speaker is [10].
Figure 3 illustrates a typical speaker identification system.

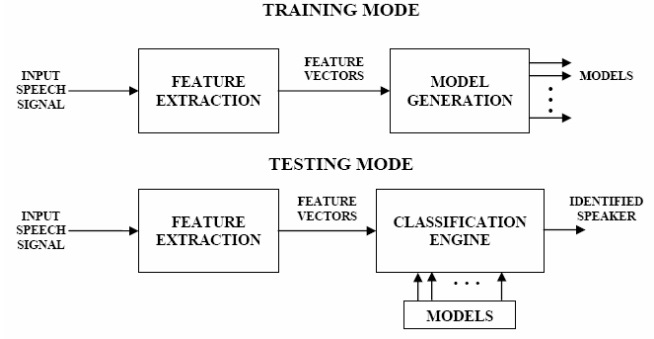


Figure 3: A typical speaker identification system

The system produces Mel-frequency Cepstral Coefficients (MFCC) in the feature extraction component. These features are aimed at emulating the spectral compression applied by the human auditory system to an incoming speech signal [10] and, are the most commonly used features used in speech-related research.
The system overview that follows describes the experimental configuration and results obtained from three beamforming techniques evaluated on a Gaussian Mixture Model (GMM) [11] based speaker identification system.

IV. SYSTEM OVERVIEW

A. Beamforming technique

In section II, three beamforming techniques outlining the important characteristics of each technique were discussed. The proposed beamforming technique for the speaker identification task is a variation of the generalized sidelobe canceller, comprising only the path with the blocking matrix.
The blocking matrix eliminates the desired signal from the lower path, allowing only the noise power to be minimized. As the desired signal is common to all the time-aligned channels, blocking will occur if the rows of the blocking matrix sum to zero. If $\mathbf{x}''$ denotes the signals at the output of the blocking matrix, then

$$\mathbf{x}''(f) = \mathbf{B}\mathbf{x}'(f) \tag{7}$$

where each row of the blocking matrix sums to zero, and the rows are linearly independent. As $\mathbf{x}'$ can have at most $N-1$ linearly independent components, the number of rows in $\mathbf{B}$ must be $N-1$ or less [9]. The standard Griffiths-Jim blocking matrix is [8]

$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \ddots & \cdots & \cdots \\ 0 & \cdots & 0 & 1 & -1 & 0 \\ 0 & \cdots & 0 & 0 & 1 & -1 \end{bmatrix} \tag{8}$$

Following application of the blocking matrix, $\mathbf{x}''$ is filtered and summed to give the lower path output y_B. If we denote the lower path filters as $\mathbf{a}$, then we have

$$y_B(f) = \mathbf{a}(f)^T \mathbf{x}''(f) \tag{9}$$

where y_B is a vector containing only noise samples. The positions of these samples are extracted in the noise canceling module (Figure 3), and the corresponding positions in the upper path output are replaced with nulls. Thus effectively canceling noise in the overall system output, y. Figure 4 illustrates the proposed beamforming

technique.

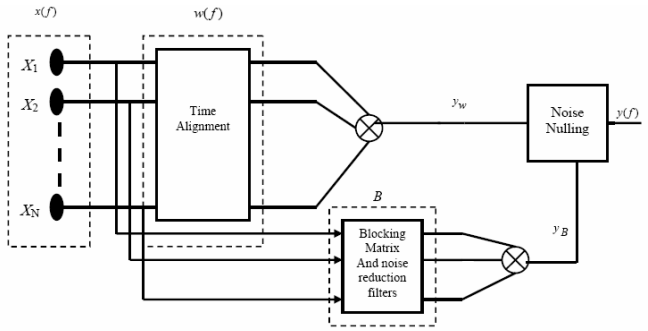


Figure 4: Active noise canceling beamforming structure

B. System description

The microphone array used in the evaluation is a 4 element (N) array placed on a table. The array is 9cm long with an equal inter-element spacing d, of 3cm giving it an effective length, L = N*d, of 12cm. It accommodates the frequency band; 2 kHz < f < 6 kHz. All signal sources are considered far-field to simplify calculations and Figure 5 shows the directivity pattern for a linear, equally spaced array of 4 microphones.

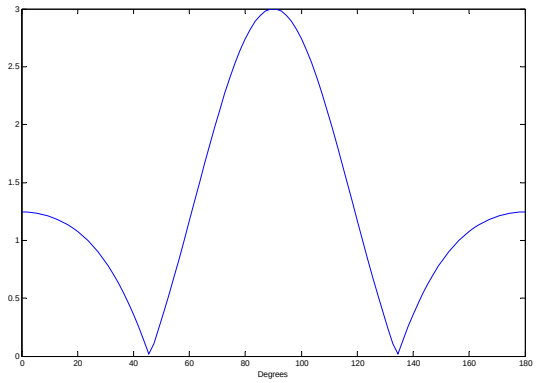


Figure 5: Directivity pattern for 4 element microphone array

The complete microphone array system comprises three main components; *the linear array, data acquisition module and processing module*. Figure 6 illustrates these three components and includes the speaker identification system.

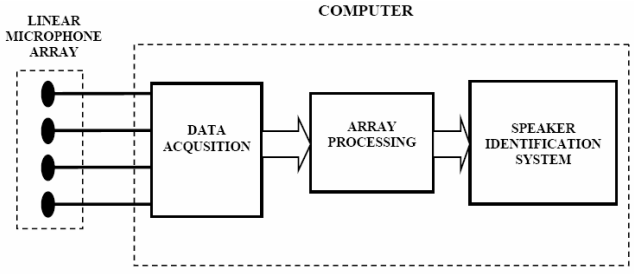


Figure 6: Microphone array system

The three components perform the following tasks:

1) *Linear Microphone Array*

The microphones act as transducers that convert sound pressure waves into electrical signals. Let us assume that a talker produces a speech message *x(t)* that is acquired by microphones 1, …., N as signals $x_1(n),\ldots, x_N(n)$. Signals sampled by microphones *i* and *k* are characterized by a relative time delay τ_{ik} of the direct wavefront arrival [12].

2) *Data Acquisition Module*

Signals from the microphone array are acquired for computer processing using a PCI703 series 16 analog input channel data acquisition board from Eagle Technology. The board has a maximum analog sample rate of 400 kHz with 14-bit accuracy. For 4 channels the sample rate used is 64 kHz (16 kHz per channel). After acquisition the data is converted to a suitable file format for processing.

3) *Array Processing Module*

Generally, array processing with regard to microphone arrays refers to beamforming. A beamformer performs spatial filtering. The beamforming capabilities of microphone array systems allow highly directional sound capture, providing superior signal-to-noise ratio (SNR) when compared to single microphone performance [1].

A total of 40 speech samples, comprising 20 training and 20 testing speech utterances, from 20 speakers were acquired using the microphone array. Each speaker was seated 50cm directly in front of the array. The speech was recorded in an office environment with interfering noise mainly from an air conditioner and other randomly distributed speakers. No additional noise was artificially introduced to the data.

C. Results

It has been shown that for clean speech recorded using a close-talking microphone, a GMM based speaker identification system similar to the one used in this research obtained a 100% identification rate [13]. It should be noted that the experimental setup and data used in [13] were different to that used in our evaluation. The baseline for the experiments to which further improvements will be compared, is the identification rate obtained using a single microphone under the same conditions as the microphone array. We obtained an identification rate of 60% for a 20 speaker database as a baseline. The performances of the delay-and-sum beamformer, filter-and-sum beamformer and the active noise canceling beamformer were evaluated and compared. All the systems compared fairly well to the baseline, with the active noise canceling beamformer attaining the highest improvement in identification rate of 85%. Table 1 displays the performance of the beamforming techniques on a 20 speaker database.

Beamforming Technique	Identification Rate
Single Mic. (Baseline)	60%
Filter-and-sum	65%
Delay-and-sum	70%

Noise Canceling	85%

Table 1: The effect of the beamforming techniques

It is clear from table 1 that all the beamforming techniques investigated improved the identification rate. These results are compared to the baseline, which is the identification rate achieved using a single microphone with speakers 50 cm from the microphone. The delay-and-sum beamformer outperformed the filter-and-sum beamformer due to signal distortions introduced by the multi-dimensional wiener filter used in these experiments [7]. The active noise cancellation technique produced the best results with a 25% increase in identification rate from the baseline.

Beamforming Technique	Identification Rate
Close-Talking Mic.	100%
Single Mic. (Baseline)	60%
Noise Canceling	**85%**

Table 2: Baseline compared to Active Noise Cancellation

We suspect the active noise cancellation beamformer performs better because of the small population used for these experiments and the cleaner signal that it produces.

V. CONCLUSIONS

The work presented here has demonstrated that using a microphone array for speech acquisition offers a performance advantage for a speaker identification application in a distant-talking environment. We reviewed an active noise canceling beamformer, a delay-and-sum beamformer and a filter-and-sum beamformer, and found that the active noise canceling beamformer proved superior when evaluated on a speaker identification task.

We aim to further the research in the field by addressing the following:

1. Investigating the use of more sophisticated beamforming techniques used with speaker tracking.
2. More experiments into the effect of microphone arrays on speaker identification performance with respect to distance.
3. Increasing the speaker database.

REFERENCES

[1] D.V. Rabinkin, R.J. Renomeron, J.C. French and J.L. Flanagan, "Optimum microphone placement for array sound capture", *Proc. SPIE, Vol. 3162, pp. 227-239, 1997.*

[2] M.L. Seltzer, B. Raj and R.M. Stern, "Speech recognizer-based microphone array processing for robust hands-free speech recognition", *Proc. IEEE Conf. on Acoustics, Speech and Sig. Proc., May, 2002*, Orlando, Florida.

[3] Q.Lin, E. Jan and J. Flanagan, "Microphone arrays and speaker identification", *IEEE Transactions on Speech and Audio Processing, Vol. 2, pp. 622-629, October 1994*

[4] LEMS Microphone-Array Papers [Online] : http://www.lems.brown.edu/ *Accessed: October,11th 2004.*

[5] V.C. Raykar, "A study of various beamforming techniques and implementation of the constrained least mean squares (LMS) algorithm for beamforming", *Course project report ENEE 624, Fall 2001*.

[6] D.A. Florêncio and H.S. Malavar, "Multichannel filtering for optimum noise reduction in microphone arrays", *ICASSP 2001, Salt Lake City, May 2001.*

[7] I. Sanches, A. Girardi, "Multi-dimensional Filtering for Speech Enhancement via Microphone Array", *2nd International Symposium of NAIST-IS 21st century COE program, Japan October 2003.*

[8] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming", *IEEE Trans. on Antennas and Propagation, Vol. 30(1), pp. 27-34, January 1982.*

[9] I.A. McCowan, "Robust speech recognition using microphone arrays", *PhD Thesis, Qeensland University of Technology, Australia, 2001*.

[10] H. Gish and M. Schmit, "Text-Independent Speaker Identification", *IEEE Signal Processing* Magazine, *pp. 18-32, Oct.1994.*

[11] D.A. Reynolds and R.C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE transactions on Speech and Audio Processing, Vol. 3, No. 1, January 1995.*

[12] M. Omologo, M. Matassoni, P. Svaizer and D. Giuliani, "Microphone array based speech recognition with different talker-array positions", *Proc. ICASSP '98*, Seattle Washington

[13] D.A. Reynolds, "Large Population Speaker Identification Using Clean and Telephone Speech", *IEEE Signal Processing Letters, Vol.2, No.3, March 1995.*

N. Zulu is currently pursuing an MSc in Electrical Engineering at the University of Cape Town and is in his first year of study.

Dr. D. Mashao is a senior lecturer at the University of Cape Town and head of the Speech research and Technology Group. He is also the supervisor of the above-mentioned author.

3D Flame Reconstruction Techniques towards the Study of Fire-Induced High Voltage Discharge Phenomena

C. G. Crompton
School of Electrical, Electronic and Computer Engineering,
University of KwaZulu-Natal
crompton@ukzn.ac.za

D. A. Hoch
School of Electrical, Electronic and Computer Engineering,
University of KwaZulu-Natal
hoch@ukzn.ac.za

Abstract

This paper considers the problem of creating three dimensional reconstructions of fire from images. By showing how the image intensity of fire can be related to the flame density, the problem is able to be viewed from a tomographic perspective. A new method is presented, based on algebraic tomography techniques and fuzzy image processing, that produces geometrically accurate 3D reconstructions of a synthetic object given only a few views.

1 Introduction

The burning of sugar cane under high voltage transmission lines is known to cause phase to phase and phase to earth flashovers [1]. While the burning can be controlled to an extent, this is not always the case, and, along with wild fires, the problem of fire-induced flashovers remains a concern. Also, while several theories exist as to the mechanism of the flashover process, there is no definitive explanation. To find a solution, the problem must first be understood. Development of a 3D reconstruction imaging system will enable more accurate study of the fire-induced flashover phenomenon.

1.1 Object Reconstruction

The simplest method of creating a volumetric reconstruction of an object is to use the concept of the visual hull. Initially the term visual hull was defined by Laurentini [2] as the largest approximation of an object consistent with all possible silhouettes. A more common definition is rather the volume formed by a finite number of silhouettes of an object. The intersection of the silhouette projections forms the visual hull, which is guaranteed to contain the object. Although more silhouettes yield a better reconstruction, one is still limited by the fact that surface concavities cannot be modelled.

Much work has been done on improving the geometric accuracy of visual reconstructions. By using information such as colour and texture, better reconstructions may be obtained through various space carving techniques. Another potential approach is the use of stereo reconstruction techniques. The problem with most of these more advanced techniques is a reliance on lambertian surfaces, where light is reflected equally in all directions from a surface. Fire, however, is non-lambertian, being partially transparent. Hence, in terms of general object reconstruction techniques may be limited to a visual hull approach. This implies that information is being disregarded since only the silhouette is being used. The focus of this paper is thus on developing a technique able to use all the information available in the images.

1.2 Flame Reconstruction

The solution to the flame reconstruction problem may be approached from two different directions. The first approach is to use a geometric method, as demonstrated by Yan *et al* [3, 4]. Using only three cameras, they extract the contours of the flame which are joined using β-spline curves to form a mesh. While similar to the visual hull, using the idea of shape from silhouette, the resulting reconstruction appears more natural and curved, suiting the fluid nature of fire. Although this method is fast and demonstrates good results for simple flames, more complex flame geometries are likely to produce limited success.

Ng and Zhang [5] present a stereoscopic method for reconstructing a flame surface. While successful, the technique would appear to be limited to turbulent impinging diffusion flames. Also, the reconstructed surface is only visible from one direction – not a full 3D reconstruction.

The second category of flame reconstruction techniques is those using tomography. In §2 it is shown that the image intensity can be related to the flame density, making the reconstruction problem analogous to that of computerised tomography. Often used in medical applications (X-ray CT, MRI), traditional reconstruction algorithms, which typically require many views, cannot be used here since only a few views are available. Algebraic tomographic techniques, however, are well suited to sparse view reconstructions [6]. This will form the basis of the proposed reconstruction algorithm.

Hasinoff [7, 8] demonstrates two novel methods of reconstructing the flame density field using a tomographic approach. The aim of his research was the creation of photo-consistent reconstructions, and thus his methods are not particularly applicable in terms of recovering geometric characteristics of the flame. His work does, however, show the viability of the tomographic approach.

1.3 Overview

A method of flame reconstruction using tomographic and image processing techniques is proposed. Using several monochromatic CCD cameras positioned around the flame, video sequences are captured. The reconstruction is done one frame at a time. A black background simplifies the process, although an arbitrary background can be accounted for.

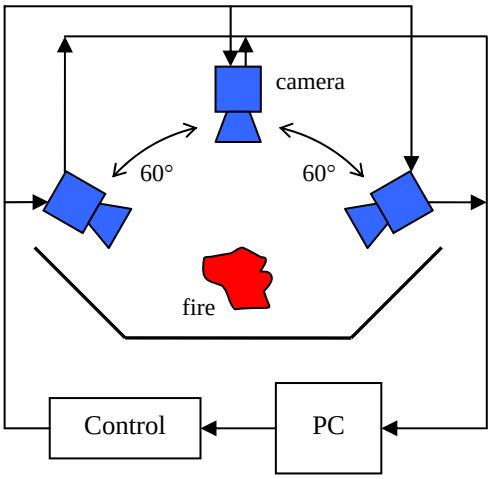


Figure 1.1: System setup. Given the semi-transparent nature of the flame the cameras need only be spaced equally within 180°. For simplicity only three cameras are shown, though more are likely be used.

To reconstruct a 3D volumetric reconstruction of the fire the reconstruction space is split up into a set of horizontal slices, each corresponding to a row of pixels in the images. By reconstructing each slice and then stacking them to form a 3D volume the reconstruction problem is reduced from 3D to 2D.

For each slice, an estimation of the density field is obtained using the Simultaneous Iterative Reconstruction Technique (SIRT), an algebraic tomography algorithm. However, since the flame geometry is of concern there is a need to incorporate a form of image segmentation to define the flame boundary. Fuzzy C-Means (FCM) segmentation is a technique often used for medical applications such as the segmentation of MRI brain scans [9, 10]. These scans are also created using tomographic techniques. This similarity would suggest that FCM segmentation could be a useful tool for segmenting the flame density field images. Image gradient and line detection type techniques would not work here, since the images created by the SIRT technique tend to be blurry and indistinct, lending themselves to a more statistical method such as Fuzzy C-Means. The theory behind SIRT and FCM is given is §3.

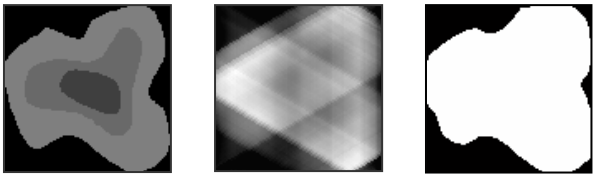


Figure 1.2: (a) Original 2D object, (b) SIRT reconstruction of object from 3 views, (c) FCM segmentation of (b).

2 Image Formation Model

In order to justify the use of tomographic methods for the purpose of flame reconstruction, it must be shown that for a non-saturated image there is a relationship between pixel intensity and flame density [7, 8].

This is achieved using the simplified radiative transfer model of fire. The two simplifications, or assumptions, that need to be made are:

i. **Negligible scattering:** For relatively smokeless fires this is a good approximation since the radiance is dominated by the self-emission from glowing soot particles [6, 7, 8]. The total emission thus consists only of self-emission.

ii. **Constant self-emission:** By modelling the brightness of the fire as being dependent only on the density field [7, 8], one may assume the self-emission to be constant, per unit mass, denoted by Q_0.

These two simplifications allow the expression of the intensity of a given ray as:

$$I = (1-\tau)Q_0 \,. \tag{2.1}$$

The total transparency, τ, along a given ray is given by the radiative transfer model as the integral of the density field, $\rho(x)$, along the ray:

$$\tau = \exp\left(-\sigma_t \int_l \rho(x).dx\right), \tag{2.2}$$

where σ_t is a medium dependent constant relating density and transparency known as the extinction cross-section.

To account for the emission constant, Q_0, it is noted how Equation (2.1) shows that, as the transparency tends to zero, the intensity approaches Q_0, thus giving a maximum intensity of $I_\infty = Q_0$. Given a digital imaging system with a saturation of I_{max}, we can then say that $I_\infty = I_{max} + 1$. Thus, eliminating Q_0 gives:

$$I = (1-\tau)I_\infty \,. \tag{2.3}$$

Rearranging this equation in terms of τ:

$$\tau = 1 - \frac{I}{I_\infty}, \tag{2.4}$$

and finally, combing Equations (2.2) and (2.4) and manipulating gives a transformed intensity:

$$\begin{aligned} I' &= \sigma_t \int_l \rho(x).dx \\ &= -\ln\left(1 - \frac{I}{I_\infty}\right). \end{aligned} \tag{2.5}$$

This transformed intensity, I', corresponds to the integral of the density field along the ray, thus showing that computerised tomography solutions can be applied to the problem of flame reconstruction.

It should be mentioned that Equation (2.5) assumes a black background. To account for an arbitrary background, Equation (2.3) is modified to include a background intensity term:

$$I = \tau I_{bg} + (1-\tau) I_\infty . \tag{2.6}$$

The transformed image intensity then becomes:

$$I' = -\ln\left(\frac{I - I_\infty}{I_{bg} - I_\infty}\right). \tag{2.7}$$

3 Tomographic Object Reconstruction

Having established that the problem of flame reconstruction can be approached from a tomographic perspective, the fundamental techniques needed to achieve this reconstruction are described.

3.1 SIRT

The Simultaneous Iterative Reconstruction Technique (SIRT) is an enhancement of the Algebraic Reconstruction Technique (ART), which should therefore be described first.

Consider an n x n square grid containing an unknown 2D object, with f_i representing the value of the ith grid element, or cell. Each image (I_k), or projection, of the object, is composed of a series of rays of width δ. The width of the ray is usually similar to the width of the grid element [11].

For a particular ray, j, the ray-sum, p_j, represents the object density along that ray, and can be expressed in terms of the grid as:

$$p_j = \sum_{i=1}^{N} w_{ij} f_i , \tag{3.1}$$

where the weight factor w_{ij} is the fraction of grid element i through which ray j passes, indicated by the shaded portion of cell A in Figure 3.1. To simplify matters w_{ij} is often replaced by a binary function, assigning a value of 1 if the centre of the cell falls within the ray [11].

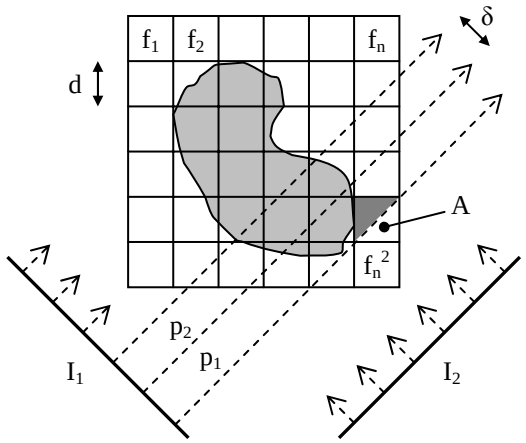


Figure 3.1: A square grid superimposed on an unknown object, with two projections (I_1, I_2) shown.

The central idea behind algebraic techniques is to iteratively adjust the cell values based on the difference between the original projections, p, and the projections created by the reconstruction, q. This adjustment, for a cell value f_i falling within ray j, can be written as:

$$\Delta f_{ij} = \frac{p_j - q_j}{N_j}, \tag{3.2}$$

where N_j is the number of image cells within the ray. This iterative adjustment of f_i is calculated for each ray in each projection and the cells within that ray are updated immediately. This is known as ART. Improved reconstructions can be obtained using the SIRT algorithm, where f_i is updated by first calculating Δf_{ij} for all the projections and using the average as Δf_i:

Further improvements can be obtained by replacing Equation (3.2) with:

$$\Delta f_{ij} = \alpha\left(\frac{p_j}{L_j} - \frac{q_j}{N_j}\right), \tag{3.3}$$

where L is the length of the ray and α is a relaxation parameter allowing control of the strength of the adjustments.

Convergence can be determined using criteria such as the difference between measured and calculated projections [12], or the total change made to the reconstruction for the iteration ($\sum \Delta f_i$).

3.2 Fuzzy C-Means Segmentation

Fuzzy C-Means (FCM) image segmentation is a pixel classification technique based on statistical pixel features, usually mean and standard deviation [13]. This is achieved in an iterative manner, assigning membership values to each pixel according to their distance from each image class centroid.

Consider an image with n pixels or cells, to be segmented into c classes. Let $\{x_1, x_2, \ldots, x_n\}$ form set X, where each element x_k is a vector containing the pixel features of pixel k. Each class, or fuzzy subset, of X, is defined by a centroid v_i, giving the central features of the class. X is partitioned into these subsets by assigning a fuzzy membership value u_{ik} to each pixel k, indicating the similarity to each class i, where the following conditions must be satisfied [14]:

$$0 \le u_{ik} \le 1 \quad \forall i,k\,, \tag{3.4}$$

$$\sum_{i=1}^{c} u_{ik} = 1 \quad \forall k\,, \tag{3.5}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \quad \forall i\,. \tag{3.6}$$

The FCM algorithm is an iterative process, continually updating U and V until Δu_{ik} is suitably small. This is done with the following equations:

$$v_i = \frac{\sum_{k=1}^{n} {u_{ik}}^m x_k}{\sum_{k=1}^{n} {u_{ik}}^m} \quad \forall i\,, \tag{3.7}$$

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{(m-1)}} \right)^{-1} \quad \forall i,k\,, \tag{3.8}$$

where $d_{ik} = \|x_k - v_i\|$ represents the similarity between pixel k and centroid i [13]. The parameter m is a real number, affecting the fuzziness of the process.

Once complete, the membership values are used to determine which class each pixel belongs to – defuzzification.

Some advantages of FCM are [14]:

- unsupervised operation,
- ability to accommodate any number of classes and any number of features,
- normalised distribution of membership values.

Mohamed *et al* [14] demonstrated a modification to the standard FCM process that improves performance for noisy images. They achieved this by altering the calculation of d_{ik}, the similarity measure, to take into account the fuzzy membership values of neighbouring pixels:

$$d_{ik}(new) = d_{ik}\left(1 - \alpha \frac{\sum_{j \in neighbours} u_{ij} \times p_{kj}}{\sum_j p_{kj}}\right), \tag{3.9}$$

where p_{kj} is a distance measure between pixel k and its neighbouring pixels j, and α is a constant, satisfying $0 \le \alpha \le 1$, that determines the strength of this neighbour effect.

4 The Fuzzy Hull Reconstruction Algorithm

The initial idea was to reconstruct each slice using SIRT, followed by FCM segmentation, which shall be referred to as the SIRT+FCM method. While the initial results were reasonable, they could be improved upon, since not all the information available was actually being used. When the reconstructed object is viewed from the same direction as one of the original camera images, it should project the same silhouette as the camera image. Obviously, the process of only using SIRT and FCM sequentially did not consider this at all. The concept of the visual hull needed to be incorporated into the reconstruction method.

The proposed algorithm, while still using the principles of SIRT and FCM, attempts to maintain silhouette consistency by iteratively comparing the reconstructed slice and the visual hull and modifying the reconstruction based on this comparison. This method shall be referred to as the Fuzzy Hull method. Figure 4.1 illustrates the process. Furthermore, by restricting processing to within the visual hull, instead of the entire slice, processing time is reduced.

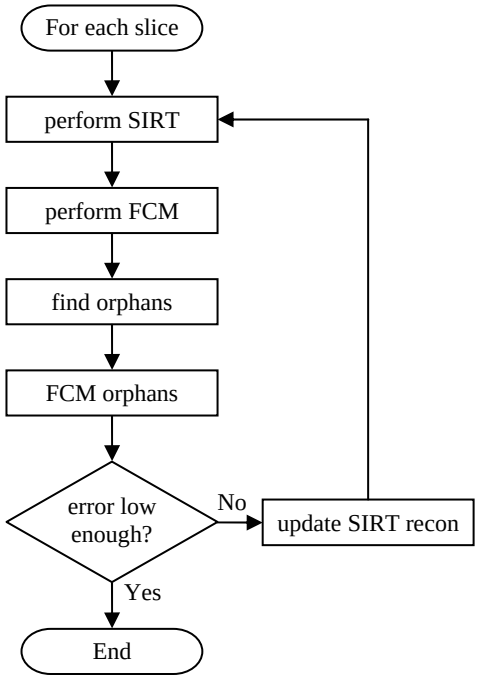


Figure 4.1: Flow diagram of the Fuzzy Hull slice reconstruction algorithm.

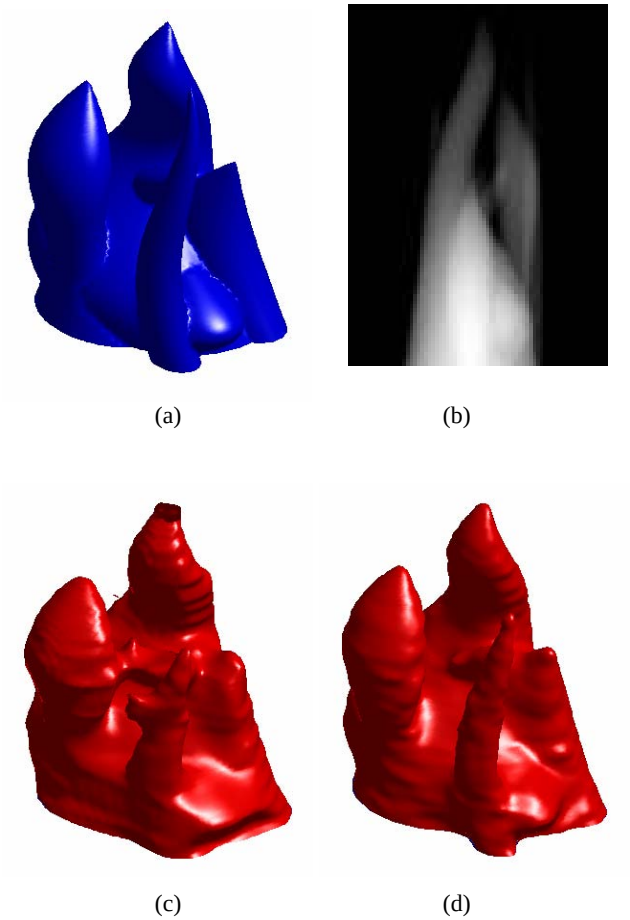
The term *orphans* is used to describe rays which are part of the original flame silhouette, but are not in the reconstruction silhouette. The pixels in the reconstruction slice projecting to these rays are then subject to a further Fuzzy C-Means segmentation, in an effort to restore silhouette consistency using the most likely pixels. The slice reconstruction is then updated using the FCM and

orphan FCM information and put through the entire process again. Once the number of *orphans* is sufficiently low, or a certain number of iterations have been reached, the process ends.

5 Results

The work presented here is still being developed, and while the Fuzzy Hull algorithm has only been tested using synthetic data, the results are promising.

The synthetic test object was created in Matlab™ by combining several semi-randomly distorted cones, giving a flame-like structure sized 100x100x100. By summing the density when viewing the object from a particular direction, an intensity image was created representing a flame image captured from a camera. These projected images are used as the input data for the reconstruction algorithms. Both the SIRT+FCM algorithm and the Fuzzy Hull algorithms were tested, so that both techniques may be compared.



(a) (b)

(c) (d)

Figure 5.1: The original object (a) is reconstructed from 5 views using: (c) the SIRT+FCM algorithm, (d) the Fuzzy Hull algorithm. (b) shows an example of the intensity images used to simulate the flame images.

Figure 5.1 shows that the Fuzzy Hull algorithm gives a visually superior reconstruction to the sequential SIRT+FCM method. For a more quantitative analysis several criteria are defined to judge the reconstruction:

i. **Total error:** The total reconstruction error, defined as:

$$E_{tot} = \frac{(f_p + f_n)}{total\ object\ voxels} \times 100\ \% \qquad (5.1)$$

where f_p and f_n are the false positive and false negative voxel identifications. A voxel, or volume pixel, is the 3D equivalent of a pixel.

ii. **Volume Error:** Percentage difference between original and reconstructed volumes.

iii. **Surface Area Error:** Percentage difference between original and reconstructed surface areas.

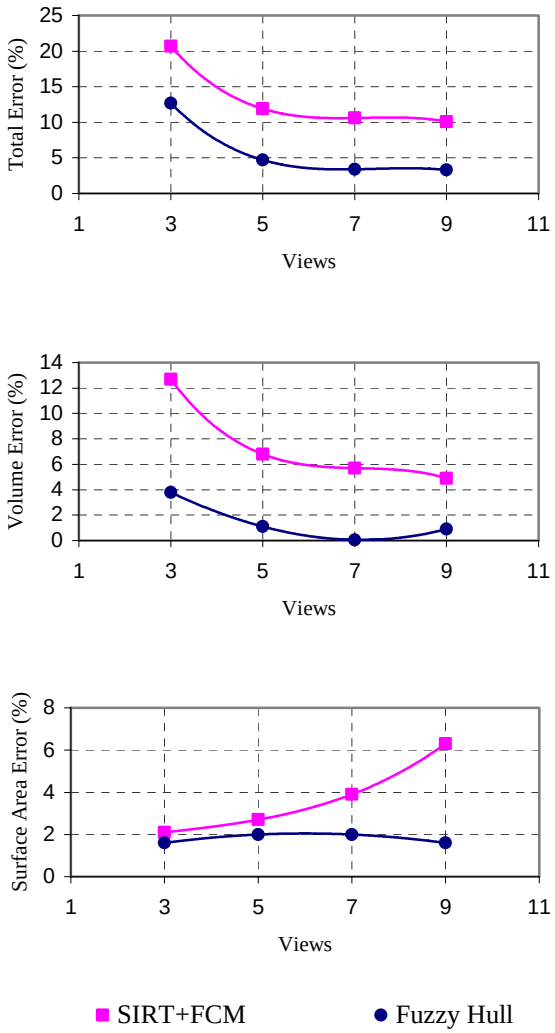


Figure 5.2: Graphs of the error estimates from the reconstructions.

Figure 5.2 shows the Fuzzy Hull algorithm to be a significant improvement on the SIRT+FCM algorithm. The error estimates are shown for different numbers of views used in the reconstruction, since the number of cameras to be used in the real setup has not yet been finalised, although five are likely to be used. As one would expect there is a general reduction of error with an increase in number of views. The surface area error would appear to be increasing, but this is most likely a result of positive errors balancing negative errors.

It is acknowledged that these results are obtained using an artificial object, and that reconstructions with real flame images may not be as good. However, it is important to ensure that the reconstruction algorithm works well on ideal data, before using real data and this has been done.

Several issues need to be addressed in order to implement the algorithm into a real system.

- **Camera calibration:** A simple calibration system needs to be developed to determine the angle between cameras and the size of the flame.
- **Contour extraction:** In order to implement the Fuzzy Hull algorithm, which uses the visual hull concept, the flame silhouette must be extracted from the images. A pixel intensity ratio method, as used by Yan *et al* [3], will be attempted.
- **Spatial Coherence:** The Fuzzy Hull algorithm does not consider the effect of neighbouring slices. This will be addressed if necessitated by results from real data.
- **Iso-Surface Creation:** Currently the volumetric reconstruction data is transformed into a polygonal model by Matlab™. The Marching Cubes algorithm, or a variation known as Adaptive Skeleton Climbing [15], will need to be implemented. Alternatively the reconstruction may be left in voxel format.
- **Flame analysis:** The flame reconstruction must be more comprehensively analysed considering the application of high voltage flashover research.

6 Conclusion

The integration of algebraic tomography and fuzzy image processing into an algorithm capable of solving sparse view tomography problems has been demonstrated. Incorporating a silhouette consistency constraint gave improved results by allowing the use of all the information available from the images. Using a synthetic flame-like object the Fuzzy Hull algorithm produced geometrically accurate reconstructions with a relatively low total error. While the algorithm has yet to be tested with real fire images, reasonable success is expected, given the promising results achieved so far.

References

[1] A. Sukhnandan and D. A. Hoch, "Fire Induced Flashover of Transmission Lines: Theoretical Models," *Africon 2002*, George, South Africa, October 2002, pp 617-622.

[2] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, February 1994.

[3] H.C. Bheemul, G. Lu and Y. Yan, "Three-dimensional visualization and quantitative characterization of gaseous flames," *Measurement Science and Technology*, vol.13, no.10, pp.1643-1650, October 2002.

[4] H.C. Bheemul, G. Lu and Y. Yan, "Digital Imaging Based Three-Dimensional Characterization of Combustion Flames," *Proceedings of IEEE IMTC (Instrumentation & Measurement Technology Conference)* 2003, Vol.1, pp. 420-424, Vail, USA 2003 20-22 May.

[5] W.B. Ng and Y. Zhang, "Stereoscopic imaging and reconstruction of the 3D geometry of flame surfaces," *Experiments in Fluids*, Volume 34, Issue 4, pp. 484-493, 2003.

[6] D.P. Correia, P. Ferrão and A. Caldeira-Pires, "Advanced 3D Emission Tomography Flame Temperature Sensor," *Combustion Science and Technology*, vol. 163, pp. 1–24, 2001.

[7] S.W. Hasinoff and K.N. Kutulakos, "Photo-Consistent 3D Fire by Flame-Sheet Decomposition," *In Proceedings of 9th IEEE International Conference on Computer Vision*, ICCV 2003, pp. 1184-1191.

[8] S. Hasinoff, "*Three-dimensional reconstruction of fire from images*," Master's thesis, Univ. of Toronto, Dept. of Computer Science, 2002.

[9] M. N. Ahmed, S.M. Yamany, N. Mohamed, A.A. Farag, and T. Moriarty, "A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data," *IEEE Trans Med Imaging.* 2002 Mar;21(3):193-9.

[10] D.L. Pham and J.L. Prince, "Adaptive Fuzzy Segmentation of Magnetic Resonance Images,"' *IEEE Transactions on Medical Imaging*, 18(9):737-752, 1999.

[11] A. C. Kak and M. Slaney, "*Principles of Computerized Tomographic Imaging*," IEEE Press, 1988.

[12] D. Raparia, J. Alessi, and A. Kponou, "The Algebraic Reconstruction Technique (Art)," Los Alamos Preprint Archive, quant-ph/9709014, 1997.

[13] S. Chuai-Aree, C. Lursinsap, P. Sophatsathit and S. Siripant, "Fuzzy C-Mean: A Statistical Feature Classification of Text and Image Segmentation Method," *Proc. of Intern. Conf. on Intelligent Technology 2000*, December 13-15, Assumption University Bangkok, Thailand, pp. 279-284, 2000.

[14] N.A. Mohamed, M.N. Ahmed, and A. Farag, "Modified Fuzzy C-Mean in Medical Image Segmentation," *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*. vol.6, pp.3429-3432, Piscataway, NJ, USA: IEEE, 1999.

[15] T. Poston, T. T. Wong and P. A. Heng, "Multiresolution Isosurface Extraction with Adaptive Skeleton Climbing," *Computer Graphics Forum*, Vol. 17, No. 3, pp. 137-148, September 1998.

The Detection and Tracking of GSM Portable Handsets Using a 5-Element Circular Array

J.R. Lambert-Porter, A.J. Wilkinson

Department of Electrical Engineering
University of Cape Town, South Africa
lmbjon001@mail.uct.ac.za ajw@eng.uct.ac.za

Abstract

Direction Finding (DF) is a process that involves estimating the directions of the arrival (DOA) for propagating wavefronts impinging on an antenna array from arbitrary directions relative to that antenna array.

GSM, the *Global System for Mobile Communications* is a mobile digital communications system which has rapidly gained acceptance on a global scale since the early 1990s. Because of its popularity on a global scale, it would be desirable to investigate the feasibility of the detection and tracking of such signals as an extension for DF platforms that are used by monitoring authorities such as the police or service providers.

This paper presents a correlative DF algorithm that is suitable for detecting and inferring DOAs for portable GSM handsets. The algorithm is applied to real datasets obtained in the field, the results of which are presented and discussed together with future work for the tracking of these handsets.

1. Introduction

GSM, is undoubtedly the fastest growing mobile communications system and currently spans over 200 countries. Because of its unprecedented growth, it would be useful for a DF platform to have the ability to infer the DOAs for GSM signals emitted from GSM portable handsets. This paper discusses the feasibility of detecting and tracking portable GSM handsets using a correlative direction finding technique. An initial experiment is described in which blocks of data are captured with a circular 5-element antenna array, using a block sampling DF platform. After suitable signal processing of these datasets, DOA estimates are obtained for each captured block.

In order to fully appreciate the nature of the problem at hand, a brief discussion of the *relevant* aspects of the GSM standard is covered in Section 2. The correlative DOA algorithm is presented in Section 3, with simulated followed by real datasets being presented in Sections 4 and 5. Conclusions and recommendations are presented in Section 6.

2. GSM Architecture

This section serves to briefly summarise the aspects of the GSM standard that are relevant to the problem at hand. The reader is asked to refer to [1, 2, 3, 4] for additional information.

2.1. TDMA / FDMA Access Scheme

GSM 900 uses a combination of a TDMA (Time Division Multiple Access) and FDMA (Frequency Division Multiple Access) schemes and makes use of two frequency bands, namely the uplink (mobile to base station) and downlink (base station to mobile) bands. These occur from 890 to 915 MHz and 935 to 960 MHz respectively. The bands are divided up into 125 narrow band carrier channels. Each channel is assigned a unique *Absolute Radio Frequency Channel Number* (ARFCN), and is 200 kHz wide (the compactness is as a result of the GMSK digital modulation scheme). In practice the first carrier is discarded to allow for possible out-of-band interference. The uplink portion of the TDMA/FDMA scheme is shown below:

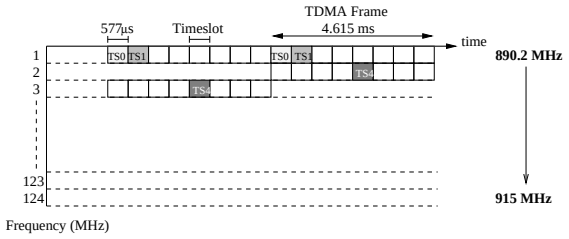


Figure 1: TDMA/FDMA uplink access scheme

Eight timeslots (comprising one TDMA frame) are assigned to a particular carrier on which up to 8 users can transmit and receive information to and from the serving base station. Reception occurs three timeslots after transmission on a carrier in the downlink band which is spaced 45 MHz above the associated uplink channel. Each time slot lasts approximately $577\mu s$, corresponding to a burst of length of 156.25 bits (148 data bits, followed by 8.25 guard bits) per slot. This translates to a gross bit rate of 22.8 kbits/s per time slot. A phone is dynamically allocated a time slot at the start of a conversation, and *maintains* this time slot for the duration of the call (unless the handset undergoes a hand-over from one base station to another).

2.2. Slow Frequency Hopping

To allow for service provision over a large area where the number of subscribers exceeds the number of available channels, GSM allows for the reuse of frequency sets. The 124 available channels are grouped into subsets which are allocated to serving base stations throughout the coverage area. To extend the coverage area, these subsets are reused where co-channel inference between base stations using of the same frequency subset is negligible.

At the start of a call, the base station may instruct the phone to enable slow frequency hopping ($\approx$ 217 Hops/sec). This involves pseudo-randomly changing the transmission frequency at the end of each TDMA frame in an attempt to average the interference over the frequency subsets. This feature is generally

dependent on the quality of the transmission channel between the mobile and the base station.

2.3. Radio Subsystem Link Control

The *Radio Subsystem Link Control* is a bi-directional set of protocols that is responsible for assessing the channel quality, and maintaining synchronisation between the base station and the mobile handset. This information is relayed to the base station approximately twice per second. If the channel quality is insufficient, the base station can instruct the mobile to increase its power level in steps of 2 dBm from 13 dBm up to the maximum power which is dictated by the power class of the mobile (typically 2W). If channel quality is still insufficient, the mobile may re-tune to a new carrier supported by the current base station, or in severe cases, hand-over to a new base station.

2.4. DTX (Discontinuous Transmission Mode)

It has been shown that during a conversation, each speaker speaks approximately 40% of the time. To conserve battery life, a voice activity detector in the handset is used to detect the presence of speech, and when no speech is detected the transmitter is turned off. This is known as *Discontinuous Transmission* (DTX). This means that apart from the channel measurements that are relayed to the base station periodically, the phone essentially transmits nothing apart from the occasional background noise sample.

3. Direction Finding

Direction finding of signals is not a new concept. Since the 1960's, substantial research in this area has been conducted and several methods have been proposed for ascertaining the directions of arrival (DOA) for several types of signals [5, 6]. This section describes a typical, DSP based DF platform that one might use to inspect the GSM band, followed by a correlative DOA algorithm that may be used on such a platform to infer the DOAs for incoming wavefronts.

3.1. Direction Finding Platform

3.1.1. DF Antenna Array

In order to estimate the DOA for an incoming wavefront, a carefully constructed antenna array must be used to capture the signals. Because of the spacing of the antenna elements, the wavefront arrives at each element with a varying time delay that is dependent on its direction of arrival. This time delay, is referred to as the time difference of arrival (TDOA), τ and results in unique set of instantaneous phase differences between the antenna elements for each DOA.

For the purposes of this research, a circular 5 element antenna array was constructed, and is depicted in Figure 2. The elements themselves are tuned monopoles (approximately quarter wavelength) and are positioned onto a ground plate of radius 15 cm. A radial element spacing of 12.5 cm was chosen as a tradeoff between the amount of antenna shadowing, and the DOA ambiguities that arise as a result of the element spacing being too large.

3.1.2. Block Sampling Scheme

Because continuous sampling generates a huge amount of recorded data, a block sampling scheme allows for the storage of recorded datasets recorded over several minutes. Each of the

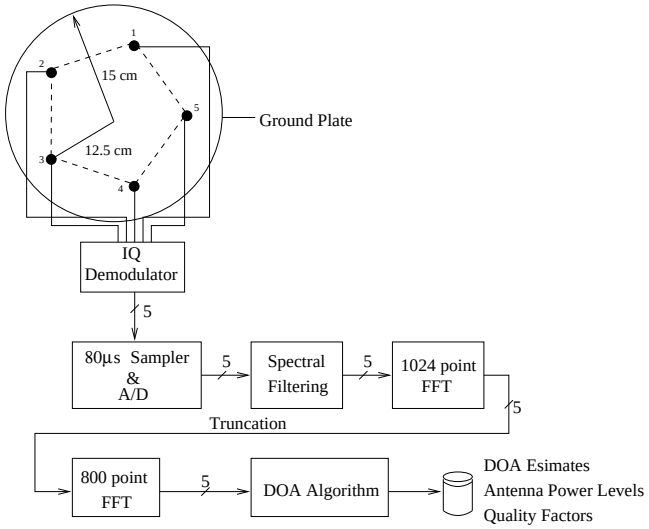


Figure 2: Signal Processing Stages for a typical DF Platform

5 channels of the DF antenna are sampled in parallel at an effective sample rate of 12.8 MHz per antenna channel. Time domain data is sampled in 40 or 80 μs blocks depending on the desired frequency resolution. Table 1 lists the two most commonly used options. Due the FFT computation on the DSPs, the final bandwidth is reduced to 10 MHz for both cases, discarding the aliased edges of the band.

Sample Capture Time	Freq. Resolution	Freq. Bins
40 μs	25 kHz	400
80 μs	12.5 KHz	800

Table 1: Hardware Sampling Options

This dataset for each of the captured channels is then down converted, FFT'd, spectrally filtered using a blackman window (to reduce frequency domain ringing due to the premature truncation of the signal) and finally truncated before being applied to the DOA algorithm. This dataset processing results in a 2 ms delay between successive $80\mu s$ captures. The $40\mu s$ capture window is generally used only when "scanning" the band before switching to an $80\mu s$ capture window for recording.

At the end of each capture performed, the DOA estimates, antenna signal power levels, and the *Quality Factors* (a measure of the certainty of the DOA estimate) are written to disk. The quality factor will be discussed in more detail later. A diagram showing the DF platform as a whole is shown in Figure 2.

3.2. Correlative Direction of Arrival Algorithm

3.2.1. Algorithm Definition

The correlative DOA algorithm correlates the recorded phase differences from every antenna element pair, with a table of precomputed phase differences providing the estimated directions of arrival (and corresponding degree of match) for which the correlation is strongest. In this way, it is very similar to the Generalised Cross Correlation method proposed by Knapp [7]. Before describing the correlation function, the notion of an aperture must be introduced. An aperture is simply an antenna pair.

Because there are 5 antenna elements, there are $n = 1 \ldots 10$ unique apertures which may be formed. To obtain an estimate for the direction of arrival, the captured apertures must be compared with a *characterisation table* $V_n(\omega, \theta)$ of predetermined aperture phase quantities whose phase information should match that of the captured signals at the direction of arrival θ.

The normalised correlation coefficient $C(\omega, \theta)$ is formed, where a perfect match between the recorded data and the characterisation table results in a purely real coefficient equal to 1.

Finally, the factor $Q(\omega, \theta) = \Re e \{C(\omega, \theta)\}$ is used as a measure of the match and will be referred to as the *quality factor* of the estimate where: $-1 \leq Q \leq 1$. The estimated DOA is given where $Q(\omega, \theta)$ attains a maximum for *each* frequency component present in the captured signal.

3.2.2. Characterisation

The process of obtaining $V_n(\omega, \theta)$ is known as *characterisation*. To characterise the DF antenna, a signal generator is connected to a transmitter which transmits a continuous sinusoid at the appropriate frequency at 0 degrees. The antenna is physically rotated from $0^\circ - 360^\circ$. The aperture information is recorded for all apertures at each DOA and are stored in the table before rotating the antenna to the next DOA. In a GSM DF environment, the uplink band is 25 MHz wide, and the centre frequency of which is 903 MHz. The fractional bandwidth is so small (2.8%), that one characterisation table may be used for all the frequencies in this band, rather than computing a new table for each frequency.

Although the characterisation may be done in software thereby generating an ideal table, it is better to obtain the table in the field, as antenna shadowing, and the misalignment of antenna elements is compensated for during the characterisation.

3.2.3. Correlation Coefficient Inspection

To illustrate the correlation coefficients, $C(\omega, \theta)$, two GMSK wavefronts impinging on a simulated antenna similar to that in Figure 2 at frequencies of 898 and 903 MHz were simulated at $DOA = 45^\circ$ and $DOA = 180^\circ$. The SNR was set at 20 dB on each element. An image of $Q(\omega, \theta)$ is shown in Figure 3.

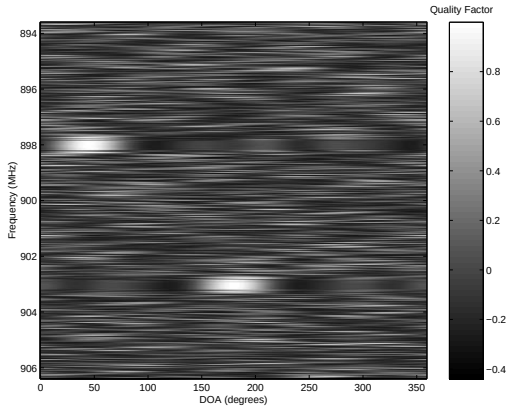


Figure 3: Image of the Quality Factor $Q(\omega, \theta)$ revealing the locations of two incoming signals as two bright spots for which $Q(\omega, \theta) \approx 1$

The DOAs can clearly be seen for the two incoming wavefronts. If the correlation coefficients are plotted for a slice through the 898 MHz carrier, we observe the following in Figure 4.

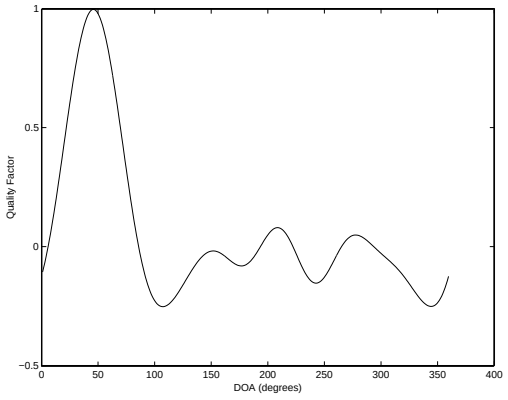


Figure 4: A plot of the correlation coefficients for a slice through the 898MHz carrier

The main peak is correctly observed at $DOA = 45^\circ$ and has a quality factor of 1. Secondary peaks can be seen at $DOA = 210^\circ$ and $DOA = 280^\circ$ as a result of phase ambiguities. In low SNR environments, these peaks can exceed the true peak, and can result in erroneous DOA estimation.

4. GSM Simulator and Display Algorithm

In order to gain a better understanding of the datasets that would be recorded with such a DF platform, a software simulator was developed in MATLAB to model the important characteristics of both the GSM standard (such as frequency hopping, DTX mode, power control etc.), and the DF platform (such as the block sampling nature of the platform, the capture times, and capture bandwidths etc).

Recall, that the platform is typically configured to sample $80\,\mu s$ of data every 2 ms and that a GSM TDMA time slot repeats every 4.6 ms. As a result of this sampling mismatch, a particular time slot moves in and out of view of the capture window over time, realigning with the edge of TS0 after 30 TDMA frames. This is more clearly shown in Figure 5 where TS0 can be seen moving in and out of the capture window.

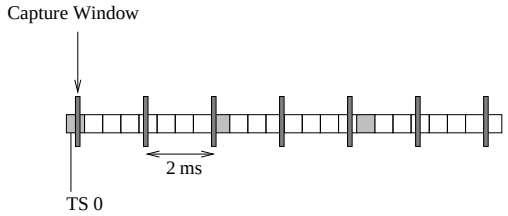


Figure 5: Block Sampling of Timeslots

This moving in and out results in short gaps in the dataset. Because the transmitter of the phone may also be turned off during a period when the time slot is in view of the capture window (DTX mode), the interval between sample instants where

the timeslot is sampled is not constant and large gaps may appear in the dataset. A high level simulator diagram is shown in Figure 6. The display algorithm will be discussed shortly after a simulated scenario has been presented.

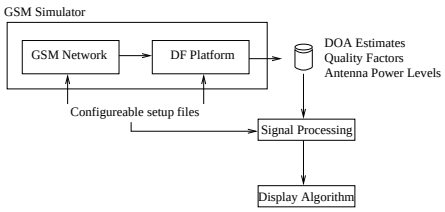


Figure 6: Overview of Simulator Flow Diagram

In order to test the simulator, a scenario was simulated, the details of which are discussed shortly.

4.1. Simulation 1

To test the simulator, GSM data were simulated for approximately 138s (30000 TDMA frames). Three phones (maximum power transmission = 2W) were assigned time slots 3, 3 and 4. The base station was arbitrarily allocated ARFNs = [13, 20, 33, 48] which correspond to frequencies [892.6, 894, 896.6, 899.6] MHz. The geometry is illustrated in Figure 7. Pseudo random frequency hopping over the carriers was activated for the simulation. During the course of the simulation, phone 1 and phone 3 moved along the lines indicated so that the output of the DOA could be verified. Phone 2 remained stationary.

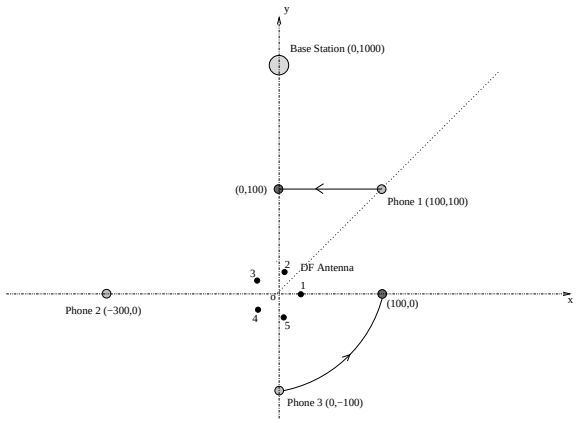


Figure 7: Geometry for Scenario 1

4.1.1. Results

Two plots can be generated from the data recorded, a spectogram, and a estimated DOA plot. Because of the size of the files written to disk (approximately 150 MB for 140 s of data), either every nth captured frame can be displayed, or alternatively n sequential captures can be compressed into one display frame. The danger in displaying every nth frame, is that potentially good DOA estimates can be skipped over. Because of this, the first option will be ignored, and we will turn our attention to the second method.

The spectogram data is thresholded above the noise floor to focus solely on the mobile transmissions. By extracting the frequency components above a particular power threshold, and then extracting from these, components above a certain quality factor threshold (say 85%), it is possible to map the frequency axis to a DOA axis. After setting a level threshold exceeding -30 dB, and quality factor exceeding 95%, the following spectogram and DOA plot were observed in Figure 8 and Figure 9 respectively.

Each vertical line of the spectogram and DOA plot constitutes the data from 67 captured frames. It is difficult to identify from the spectogram plot how many mobiles are present in the area, or if they are actually hopping. The only useful information that an observer can infer from this plot, is the number of carriers over which the phones are potentially hopping.

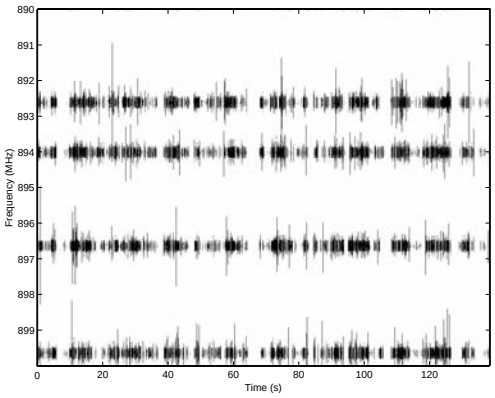


Figure 8: Spectogram for Simulation 1

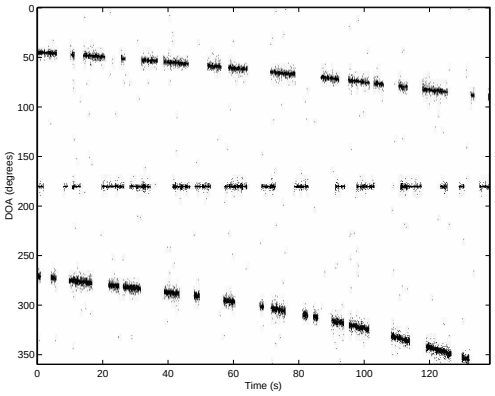


Figure 9: DOA for Simulation 1

We can clearly see the tracks of the three phones in the DOA plot. If we look carefully at the DOA plot, we can see that speckles occur at arbitrary directions of arrival. This is because the level threshold is not sufficiently high to discard these points (by making it too high, some useful data points may be also be discarded). We can also see that several data points concentrate around the true DOA. The spread is due to the fact that DOA estimates are corrupted by noise. If the algorithm could incorporate the GSM frequency information (in particular the known bandwidth of the GMSK signals, and the GSM carrier frequen-

cies), better DOA estimates could be computed by averaging the aperture information over the GMSK band, as the phase information for a particular aperture should be virtually identical across the 200 kHz bandwidth of a GMSK transmission.

For each GSM carrier location ω_c, the averaged aperture information for N values across a 200 kHz wide GMSK waveform is defined as:

$$\overline{S_n(\omega_c)} \quad = \quad \frac{1}{N} \sum_{l=-N/2}^{N/2} S_n(\omega_c + l\Delta\omega) \qquad (1)$$

where $\Delta\omega$ represents the FFT resolution and $S_n(\omega)$ refers to the nth captured aperture. For a 12.5 kHz resolution, $N = \frac{200}{12.5} = 16$.

The resulting DOA vs time image is shown in Figure 10 and shows a definite improvement over Figure 9. The speckle has been reduced, and the points around the true DOA have converged.

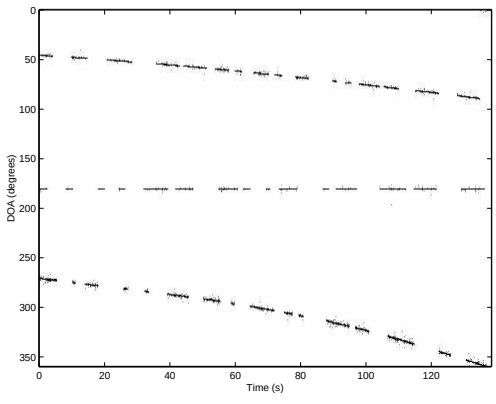


Figure 10: Improved DOA Estimates for Simulation 1

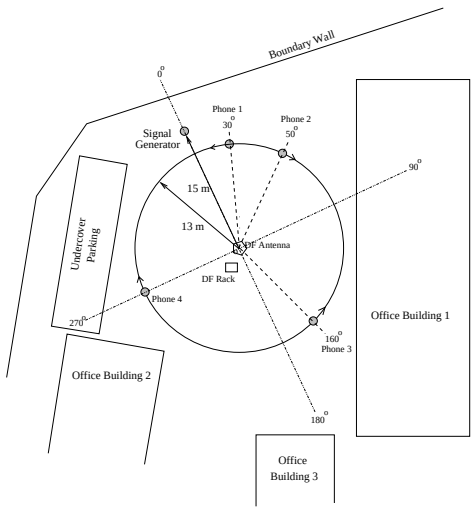


Figure 11: Geometry for Real Data Set Acquisition

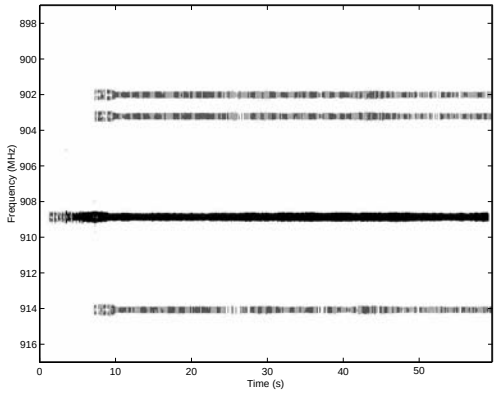


Figure 12: Spectogram for Real Dataset

5. Real Dataset Analysis

In order to inspect real data sets, recordings of cellphone emissions were taken in an empty car park using a wide band DF platform capable of a 20 MHz bandwidth capture every 2 ms (basically two concatenated 10 MHz captures, of 25 kHz frequency resolution). Four MTN network phones were made use of in the experiment, and the geometry is shown in Figure 11. The operator of each phone was instructed to walk in a complete circle about the DF antenna as shown. A signal generator was positioned at 0° for characterisation, but was turned off at the start of the experiment. Hardware limitations did not permit both aperture information and direction information to be stored, and in this first experiment, only direction information was stored.

5.1. Results

Data were displayed with a level threshold of -70 dBm and a quality factor threshold of 80%. The following was observed for the spectogram in Figure 12 and the DOA plot in Figure 13.

It was observed that four RF channels were active, but with a higher concentration on the 908.8 MHz carrier (ARFCN=91). In fact, visually it is virtually impossible to determine how many phones are actually present via inspection of the spectogram alone. Note that the results illustrated in Figure 13 did not undergo the aperture averaging operation. This also presents an extreme case, as most phones would not change angle this quickly as a function of time.

Cellphone tracks are visible, however it is difficult to determine which points belong to which phones without knowing a-priori, the positions of the phones and their motion paths during the recording. However, if we refer back to Figure 11, noting the starting positions of the phones and their directions of travel, we can estimate and classify which points belong to which phones over time. This is shown in Figure 14.

It is predicted that if the averaging operation mentioned in Section 4.1.1 is applied to the aperture data before computing the directions of arrival, the uncertainty in the estimates will be reduced significantly.

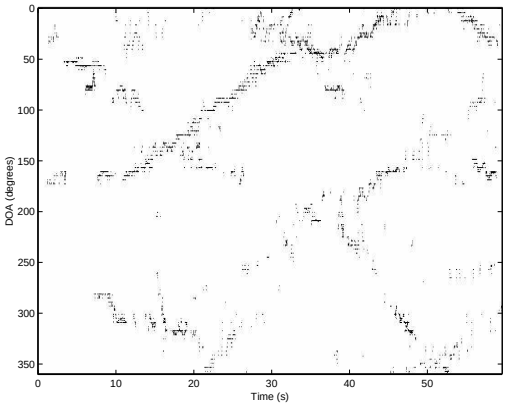


Figure 13: DOA Estimation for Real Dataset

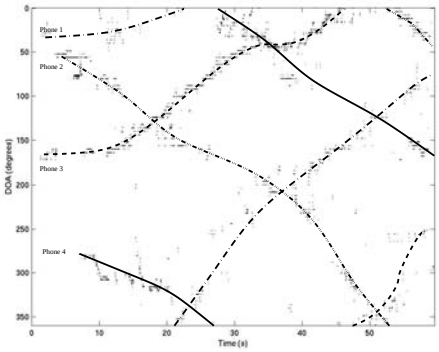


Figure 14: Identification of mobile paths in dataset for Scenario 2

6. Conclusions and Recommendations

From the research conducted thus far, it is concluded that the 5 element DF antenna array will be suitable for detection of mobile phones. At this stage of the research, we have seen that it is possible to spatially separate out and observe the motion paths for a number of phones in an area, if they are spatially unique. It was also observed that by averaging the aperture information around the GSM carriers, improved DOA estimates could be obtained, although this has not yet been implemented on the real data.

The reader should note that conditions were not ideal for the recording of the real datasets. The surrounding buildings would have caused reflections, and in addition, characterisation was performed by mounting the DF antenna on a stepper motor. Work should be done on refining the calibration procedure, in particular averaging the apertures for a each DOA, to average out noise. Currently, the timing information from the datasets has not been explored. If the DF block sampling technique is aligned to the GSM network (not necessarily at the start of a TDMA frame), it would be possible from sample to sample to compute which data point belongs to a particular time slot.

7. Acknowledgements

We would like to thank Peralex Electronics, Cape Town, South Africa for financial support and for providing access to GEW direction finding equipment without which this research would not have been possible. The NRF is also acknowledged for their support via the THRIP programme.

8. References

[1] ETSI Publications, http://www.etsi.org, *GSM Standards*, Various.

[2] C. J. Eberspacher, H.Vogel, *GSM Switching Services and Protocols*. John Wiley and Sons, 2003.

[3] A. Mehrotra, *GSM System Engineering*. Artech House Publishers, Boston, London, 1997.

[4] J. W. V.K Garg, *Principles and Applications of GSM*. Prentice Hall, 1999.

[5] K. Varma, "Time-delay-estimate based direction-of-arrival estimation for speech in reverberent environments," Master's thesis, Virginia Polytechnic Institue and State University, Oct. 2002.

[6] M. V. H. Krim, "Two decades of array signal processing research," tech. rep., IEEE Signal Processing Magazine, 1996.

[7] C. Knapp and G. Carter, "The generalized correlation method of estimation of time delay," *IEE Trans. Acoustics, Speech and Signal Proc.*, 1976.

Using Randomization in the Analysis of MRI data

G R Drevin

School of Computing
Middlesex University, London, United Kingdom.
and
School of Computer, Statistical and Mathematical Sciences
North-West University, Potchefstroom, South Africa.
g.drevin@computer.org

S M Smith

Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB)
University of Oxford, John Radcliffe Hospital, Oxford, United Kingdom

Abstract

Randomization is used to analyse the MRI scans of a group of subjects to determine whether subjects with hypertension are more prone to have a stroke lesion at a given voxel.

1. Introduction

With parametric hypothesis testing a null-hypothesis is accepted or rejected by comparing the value of a statistic with a known distribution. Randomization is used were no assumptions about the distribution of a statistic can be made. The data is reordered randomly and the value of the statistic is calculated for this random reordering. A distribution of the statistic is obtained by repeating this process a large number of times. The original value of the statistic, calculated for the original ordering of the data, is then compared against the randomized distribution obtained in this way. The significance level of the value of the statistic is the proportion of values in the randomized distribution that are equal to, or greater than, the original value of the statistic.

2. Methodology

The analysis was done using a group of 189 subjects, of whom 181 had a scan at a second time point and 110 at a third time point. Based on clinical data the subjects were divided into two groups, the HT group with hypertension subjects and the No-HT group with the non-hypertension subjects (Table 1). The ages of the subjects ranged from 46 to 79 years with an average of 61 years. 94 subjects were male and 95 were female. The scans were done at approximately three year intervals.

Analysis was done using four subsets of the data. The first subset (S1L1) consists of all the scans taken at time point 1. The other three subsets consist only of the subjects that had scans at all three time points, with S1L3 consisting of the scans at time point 1, S2L3 the time point 2 scans and S3L3 the time point 3 scans. The first subset has 189 (68 HT and 121 No-HT) subjects while the latter three have 110 (43 HT and 67 No-HT) subjects each (Table 1). The lesions were identified and drawn in by hand. Using FLIRT [3], all scans were registered to a standard space (avg152T1).

Table 1: *Number of subjects in each clinical group at each time point.*

Time point	HT	No-HT	Total
1	68	121	189
2	65	116	181
3	43	67	110

For each voxel it was then determined what proportion of subjects in each of the HT and No-HT groups had that voxel labeled as a lesion voxel:

$$\begin{aligned} P_1 &= \frac{n_1}{N_1} \\ P_2 &= \frac{n_2}{N_2} \end{aligned} \tag{1}$$

where N_1 and N_2 are the number of HT and No-HT subjects respectively, n_1 and n_2 are the number of subjects with a given voxel labeled as a lesion voxel and P_1 and P_2 are the resulting proportions.

For each of the voxels the null-hypothesis tested was that the proportion of HT subjects with a lesion at that voxel is not greater than the proportion of No-HT subjects:

$$\delta_P = P_1 - P_2 \not> 0 \tag{2}$$

with δ_P being the test statistic. The alternative hypothesis is that the proportion of HT subjects with a lesion in a given voxel is greater than the proportion of No-HT subjects:

$$\delta_P = P_1 - P_2 > 0. \tag{3}$$

In the randomization therefore only the voxels for which $P_1 > P_2$ were used. Table 2 gives the number of voxels that were

Table 2: *Summary of subset statistics. The HT+No-HT Voxels column gives the number of voxels which subjects in either one or both the HT and No-HT groups have labeled as lesion. The HT>No-HT Voxels column gives the number of voxels for which the proportion of subjects in the HT group that have that voxel labeled as lesion is greater than the proportion of No-HT subjects.*

Group	HT+No-HT Voxels	HT>No-HT Voxels
S1L1	15181	10542
S1L3	13176	10128
S2L3	16576	13430
S3L3	19878	16726

identified as lesion voxels as well as the number of voxels for which (3) is true.

Each of the four randomization experiments were done using the observed sample as well as 49999 randomized samples giving 50000 samples. The groups in the randomized samples had the same number of subjects as the groups in the the observed sample, therefore, in the case of S1L1 one group representing a random sample for the HT group would be 68 randomly chosen subjects and the other, representing the No-HT group, the remaining 121 subjects. In the case of the randomization experiments for the other three subsets there were 43 and 67 subjects in each group. For each randomization the proportions, P_1 and P_2, and the test statistic, δ_P, were calculated for each voxel. The number of test statistics obtained from the randomized samples that were greater or equal to the observed value were counted. These counts were then used to calculate the Z-value for each voxel.

Table 3: *Clusters identified by EASYTHRESH.*

		Max Z			CoG		
Cluster	Max Z	x	y	z	x	y	z
S1L1C1	4.11	30	37	50	31	37	50
S1L1C2	4.11	55	37	52	55	38	50
S1L3C1	3.13	30	38	49	31	38	49
S1L3C2	3.08	56	35	47	57	37	49
S1L3C3	3.16	61	66	46	61	66	47
S2L3C1	4.11	30	36	44	31	37	48
S2L3C2	4.11	60	37	47	58	36	48
S2L3C3	2.05	57	72	46	54	60	54
S2L3C4	2.43	35	63	54	34	60	54
S3L3C1	4.11	33	37	48	31	38	50
S3L3C2	3.63	60	37	47	58	36	47
S3L3C3	3.13	61	65	48	56	58	53
S3L3C4	4.11	29	76	45	31	51	50
S3L3C5	3.03	55	58	56	55	57	55

3. The multiple comparison problem

Given that between 10000 and 17000 hypothesis tests were done (HT>No-HT Voxels column of Table2) one could expect between 100 and 170 false positives due to random chance if a significance level $\alpha = 0.01$ was used for each test. One way to correct for this problem is to use the Bonferroni correction whereby the significance level is divided by the number of comparisons or hypothesis tests. Therefore, to reject a null-hypothesis, the value of the statistic obtained has to be less than α/N, where N is the number of comparisons that were done. This leads to an extremely conservative test. However the correlation between voxels that are is close proximity to one another make the Bonferroni correction inappropriate.

To overcome the multiple comparison problem EASYTHRESH was used to eliminate false positives from the resulting Z-values. Different input Z-values were used resulting in clusters which got smaller as the Z-values increased. A number of clusters (Table 3) were identified, with the following transitions between time points:

- S1L3C1 → S2L3C1 → S3L3C1
- S1L3C2 → S2L3C2 → S3L3C2
- S2L3C4 → S3L3C4 (which also contains S3L3C1)

The cluster number after the "C" in the codes are chosen so that they refer to the same cluster in the different images. The center of gravity (CoG) of the clusters varied slightly as different input Z-values were used, but generally the variation was only within one or two voxel spaces in each dimension. The CoG given in Table 3 is therefore a rounded mean position. Clusters S2L3C3 and S3L3C3 were eliminated by EASYTHRESH with input Z-values larger than 1.55 (Table 4).

Table 4: *Clusters identified by EASYTHRESH.*

Input Z	1.55	
Cluster	p	Voxels
S1L1C1	0.0157	246
S1L1C2	0.0577	190

Input Z	2.0	
Cluster	p	Voxels
S1L3C1	0.0005	124
S1L3C2	0.0125	78
S1L3C3	0.0625	57

Input Z	2.0		2.3	
Cluster	p	Voxels	p	Voxels
S2L3C1	1.87e-05	235		
S2L3C2	1.92e-04	183	4.82e-05	130
S2L3C4	0.0956	67		

Input Z	2.0		2.3	
Cluster	p	Voxels	p	Voxels
S3L3C1			6.47e-04	114
S3L3C2	2.56e-06	338	2.15e-04	131
S3L3C4	1.53e-10	663		
S3L3C5	6.85e-04	187		

Clusters 1 and 2 are at the right and left posterior while clusters 4 and 5 are at the right and left anterior. Cluster 3 is in close proximity to cluster 5. In S3L3 with input Z equal to 2.0 cluster S3L3C4 contains both clusters 1 and 4. (Figure 3)

4. Number of randomizations

Generally the literature states that "a large number" of randomizations should be used. Marriott [5] investigated the relationship between the number of simulations in Monte Carlo tests and the probability of accepting or rejecting H_0 at significance levels of 1% and 5%. Manly [4][pages 80–84] expanded on this work and states that, except in extreme borderline cases, a minimum of 1000 randomizations are needed to obtain a significance at the 5% level while a minimum of 5000 randomizations are needed to be able to obtain a significance at the 1% level.

Table 5: *Intervals in which 99% of estimates will fall. Determined using L_E with $p = 0.05$ and $p = 0.01$. N is the number of randomizations used.*

	L_E	
N	$p = 0.05$	$p = 0.01$
5000	0.0422–0.0581	0.0046–0.0138
10000	0.0445–0.0557	0.0075–0.0127
15000	0.0455–0.0547	0.0080–0.0122
20000	0.0461–0.0540	0.0082–0.0119
50000	0.0475–0.0525	0.0089–0.0112

Table 6: *Intervals in which 99% of estimates will fall. Determined using L_M with $p = 0.05$ and $p = 0.01$. N is the number of randomizations used.*

	L_M	
N	$p = 0.05$	$p = 0.01$
5000	0.0420–0.0580	0.0064–0.0136
10000	0.0444–0.0556	0.0075–0.0126
15000	0.0454–0.0546	0.0079–0.0121
20000	0.0460–0.0540	0.0082–0.0118
50000	0.0475–0.0525	0.0089–0.0111

Edgington [1][page 50] proposed that to test randomization programs one should use the program to determine a p-value for a data set for which the true p-value is known or determined by systematically generating all possible permutations. The interval in which 99% of estimates should fall is given by

$$L_E = \frac{(N-1)p \pm 2.58\sqrt{(N-1)p(1-p)} + 1}{N} \quad (4)$$

where L_E is the upper and lower limits of the interval, N is the number of randomizations and p is the true p-value. This equation, however, results in an asymmetrical range round p (Table 5). Rewriting (4) as:

$$L_E = \frac{(N-1)p \pm 2.58\sqrt{(N-1)p(1-p) + 1}}{N} \quad (5)$$

results in a symmetrical range round p. Under the assumption that the estimated p-values are normal with mean p and variance $p(1-p)/N$, the interval

$$L_M = p \pm 2.58\sqrt{p(1-p)/N} \quad (6)$$

would contain 99% of the estimates (Manly [4][page 82]). This gives the interval for $N = 1000$ and $p = 0.05$ as (0.032 0.068) while the interval for $N = 5000$ and $p = 0.01$ is (0.006 0.014). Using this result Manly once again argues that 1000 randomizations is a reasonable minimum for a test at the 5% level and that 5000 randomizations is a reasonable minimum for a test at the 1% level

The intervals in which 99% of estimates should fall given $p = 0.05$ and $p = 0.01$ and various numbers of randomizations are shown in Table 5 calculated using L_E and in Table 6 calculated using L_M. Using (5) to calculate L_E results in values that are equal to L_M as in Table 6.

Jackson and Somers [2], however, state that in many biological studies too few randomizations have been used. Using four different data sets, they found that there was a 5%–6% variation in the estimate using between 500 and 2000 randomizations.

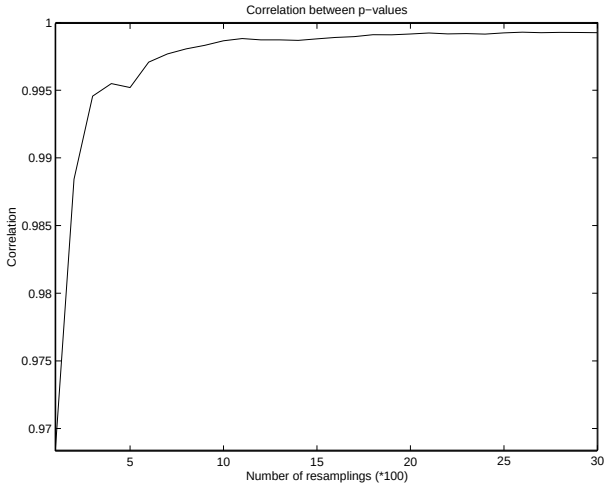


Figure 1: *Correlation between two randomization experiments. The correlation between the p-values of the two experiments are shown after every 100 randomizations.*

This variation fell to less than 1% when 10000 to 50000 randomizations were used and was as low as 0.1% with 100000 or more randomizations. Their recommendation was that 10000 to 50000 randomizations be used and that this be increased to 100000 where the observed probability approaches a critical value. Manly comments on this that using this many randomizations would depend on how serious one is to determine an exact p-value. If one is only interested in the significance level as a measure of the strength of evidence against the null hypothesis then 1000 or 5000 randomizations would be adequate.

To empirically determine the effect of the number of randomizations two randomization experiments were done using the same data subset. With each of the randomization experiments p-values were determined after every 100 randomizations. The correlation between the two sets of p-values increased rapidly to a level of 0.998 over the first 1000 randomizations (Figure 1) after which the increase was much slower. After 50000 randomizations the correlation between the two sets of p-values was 0.9999.

Furthermore it was found that the difference between the sequence of p-values and the final p-values decreased rapidly over the first 1000 to 2000 randomizations (Figure 2) after which the decrease was much slower. The maximum absolute difference went below the 0.01 level after between 15000 and 20000 randomizations.

5. Conclusions

The use of randomization for hypothesis testing on MRI data was demonstrated. In the case of MRI data there are a large number of hypothesis tests that have to be done and therefore some form of thresholding of the results have to be done. An exact significance level, as opposed to the acceptance or rejection of a null-hypothesis, is needed to do a thresholding. It was shown that where an exact significance level is needed one needs to do as many as 50000 randomizations.

Randomization is computationally expensive therefore every effort should be done to streamline the process. In the case of MRI data one way to do this would be to identify all the voxels of interest and only do the randomization for those voxels.

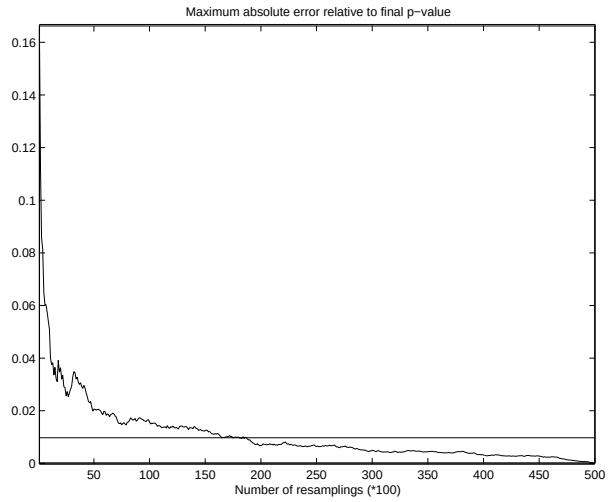


Figure 2: *The maximum absolute difference between the p-values after every 100 randomizations and the final p-values. The final p-values were obtained after 50000 randomizations.*

The method was successfully used to identify the voxels where subjects with hypertension are more prone to have a stroke lesion than subjects who do not suffer from hypertension (Figure 3).

6. References

[1] Edgington, E.S., Randomization Tests. Third edition. Marcel Dekker, Inc., New York. 1995.

[2] Jackson, D.A. & Somers, K.M., Are Probability Estimates from the Permutation Model of Mantel's Test Stable? *Canadian Journal of Zoology*, 67:766–769, 1989.

[3] Jenkinson, M., *et al.* Improved optimisation for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841, 2002.

[4] Manly, B.J.F., Randomization, Bootstrap and Monte Carlo Methods in Biology. Second edition. Chapman & Hall, London. 1997.

[5] Marriott, F.H.C., Barnard's Monte Carlo Tests: How Many Simulations? *Applied Statistics*, 28(1):75–77, 1979.

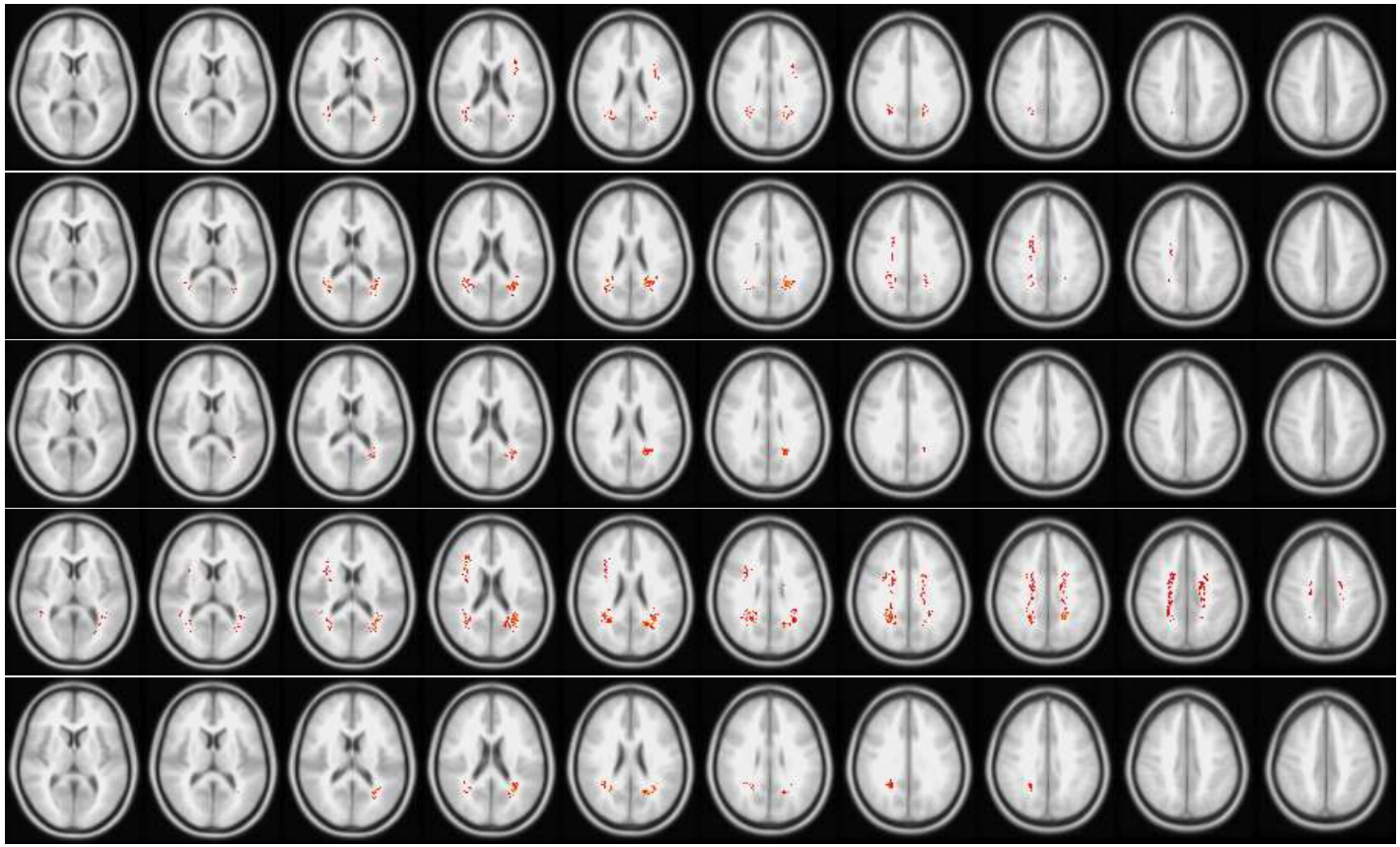


Figure 3: Every second slice from slice 40 to slice 58 is shown. The images from top to bottom are: S1L3 (Z=2.0), S2L3 (Z=2.0), S2L3 (Z=2.3), S3L3 (Z=2.0) and S3L3 (Z=2.3).

Reducing Inter-Agent communication due to negotiation in Multi-Agent systems through Learning

Bradley Van Aardt
School of Electrical and Information Engineering
University of the Witwatersrand
Johannesburg
South Africa
b.vanaardt@ee.wits.ac.za

Tshilidzi Marwala
School of Electrical and Information Engineering
University of the Witwatersrand
Johannesburg
South Africa
t.marwala@ee.wits.ac.za

***Abstract* – This paper studies the effect that agent learning can have on inter-agent communication in a Multi-agent system. The agents are equipped with MLP neural networks to learn solutions to problems that have been solved through explicit negotiation and communication. We implement a test problem in the form of a Pursuit game, where the Multi-Agent system is a set of captor agents. The result is up to 44% fewer negotiation sessions with learning-enabled agents. The importance of learning, in terms of agent knowledge and overall system effectiveness is discussed.**

I. INTRODUCTION

The concept of Multi-agent systems is an engineering paradigm that has been gaining momentum over the past years [1]. A particular form of Multi-Agent systems, Swarm based systems, have been successfully applied to a number of problems [2]

Swarm systems are popular because of the lightweight processing power needed for them. A particular advantage in some instances is the low level, or often no direct, communication between agents, especially when communication bandwidth is limited or costly [2]. In [3], for example, the time taken for a system during negotiation to solve a task for a particular planning scheme, GraphPlan, is noted. The amount of time can be seen to increase significantly on some problems. This is potentially very inefficient, especially if the agents face the same problem frequently. However, it is often considered an absolute fundamental attribute of agents to be able to communicate with other agents in the system to improve the overall performance of the system [4].

In this paper, a study is performed on the effect that learning has on communication between agents. It is reasoned that once an agent community has solved a particular problem successfully, there should be no reason to repeat all the steps that led to the solution. In other words, the agents should be able to re-use solutions that they have discovered previously. In the experiment described here, the agents first negotiate a solution for each problem they are presented. Each agent then learns the problem situation from its point of view, and the resultant action that it took after successful negotiation with other agents. Once they have learned the appropriate behaviour, they need not perform the task of negotiation for that situation, as long as the same conditions hold. It is reasoned that this can significantly cut down on inter-agent communication traffic.

II. BACKGROUND

A Multi Agent Systems

The multi-agent paradigm, in which many agents operate in an environment, has become a useful tool in solving large scale problems through a "divide and conquer" strategy [5]. The Multi-agent system is a distributed, decentralised system. The paradigm of individual entities collaborating to solve a particular problem that is beyond each entities own capabilities is a natural concept, and one that is proving to be very powerful in practice [6]. However, while the concept is easily understandable, the implementation is not trivial. There are many complexities and subtleties in these such as [5]:

- Decomposing and allocating problems to the agents
- Describing the problem to the agents
- Enabling communication and interaction among the agents

Decentralised systems, in the context of Multi-Agent systems, promise the following advantages [5][7] [8][9][10]

- No single failure point, therefore greater robustness. Multi-agent systems have the capacity to degrade gracefully.
- Possibility of faster response times and fewer delays as the logic/intelligence is situated nearer to the problem domain.
- Increased flexibility to take account of changes occurring in the problem domain.
- Modularity and Scalability. Multi-agent systems can be increased in size dynamically according to the demands of the problem.

The problems associated with Multi-agent systems are [10][2][5][1]:

- Difficulty in measuring and evaluating the stability and security of the system.
- Excessive communication between agents can slow down the system. This is often countered by heavily restricting the amount of communication between agents.
- Possibility of getting stuck in non-optimal solutions of the problem, often due to the lack of global knowledge of the problem from each agent's point of view.

- Most Multi-agent systems are built in an ad-hoc way since there is no absolute theory for these types of systems. There have been recent attempts however, to formalise the design of agent based systems, such as the Gaia Methodology [7]. However these are not yet in widespread use.

B Swarm Based systems

Swarm intelligence is a particular paradigm for multi-agent systems which emphasizes distributedness and agent simplicity. It is based on the observations of social insects in nature, such as ants, termites and bees [2]. Such insect societies are extremely organized, even though there is no central control or planning. Each agent in the system is programmed only to achieve its own Local Goal. The agent's behaviours in Swarms are very simple: the intelligence of the system is 'emergent' from the overall behaviours of all the agents in the system. The communication between agents is usually performed indirectly [9], by agents making changes to the environment, which other agents act upon. This is analogous to insects laying pheromone trails to food source, etc. Emphasis is therefore placed on reactivity in these systems. Swarm systems have been successfully applied to many problems, notably routing in computer and telecoms networks [2] and recently to a manufacturing control system [9].

The disadvantage of swarm based systems is that no agents actually have a global view of the problem to be solved. All agents are entirely focused on achieving their own Local Goals, whether or not these goals are to the benefit or detriment to the overall community. This can exacerbate the problem of the system getting stuck in local optima, or worse, cause the system to fail.

C Neural Networks

Artificial neural networks are inspired by the functioning of biological neurons in animal and human brains. Artificial neural networks differ from most other computing techniques in that they are able to learn arbitrary relations between sets of data presented to them. That is, rules are not explicitly programmed or set, but are learned from experience by the network [10]. The basic architecture of a neural network is also based on that of their biological counterpart: both networks consist of many simple elements, known as neurons, operating in parallel [12] [13]. The most widely used neural network models are the Multi-layer perceptron (MLP) and Radial Basis Function (RBF) networks.

III. HYPOTHESIS AND METHOD

Reference [11] states that the method of human learning when presented by a new task is to use rational reasoning to perform the decision making process behind solving the task. After using this deliberative approach, and we begin to "master" a task, we are able to perform it "naturally", without explicitly performing rational reasoning. At this point, people use their pattern recognition skills to perform the task.

It is hypothesised that the human model of discovery and learning can be applied to agent strategising. An algorithmic discovery stage using negotiation between agents can be used on new problems. This will be the deliberative stage, where a problem solution is formulated. Once a strategy has been discovered and used, a neural network can be trained with the results. This is the Pattern Recognition stage. Thus over time, the neural networks should learn a large number of strategies, while generalising these strategies to other scenarios. Thus, as agents perform more tasks, the problem solving for known tasks should become "natural" to the *entire community*, as the neural networks start taking over from the negotiation modules.

Our proposal is to create a system that uses negotiation to enable the agents to find solutions to problems they encounter, and then learn this solution, in order to avoid the necessity of re-negotiating a solution, with the resultant communication and time costs, if the problem occurs again. Learning is considered to be an important feature of true autonomous agents [5]. One aspect of learning can be expressed as the ability to perform old tasks better as a result of learning and observing [4]. In this case it can be interpreted as the agents learning to perform old tasks more efficiently.

A Test Problem

The proposed system is applied to the game of Pursuit. A Pursuit board is represented by a 2 dimensional grid pattern, as shown in Figure 1. The asterisks represent the Captor agents, while the circle represents the Fugitive agent. The aim of the game is for the Captors to surround or corner the Fugitive. The fugitive and captors take alternate turns to move. Each agent may only move one block per turn. Legal moves on a Pursuit board are Up, Down, Left, Right and Stay. No diagonal moves are allowed. Furthermore, the board does not wrap around. A game is ended when the fugitive is captured, implying that the fugitive has no place to move to.

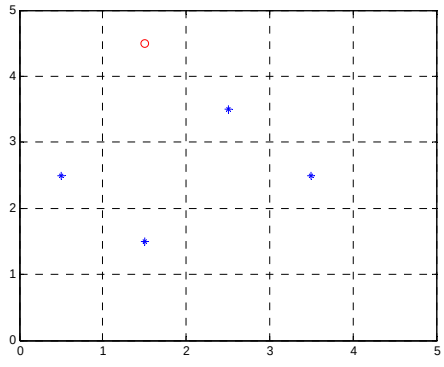


Figure 1 Pursuit Board representation

B System Description

The system has four separate agents representing the captors. Each of the agents has a neural network that is trained independently of other agents, as well as a negotiating algorithm. Each agent is responsible for making a valid move. This is defined as a move that is within the bounds of the board, and does not infringe on any block that another agent is in.

The fugitive agent has no real intelligence; it merely chooses a (legal) move at random.

The agents interact by proposing the move that they would prefer to make. This is sent to all other agents. If there is a conflict among the proposals, the agents send out a signal, and the agents negotiate a new proposal. This occurs until there are no conflicts among the agents. The agents then implement their moves. The first move proposed for each agent is always drawn from the neural network. If this is not successful, the agents start the negotiations.

C Captor Agent Description

As mentioned above, the Captor Agents consist of two parts: a pattern recognition unit, implemented as a neural network, and a negotiating algorithm that solves disputes with other agents.

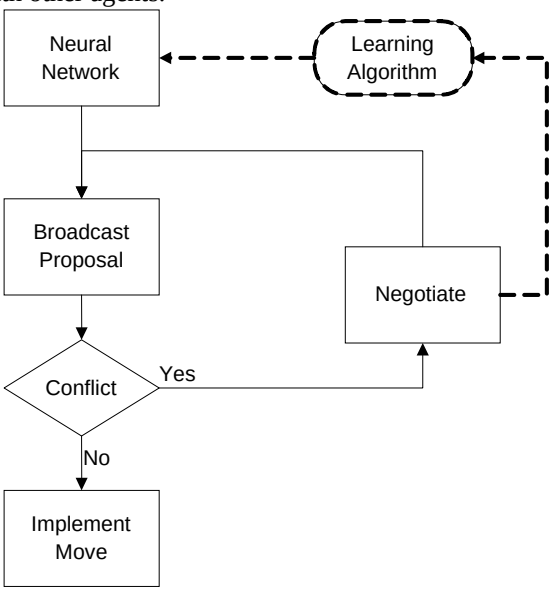


Figure 2 High Level Agent execution

As can be seen in Figure 2, the agents start off by proposing the solutions predicted by their neural networks. Only if this fails is the negotiation algorithm invoked. Whenever there is a conflict, all the agents are involved in the negotiation. The dotted feedback lines from the negotiation block to the neural network represents the network learning from the successful move made by the negotiation algorithm. In this way, the neural network learns cooperative behaviour.

The negotiation algorithm is shown in Figure 3. The agents each have a local goal based on the global goal of the system. The global goal of the system is to win the game, i.e. capture the agent. In this system the global goal is expressed as the minimisation of the distance of all the Captor agents from the Fugitive agents. This is given by:

$$S = \sum_{N=1}^{M} \sqrt{(X_N - X_A)^2 + (Y_N - Y_A)^2} \qquad (1)$$

Where: S is the sum of the distances to the fugitive; M is the number of Captor agents in the system; X_A, Y_A are the Fugitive agent's coordinates and X_N, Y_N are the current Captor's coordinates. If the captors follow this rule, they will win on a finite sized board, although not always in the least number of moves.

It would follow that if each captor agent minimised its own distance from the fugitive, the global goal would be satisfied. However, since each block on the board can hold only one agent exclusively, not all agents will be able to reach their own minimum on every board state. Therefore, to minimise the function in reality, some agents may not be in their optimal positions. In a Centralised system, where one agent controls the movements of all the Captors, this would be straightforward to implement, as the controller would have a global view of the problem, as well as only having to act in its own best interest. However, in a decentralised or Multi-Agent system, the implementation is not as clear. Since each agent will have its own local goal to pursue, it is not clear what the formulation of this local goal would be in order to reap the maximum benefit for the entire community [14].

In this case, the local goal of each captor agent is to minimise its own distance from the fugitive through negotiation with the other Captors. The negotiation algorithm of each agent is depicted in Figure 3. The agents are aware of the possible moves they can make at each turn. By evaluating the payoff that each of the possible moves will bring, the agents construct a Preference List of the possible moves and payoffs ordered from highest payoff (least distance) to lowest (furthest distance). Once the agent has made its list, it transmits its preferred move (which is the block on the board it would like to occupy), and the payoff value of the next preferred move on the list. In this way, the agent expresses its "need" for the block. The agent then creates a list of all the moves and payoffs received from the other agents. If there is a conflict between itself and one or more other agents on the choice of blocks, the agent attempts to resolve this. An agent will give way to the other agent(s) if it does not value the position as highly as any other conflicting agent (i.e, if it does not need the block as badly as other agents). An agent gives way by using the next move on its preference list. Once it gets to the end of its list, it wraps around to the start. If there is a tie for the highest value on the block, the agents involved in the tie go into a "lotto" mode, where they each randomly pick a number, with the highest number winning the right to keep the block. The losing agents then choose the next move on their Preference list. This procedure is iterated until no conflicts exist. At this point the agents implement their moves. In this manner, an approximation to the global optimum is performed. This sequence can be performed a number of times, which can

be seen to create many messages between agents, and thus a high usage of the communication medium.

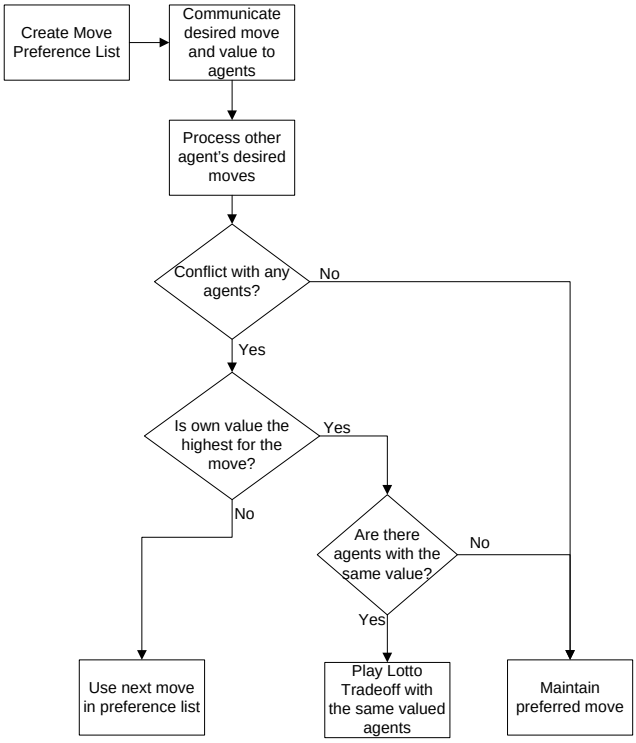


Figure 3 Agent Negotiating Algorithm

From this, it can be seen that the neural network of each agent learns solutions to board problems that are not only legal, but also cooperative with other agents on the board. Once these moves have been learned, the agents need only inform each other of the predicted move, and if there are no conflicts, proceed with the move. This then eliminates the need for the explicit negotiation transmissions.

Board Representation and Training

The input representation to a neural network is extremely important to the success of practical network applications [15][16][17]. Smooth representation of the input data, as well as using an input representation that describes features of the data can help reduce the number of states the network has to learn in a lookup-table type fashion, and increase the ability of the network to generalise learned data to new situations [17].

For this study, we implement the board representation as a set of relative differences in position. Each Captor agent has the relative positions of the other captors inputted in order of Cartesian distance. The relative position of the fugitive is then given as the last two inputs. This is depicted in Figure 4.

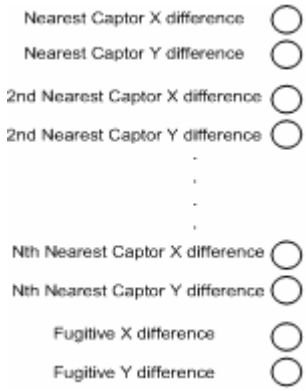


Figure 4 Network Input representation

This representation allows various formations of the agents to be captured, regardless of the exact location on the Pursuit board. Such formations are common when the captors are chasing the fugitive. s

Each Captor is trained with data specific to its own experience on the board. Training the agents from their past experience means that the direct communication between agents can be drastically reduced. This is due to the fact that each agent will take into account other agents moves when moving itself. Therefore, in terms of communication, the system acts as a swarm system, but with one major difference: each agent is "aware" of the global goal through the experience it has gained through negotiation, and so inherently acts cooperatively with other agents.

IV. IMPLEMENTATION

The system is implemented in MATLAB. NETLAB [19] is used to implement the neural networks for the Captor Agents. A two-layer MLP architecture is used for the networks. The networks use a Logistic function for the output units. A variable number of hidden units are used, depending on the amount of data the network has seen.

Since NETLAB does not directly support on-line training, an alternative scheme is used. Every time the agents negotiate a solution, the board position is recorded, as well as the moves agreed upon through negotiation. This data is added to the set of training data. After each game is complete, the agents are re-trained using the updated training data.

V. RESULTS

The simulation was run for 300 games. The results are summarised in Table 1. The measure of effectiveness used is the average number of game moves per game successfully predicted by the neural network. This is shown as the Average Learned moves/Game. The average number of moves per game when not using learning, i.e. using only the negotiation modules, is 6.5 over 300 games. This figure is used as the control number, to measure the quality of the predicted moves. If the average number of moves per game increases from this control, it implies that learning is actually having a detrimental effect on the agent's performance.

The results of the learning simulation are broken into 3 stages of the game: 0 -100 games, in which it can be seen that the neural networks are still learning appropriate moves. The result is that the average number of moves per game in this phase actually increases from the control number, and it is for this reason that a percentage saving in bandwidth is not given, as it is meaningless in this case. A suggestion is this case would be to ignore the neural network's proposal until a critical number of training data has been obtained.

In the second phase, from game number 100 -200, it can be seen that the neural networks have learned sufficient appropriate moves to be effective in the system. In this phase, the average moves/game is at the level of the Control, but the amount of communication is decreased. This shows that the learning is useful in this stage.

In the third stage, the neural networks account for nearly half of the game moves. This represents a substantial reduction in negotiation communication.

TABLE 1

BREAKDOWN OF SIMULATION RESULTS AFTER 400 GAMES

	Game numbers		
	0 -100	100- 200	200-300
Average moves/ game	7.6	6.3	6.23
Average learned moves/game	2.4	1.92	2.7
% Bandwidth Reduction	N/A	30%	44%

VI. ANALYSIS

The knowledge that the agents have obtained as a result of their learning can be thought of as obtaining a model of their environment, or more specifically, as a model of the behaviour of the other agents within the system. This is due to the fact that when the agents correctly predict moves using their neural networks, they are not only maximising their own goal, but taking into account the needs of the other agents in the system. This is despite the fact that they have no explicit knowledge of the behaviour or workings of the other agents: the knowledge that they have obtained is purely through observation and interaction. In the negotiation, or deliberative algorithm, the agents take no account of the moves that the other agents might make: they wait for another agent to actively inform of a problem. To programme into such algorithms the ability to anticipate other agent's moves would effectively mean that the agents would each have a global view of the problem, as well as having intimate knowledge of every agent's abilities and desires in the system. This would defeat the object of a Multi-agent system, where each agent is expected to operate in a highly distributed environment, and having a limited viewpoint of the problem, as well as considerable duplication of abilities. Furthermore, it may not even be appropriate for each agent to have this knowledge, especially when the agents are divided into sub-teams [3].

After a large number of games, each agent will have observed a large variety of board positions, and for this system, the neural networks will tend to be the same. This is only because the agents in this system all have the same "abilities" or allowed moves. In a scenario where there are agents with different abilities, as for example the separate pieces in a chess game, the networks will each train on totally different output moves, and so will tend to be different even after observing an infinite number of moves. However, there is no reason why the agents should not still be able to anticipate each others moves, since they will be trained with this data.

The reduction in communication as shown here can have large benefits in certain situations, such as when the communication channel carries a significant monetary cost, or if the channel is not reliable or slow. All these factors contribute to the overall effectiveness of the agent community, and should be minimised.

VII. FUTURE RESEARCH

There is already considerable research in the field of agent learning. We believe that learning is not only useful for added functionality in agents, but also to enhance existing agent operation. The main problem in any learning system is often the presentation of the information to the system [15]. With other means of representation, it may be possible to increase the benefits of learning, and create more efficient agents. In complicated environments, which are common of multi-agent domains, it would be desirable, and probably necessary, to have the agents automatically determine the relevant representation of information, as well as choosing the information that would help in the decision.

Furthermore, the area of agent coordination is a currently an active area of research. . A methodology whereby agents are equipped with a host of coordination strategies at design time, and then learn through interaction the appropriate local strategy to use in a given situation has been proposed by [18]. A mechanism to accommodate this type of agent behaviour will need to be investigated.

VIII. CONCLUSION

The possibility of reduced communication between agents, while still cooperating with each other is demonstrated in this paper. We show that learning can be successfully applied to this aim. This results in agents that

effectively anticipate other agent's behaviours, despite having no explicit model of the agents programmed.

A reduction in communication of up to 44% is achieved in this study. This level of reduction can have extremely positive effects if communication in the system is in some way a significant cost factor in the overall effectiveness of the system.

IX. REFERENCES

[1] Maria Chli, Philippe De Wilde, Jan Goossenaerts, Vladimir Abramov, Nick Szirbik, Pedro Mariano, Rita Riberio, "Stability of Multi Agent Systems", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* 2003, Volume 1, 2003, Pages: 551-556

[2] E. Bonabeau, M. Dorigo, G Theraulaz, *Swarm Intelligence – From Natural to Artificial Systems*, Oxford University Press, 1999.

[3] D. Kalofonos, T.J. Norman, "An Investigation into Team-Based Planning," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* 2004, pp 5590- 5595.

[4] Z. Ren, C.J. Anumba, "Multi-agent systems in construction – state of the art and prospects," *Automation in Construction*, no. 13, 2004, pp. 421-423.

[5] K.P. Sycara. "Multiagent Systems". *AI Magazine, American Association for Artificial Intelligence*, Summer 1998. pp 79-92.

[6] J.J. Castro–Schez, N.R Jennings, X. Luo, NR Shadbolt. "Acquiring Domain Knowledge for negotiating agents: A case study", *International Journal of Human-Computer Studies* 61 (1), 2004, pp 3-31.

[7] N.R. Jennings, M. Wooldridge, D. Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design", *Autonomous Agents and Multi-Agent Systems*, 3, 2000. pg 285-12

[8] N.R Jennings, M. Woodridge,"Intelligent Agents: Theory and practice", *The Knowledge Engineering Review*, 10 (2), pg 115-152, 1995.

[9] Hadelia,, P. Valckenaersa, M. Kollingbaumb, H. Van Brussel, "Multi-agent coordination and control using stigmergy," *Computers in Industry* 53, 2004, pp 75–96

[10] Nils. J Nilsson, *Artificial Intelligence: A new synthesis*, Morgan Kaufmann Publishers, San Francisco, California, 1998.

[11] R. Kurzweil, *The age of intelligent Machines*, Massachusetts Institute of Technology, 1990. pp 231- 234

[12] Alison Cawsey, *The Essence of Artificial Intelligence*, Prentice Hall Europe, 1998.

[13] M. Jordan, C. Bishop, *Neural Networks*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Last accessed 12 August 2003, ftp.publications.ai.mit.edu.

[14] J.A. Marshall, Z. Lin, M.E. Brouke, B.A Francis, "Pursuit Strategies for Autonomous Agents", *Proceedings of the 2003 Block Island Workshop on Cooperative Control*. Springer-Verlag Series: Lecture Notes in Control and Information Sciences, 2004.

[15] Sebastian Thrun, "Learning to Play the Game of Chess", *Advances in Neural Processing Systems* 7, 1995.

[16] G. Tesauro, "Programming backgammon using self-teaching neural net", *Artificial Intelligence* 134(2002) pp 181-199.

[17] G.Tesauro, "Practical Issues in Temporal Difference Learning", *Machine Learning* 8 (1992), pp. 257-277.

[18] C. B. Excelente-Toledo and N. R. Jennings "The dynamic selection of Coordination Mechanisms" *Autonomous Agents and Multi Agent Systems 9*, 2004, pp 55-85

[19] I. Nabney, *Netlab – Algorithms for Pattern Recognition*, John Wiley and Sons, New York, 2001

Fuzzy clustering for the detection of Tuberculous Meningitis from brain computed tomography scans

W. Halberstadt, T.S Douglas

MRC/UCT Medical Imaging Research Unit, Department of Human Biology, Faculty of Health Sciences, University of Cape Town, Observatory 7925, South Africa, +27 21 4066235
hlbwar001@mail.uct.ac.za tdouglas@cormack.uct.ac.za

Abstract

Computed tomography (CT) is gaining widespread usage as an aid to tuberculous meningitis (TBM) diagnosis for the examination of a number of visual indicators of the disease. We present an algorithm that uses modified fuzzy c-means clustering to segment CT images of the brain into various tissue types. The end result is that hyperdense areas of the brain are delineated, which aids the radiologist in the diagnosis of TBM.

1. Introduction

The rise in HIV infection in South Africa has lead to an increase in cases of Tuberculous Meningitis (TBM) in children. The disease can lead to serious brain damage and death and is difficult to diagnose due to the presence of many non-specific symptoms. The gold standard in testing relies on culturing of the TBM virus; unfortunately the incubation time is four weeks, which is too long for a patient to wait without treatment. The use of Computed Tomography (CT) to detect the disease is speeding up diagnosis and improving prognosis. A new sign associated with TBM, namely hyperdensity in the basal areas of the brain in pre-contrast scans, has been observed by Dr S Andronikou. A hyperdense area is an area of tissue that has increased density, or increased intensity on CT scans, compared to the surrounding tissue. The technique avoids invasive contrast enhancement and is unique to TBM.

The use of clustering techniques in order to separate brain CT images into their separate tissue types will result in abnormal tissue being easier to identify and assist radiologists with their diagnosis. The fuzzy c-means clustering algorithm forms the basis of the segmentation method presented in this paper.

1.1 Image Segmentation

Many segmentation methods are available for clustering images into principle components. The problem with most hard clustering techniques such as k-means clustering, is that they do not take into account uncertainty in an image, leading to inaccurate clustering. In a study by Nevin *et al* (1998), comparisons were made of different clustering methods, by quoting results from various papers. The classical methods, such as thresholding and edge based techniques were the least successful methods of segmenting medical images. Statistical methods were an improvement, but were highly affected by noise. The use of neural networks proved to be superior to the classical and statistical methods, but still had problems with uncertainty. The method of segmentation that was most effective for medical images, was fuzzy clustering.

1.2 Fuzzy Logic

Fuzzy sets were first introduced by Zadeh (1965) as a way of representing uncertainty or vagueness in real world problems. Fuzzy models essentially attempt to capture and quantify non-random imprecision (Bezdek *et al,* 1992).

Fuzzy sets have a way of representing imprecise data by mapping each number in the set into the interval [0:1]. For an example relevant to this paper we look at the case of a CT image. The image is made up of a number of pixels having different grey values. Each pixel can be regarded as having a value that represents its fuzzy membership to the set. The borders of the set are 0 which represents the colour white, and 1 which represents black. Therefore a pixel having grey value 0.8 has a high fuzzy membership to the colour black.

1.3 Fuzzy Clustering

Cluster analysis is based on partitioning data into a number of subgroups or clusters. The objects located within each cluster must show a degree of similarity. In hard clustering such a k-means, each point in the data is assigned to only one cluster. With the use of fuzzy clustering, each pixel has some degree of membership to each cluster. The degree of membership is an indication of how similar or close a pixel is to some criterion (Gath and Geva, 1989).

The advantage of the fuzzy c-means over other methods of segmentation such as classical and statistical methods, is that the algorithm does not require any prior knowledge of the data and it is fairly robust to noisy data.

2. Methods

2.1 Fuzzy c-means clustering

The fuzzy c-means algorithm by Bezdek *et al* (1973) is based on minimization of the following objective function, with respect to U, a fuzzy c-partition of the data set X, and to V, a set of cluster prototypes.

$$J_q(U, V) = \sum_{j=1}^{N} \sum_{i=1}^{K} (u_{ij})^2 d^2(X_j, V_i); \qquad K \le N \qquad (1)$$

Where q is any real number greater than 1 and is a weighting exponent on each fuzzy membership. X_j is the jth *m*-dimensional feature vector, V_i is the centroid of the *i*th cluster. u_{ij} is the degree of membership of the data point X_j in the *i*th cluster, $d^2(X_j, V_i)$ is any distance measure between the cluster centre V_i and the data point X_j, N is the number of data points , and finally K is the number of clusters.

To create a fuzzy partition of the data, iterative optimization of the above equation needs to be carried out. This is done by the following steps:

1. Choose primary cluster centres V_i
2. Compute the degree of membership of each data point to all the clusters. Membership is calculated from equation 2 below:

$$u_{ij} = \frac{\left[\frac{1}{d^2(X_j, V_i)}\right]^{\frac{1}{(q-1)}}}{\sum_{k=1}^{K}\left[\frac{1}{d^2(X_j, V_k)}\right]^{\frac{1}{(q-1)}}} \qquad (2)$$

Where q adjusts the fuzziness of the equation, for q =1 it becomes the simple k-means algorithm. A value of q =2 was used in the algorithm

3. Compute new cluster centres $\hat{V}_i$ according to equation 3 below:

$$\hat{V}_i = \frac{\sum_{j=1}^{N} (u_{ij})^q X_j}{\sum_{j=1}^{N} (u_{ij})^q} \qquad (3)$$

Once the new clusters have been calculated, the degree of fuzzy membership must be updated from u_{ij} to $\hat{u}_{ij}$

Check the termination criterion to determine whether another iteration is required. The criterion is given by the equation:

max $\left|u_{ij} - \hat{u}_{ij}\right| < \epsilon$,

where ϵ is a termination criterion between 0 and 1.

Once the error criterion is reached, the iterative process is complete and the data is separated into a fuzzy partition. From equation 2, when calculating the degree of fuzzy membership, the distance measure $d^2(X_j, V_i)$ is used. When the distance measure represents the Euclidean distance, the fuzzy c-means algorithm is what results.

2.2 Fuzzy Maximum likelihood estimation

To increase the accuracy of the segmentation, especially in images where the clusters have differing densities and the number of points are not equally distributed, the use of the fuzzy maximum likelihood estimation algorithm (FMLE) yields better results. The algorithm was proposed by Gath and Geva (1989). It combines statistical clustering methods with fuzzy logic, yielding an improved overall algorithm.

The FMLE has the advantage of being more robust to noise, but needs well defined starting clusters as it searches for an optimum in a very narrow local region. Therefore the fuzzy c-means algorithm is first run in order to calculate good starting clusters before executing the FMLE.

The FMLE algorithm is very similar to the fuzzy c-means, with the main difference being that the distance measure $d^2(X_j, V_i)$ is no longer the Euclidean distance but changes to an exponential distance measure. This new distance is used to calculate $h(i|X_j)$, the posterior probability.

$$h(i|X_j) = \frac{1/d_e^2(X_j, V_i)}{\sum_{k=1}^{K} 1/d_e^2(X_j, V_i)} \qquad (4)$$

$$d^2{}_e(X_j, V_i) = \frac{[\det(F_i)^{1/2}]}{P_i} \exp[(X_j - V_i)^T F_i^{-1} (X_j - V_i)/2 \qquad (5)$$

where F_i is the fuzzy covariance matrix of the *i*th cluster:

$$F_i = \frac{\sum_{j=1}^{N} h(i|X_j)(X_j - V_i)(X_j - V_i)^T}{\sum_{j=1}^{N} h(i|X_j)} \qquad (6)$$

and P_i is the *a priori* probability of selecting the *i*th cluster:

$$P_i = \frac{1}{N} \sum_{j=1}^{N} h(i|X_j) \qquad (7)$$

The algorithm follows the same procedure with equation (2) being replaced by equation (4) and the additional equation (7) being calculated in step 3 of the algorithm.

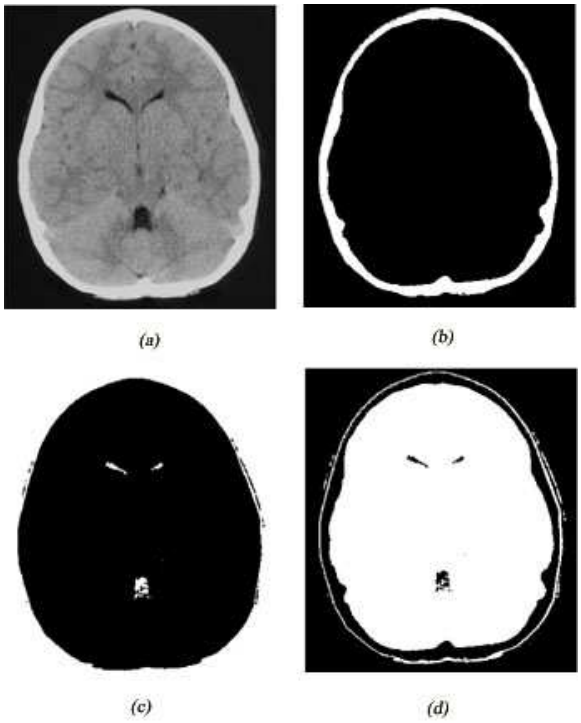


Figure 1: (a) CT brain scan of a normal child. (b) to (d) results of fuzzy clustering into different tissue types. (b) skull, (c) ventricles and (d) brain matter

2.3 Cluster Validity

Due to the nature of the CT images, it is difficult to know how many clusters the image requires for segmentation. A cluster validity measure provides us with a method for deciding whether to add more clusters or not. It is based on whether the clusters have a high density of pixels surrounding them (partition density), and also minimizing the volume of the clusters (fuzzy hypervolume).

Fuzzy hypervolume is calculated as:

$$F_{HV} = \sum_{i=1}^{K} [\det(F_i)]^{1/2}$$

Partition density is calculated from:

$$P_D = \frac{\sum_{i=1}^{K}\sum_{j=1}^{N} u_{ij}}{F_{HV}}$$

From the two validity measures, our definition of a 'good' cluster is one that minimizes the classification error rate; this is true when the fuzzy hypervolume is minimized and the hyperdensity is maximized.

Once the cluster validity is calculated a new cluster must be added and calculating where to locate the new cluster is important to performance and time of the algorithm.

Cosic and Loncaric (1996) compared 3 different equations for placement of new clusters. The approach was to place the new cluster in a region where the data points have a low degree of fuzzy membership. The equation that proved most successful is used in this algorithm:

$$p = \arg\min_j \max_i u_{ij} \quad V_{k+1} = X_p$$

Where the expression calls for the value of *i* that maximizes the expression $u_{ij,}$ and the j value that minimizes u_{ij}.

Once all aspects of the algorithm are combined, the resultant algorithm is one that is unsupervised and automatically decides the optimum number of clusters while performing segmentation.

The final algorithm is the combination of the FCM and the FMLE algorithm and proceeds as follows:

(1) Select 2 prototype clusters
(2) Run FCM
(3) Run FMLE
(4) Check cluster validity
(5) Either add a new cluster and repeat or end

3. Methods and Results

The data sets used in this project were obtained from the Red Cross Children's Hospital in Cape Town. They

included 17 patients that were proven to have TBM by culturing, as well as 40 cases with probable TBM.

A number of normal CT scans were also obtained to act as controls and for comparison.

The CT images first underwent pre-processing in order to reduce extraneous information. They were resized and clipped so that only the relevant information was left for processing, thus reducing run time for the algorithm.

The images underwent histogram equalization and contrast stretching in order to provide a better contrast to the images. The algorithm was then applied, resulting in the images being segmented into the various tissues.

In figure 1, the separate tissue types are displayed, those of brain, skull and ventricle. The number of tissues that each image was segmented into was left to the algorithm's cluster validity measure to decide. When the optimal number of tissues had been found, all the images had 3 different tissue types segmented. Once this was complete further processing was required in order to segment the hyperdense region

The area containing the brain was again segmented using the fuzzy clustering and one of the segmented areas that resulted was that of hyperdense tissue, indicating TBM. Of the 17 positive cases, all images produced areas of non-uniform asymmetrical hyperdensity.
With the 5 negative cases that were tested, all segmented hyperdense regions, but they were symmetrical and uniform around the outside of the brain indicating beam hardening, a common artifact of CT images.

Figure 2 below shows an example of further clustering of the brain tissue. It clearly demonstrates the hyperdensity in the image on the right, as indicated by the arrows. The image on the left is a normal patient, with very little hyperdensity.

The segmented images were also assessed by a radiologist, and the TBM patients were all confirmed to have positive hyperdensity.

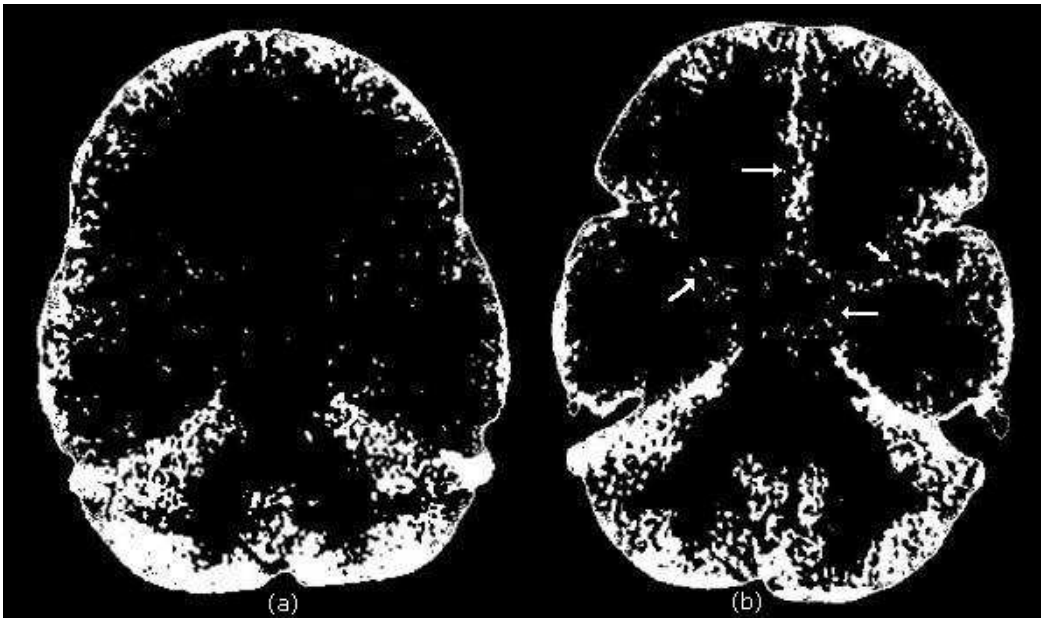


Figure 2: (a) the result of further clustering of the brain segment of a normal brain CT; (b) the result of the clustering when applied to a patient with TBM. The arrows highlight the abnormally hyperdense areas in the TBM patient.

4. Conclusion

An algorithm has been presented for detection of hyperdensity in CT brain scans. The algorithm is based on the Unsupervised Optimal Fuzzy Clustering algorithm presented by Gath & Geva (1989). The method presented differs from that of Gath and Geva (1989) in that the cluster selection method of the algorithm has been optimized for maximum speed of convergence. The use of contrast stretching and histogram equalisation also differed from the original algorithm and contributed to better results. The new method for cluster selection was based on a method proposed by Cosic and Loncaric (1996) in which several different cluster selection methods were compared.

The algorithm was applied to brain CT images of TBM patients in order to confirm a new diagnostic sign presented by Dr S. Andronikou. Pre-processing of the images, the use of histogram equalization and contrast stretching contributed to a fast, accurate algorithm for segmenting CT images.

The results of applying the algorithm to the 17 TBM positive images all produced asymmetrical hyperdensities confirming the sign observed by Andronikou. The application of the algorithm to normal CT images did not produce the asymmetrical hyperdensity and further verified the observation.

The fuzzy clustering algorithm has been successful in segmenting hyperdensity, but may be used to enhance many other features in brain CT images associated with TBM, therefore providing more information for the radiologist on which to base his or her diagnosis.

5. References

Cosic D, Loncaric S (1996) New Methods for Cluster Selection in Unsupervised Fuzzy Clustering. *Proceedings of the 41st Annual Conference of koREMA'96* 4:1-3, Opatijia, Croatia.

Gath I, Geva A (1989) Unsupervised Optimal Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7): 773-781.

Bezdek J (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York.

Nevin , Mohamed A, Ahmed M, Farag A (1999) Modified Fuzzy C-Mean in Medical Image Segmentation. *IEEE Proceedings International Conference on Acoustics, Speech, and Signal Processing* 6: 3429 – 3432.

Zadeh L (1965) Fuzzy sets. *Information and Control* 8:338-353.

Option Pricing Using Bayesian Neural Networks

Michael Maio Pires, Tshilidzi Marwala

School of Electrical and Information Engineering, University of the Witwatersrand, 2050, South Africa

m.pires@ee.wits.ac.za, t.marwala@ee.wits.ac.za

Abstract

Options have provided a field of much study because of the complexity involved in pricing them. The Black-Scholes equations were developed to price options but they are only valid for European styled options. There is added complexity when trying to price American styled options and this is why the use of neural networks has been proposed. Neural Networks are able to predict outcomes based on past data. The inputs to the networks here are stock volatility, strike price and time to maturity with the output of the network being the call option price. There are two techniques for Bayesian neural networks used. One is Automatic Relevance Determination (for Gaussian Approximation) and one is a Hybrid Monte Carlo method, both used with Multi-Layer Perceptrons.

1. Introduction

This document deals with the use of two kinds of Bayesian neural networks applied to the American options pricing problem. Both Bayesian techniques used were used with Mult-Layer Perceptron (MLP) networks. The techniques can also be used with Radial Basis Function (RBF) networks [1] but they were only used with MLP networks here. The two Bayesian techniques used are Automatic Relevance Determination (ARD) (for Gaussian Approximation) and the Hybrid Monte Carlo method (HMC) which will be discussed.

Firstly we need to introduce the notion of an option. An option is the right (not the obligation) to buy or sell some underlying asset at a later date but by fixing the price of the asset now [2]. For someone to have this option, he/she has to pay a fee known as the option price. There are two kinds of options, namely a call and a put option. A call option gives the person the right to buy the underlying asset and a put option gives the person the right to sell the underlying asset [2]. The pricing of either call or put options is equally difficult and something that has brought much research interest.

Black et al. [3] provided equations in 1973 that provided a pricing formula for call and put options. To obtain these equations, several assumptions had to be made. The most important assumption made is that the formulas only held for European styled options [4]. European styled options only allow the exercise of the option on the maturity date (which is the later date that the person is allowed to buy or sell the underlying asset) [5]. What are used extensively worldwide, though, are American styled options where the person is allowed to buy or sell the underlying asset at any date leading up to the maturity date. This introduces another random process into the pricing of the option (because it cannot be predicted when the exercise of the option will occur) and so the pricing of these kind of options is much more complex than European styled options [6].

Neural Networks (NN's) are a form of prediction based on trends that have occurred in the past. The outputs of the network are that which are to be predicted and the inputs are chosen as variables that affect the outputs in the real world and whose trends can be used to predict the output variables. MLP and Support Vector Machines (SVM's) have been used to price American options [7] and here what will be tested is the effectiveness of Bayesian Neural Networks.

2. Bayesian Neural Networks

2.1 Bayesian Techniques

With NN's there is always an error in the predictions made and we thus have

$$y = f(x; w) + \varepsilon \tag{1}$$

where y is the actual output desired, f is the output predicted by the network, ε is the error, $\boldsymbol{w}$ are the weights [1] and $\boldsymbol{x}$ is a vector of inputs. Even if we are given ε and the same network is run twice with the same parameters, we will obtain different weights $\boldsymbol{w}$ both times and thus there is an uncertainty in the training of the networks [1] and this can be attributed to the

randomness in the assignment of weights. Generally some complex models try to fit the noise into the predictions which cause problems when trying to predict with unseen inputs (the problem of over training) and thus cause there to be even more error in the predictions [1].

p(.) wherever used from now on is used to denote the probability function from statistics. In the Bayesian approach, the uncertainty in the parameters estimated when training a network is assumed to follow a particular distribution. We first start with a *prior* distribution $p(\mathbf{w})$ which gives us an idea of the parameters before the data is used [1] but this only give us a vague idea as the distribution is quite broad. The prior distribution can be of any kind for example Poisson or Geometric. In this case we will only use a Geometric distribution. We then wish to narrow this distribution down by finding the posterior probability density of the parameters *w* given a particular dataset *D*, $p(\mathbf{w}|D)$ where

$$p(w \mid D) = \frac{p(D \mid w)\, p(w)}{p(D)} \tag{2}$$

and $p(D|\mathbf{w})$ is the dataset likelihood and $p(D)$ is the evidence and ensures that the posterior integrates to 1 and is calculated by an integral over the parameter space. Once the posterior

$$p(D) = \int p(D \mid w')\, p(w')\, dw' \tag{3}$$

is calculated we can then make a prediction at a new input by first calculating the prediction distribution

$$p(y \mid x^*, D) = \int p(y \mid x^*, w)\, p(w \mid D)\, dw \tag{4}$$

where $\mathbf{y}$ is the predicted values and then the actual prediction is found by

$$E(y \mid x^*, D) = \int y p(y \mid x^*, w)\, p(w \mid D)\, dw \tag{5}.$$

E(.) is the expected value in statistical terms. As can be seen from equations (3) and (5), there is an integral involved and the dimensionality of the integral is given by the number of network parameters (weights) and this is not analytically possible and simple numerical algorithms break down [1]. Therefore approximations to the posterior are made (the toolbox used to train Bayesian Neural Networks is the NETLAB toolbox used with MATLAB®) and this is known as the evidence function in NETLAB and is used together with a Gaussian Approximation and ARD (see section 2.2). What can also be used is Hybrid Monte Carlo (HMC) methods combined with Monte Carlo sampling used for integral approximation [1] (see section 2.3).

The main reason for the use of Bayesian techniques is simply to reduce the uncertainty in the weights and thus try to reduce the problem of over fitting (i.e. over fitting occurs when a network predicts badly because it is trained too much to its training data and predicts badly with unseen inputs [1]). Bayesian techniques do reduce the problem of over fitting as has been proved by Nabney [1]. In NN's there is a need to optimize the network and thus reduce the error function [8]. In Bayesian techniques this is done by obtaining a posterior distribution for the weights so that they can only be found within a particular distribution thus narrowing the search for the optimal weight values [1]. Bayes' theorem helps us do this but there are large integrals and there are several ways of evaluating these integrals. There are Gaussian Approximations and HMC.

2.2 Automatic Relevance Determination

The prior distribution is chosen to be Gaussian [1] and thus is of the form

$$p(w) = \frac{1}{Z_W(\alpha)} e^{-\frac{\alpha}{2}\sum_{i=1}^{W} w_i^2} \tag{6}$$

where the normalization constant $Z_W(\alpha)$ is

$$Z_W(\alpha) = \left(\frac{2\pi}{\alpha}\right)^{W/2} \tag{7}.$$

α is known as the hyperparameter because it is a parameter for the distribution of other parameters. It is then helpful to have different hyperparamaters, one for each set of the weight sets $W_1 \ldots, W_g$. The way to choose these different hyperparameters is to have values for them associated to how important each input variable is. This is known as Automatic Relevance Determination (ARD).

ARD is used because there is often the need to find the relevance of certain input variables. This is not easily done if there are hundreds of input variables. In Bayesian NN's we associate each hyperparameter with an input variable. Each hyperparameter represents the inverse variance of the weights and so the lower the value for a hyperparameter associated with a particular input, the more important that input is in the prediction process because it means that large weights are allowed [1].

2.3 Hybrid Monte Carlo Method

As stated before, Monte Carlo methods can be used to approximate the integrals involved in Bayesian techniques rather than using a Gaussian approximation with ARD and an evidence procedure [1].

Since there is an uncertainty in the process, we need to find the predictive distribution, i.e. the distribution that represents the possible outcomes of the network due to the uncertainty in the weights [1]. This distribution is an integral but in Monte Carlo methods it is approximated to a sum

$$p(y \mid x, D) = \frac{1}{N} \sum_{n=1}^{N} p(y \mid x, w_n) \quad (8).$$

where N is the number of samples chosen by the trainer of the network and $\boldsymbol{w}_n$ is the sample of weight vectors. These samples of weights can be chosen through different methods. A Metropolis-Hastings algorithm can be used to sample these weights but has proved to be very slow. This is because the method makes no use of gradient information and for NN's the method of error back-propagation provides an algorithm for evaluating the derivative of an error function and thus optimizing the network more computationally efficiently [1]. Another method that can be used is the Hybrid Monte Carlo (HMC) algorithm for sampling which is the one that is used in this application and makes use of the gradient information.

The HMC algorithm is a sampling algorithm that takes into consideration certain gradient information. The algorithm follows the following sequence of steps once a step size ε and the number of iterations L has been decided upon:

1) Randomly Choose a Direction λ: λ can be either -1 or +1 with the probability of either being chosen being equal.

2) Carry Out the Iterations: Starting with the current state $(w, p) = (\hat{w}(0), \hat{p}(0))$ randomly selected, where p is a momentum term which is evaluated at each step, we then perform L steps with a step size of $\lambda\varepsilon$ resulting in the candidate state $(\hat{w}(\lambda\varepsilon L), \hat{p}(\lambda\varepsilon L)) = (w^*, p^*)$.

3) The candidate state is accepted with probability $\min(1, e^{(-H(w^*, p^*) - H(w, p))})$ where *H(.)* is the Hessian matrix. If the candidate state is rejected then the new state will be the old state.

These three steps, in essence, describe how the sampling is done so that the summation of equation (8) can be accomplished and so that the posterior distribution can be found and thus allowing the optimization of the NN. The momentum term p can be randomly generated or it can be changed dynamically at each step and there are different ways of doing this [9]. The sets of weights are thus selected or rejected according to the three steps above and the number of samples that are wished to be retained are the number of weights retained. For each set of weights there is a corresponding NN output. The prediction of the network is the average of the outputs.

The usefulness of the Bayesian approach comes into the fact that the prediction comes with certain confidence levels. In fact the prediction mathematically is the same as that of the standard MLP. If we plot the prediction and upper and lower bounds (where the upper bound is the prediction plus the standard deviation of the outputs and the lower bound is the prediction minus the standard deviation of the outputs of the network) then we say that the prediction is known to within a certainty of 68% (because in the normal distribution 1 standard deviation from the mean constitutes 68% of the possible outcomes [10]). This is done for the Gaussian and HMC approaches.

3. Results of Bayesian Neural Networks

3.1 Automatic Relevance Determination Approach

Data was obtained from the JSE Securities Exchange of South Africa. It was obtained for a particular stock option for the period January 2001 to December 2003. This resulted in there being 3051 points of data that could be used for training and testing of the networks. The inputs to the network were stock volatility, strike price and time to maturity (in days). The output of the network would simply be the call price of an option. Call prices were obtained for different options with there being both high and low prices. What was decided was to use the average of the high and low prices as the actual call price and these are the values used to train and test the network.

There are demos available in the NETLAB toolbox that show the procedure of training Bayesian NN's with the Gaussian Approximation and ARD, and HMC. These demos were edited so that the procedures could be experimented with on the options pricing problem. In the Gaussian Approximation with ARD, it was found that 500 training cycles showed the best results with 1000 data points being used to train the network. The network was tested with 300 data points so that the plots could be easily seen when viewing the error bars. The evidence procedure utilized in the toolbox has a certain amount of cycles associated with it as well and it was found that 10 cycles for this sufficed for the training of the Bayesian NN. The parameters changed were the number of hidden units, the number of loops used to find better hyperparameter values and the value for β that is associated with MLP NN's and is the coefficient of data error associated with the MLP. The results of the Gaussian Approximation approach with ARD can be seen in table I.

There was a problem when trying to find the standard deviations of the outputs for the Bayesian NN's using the ARD approach. The function that provides the standard deviations, at times, produced some imaginary numbers so what was done was to search through the standard deviations and replace the imaginary numbers with the first standard deviation value in the array. This got rid of the errors in MATLAB® but showed that the ARD approach does have some bugs. In fact it is said that the Gaussian approximation is the same as the HMC under certain conditions but these conditions are not known and in fact the only reason that Gaussian approximations are used in Bayesian techniques is because they are more mathematically neat than other Bayesian approaches.

As can be seen from table I, the network performed the best with the coefficient of data error at 10, with 50 hidden units and the number of loops to find different hyperparameter values only set to 1. The values found for the different hyperparameters show that each input was important in the determination of call prices because each hyperparameter was in the same order of magnitude and there isn't one that is significantly smaller or

TABLE I
ARD RESULTS

β	Hid. Uni-ts	Mean Error (%)	Ti-me (s)	n	σ	Alphas
1	25	64.7	52	1	1516	[1.2177 1.2036 0.6417]
10	25	53.16	52	1	1512	[0.8366 0.9051 0.5723]
100	25	61.94	49	1	1354	[1.0101 0.9760 0.4525]
1	50	57.44	104	1	1541	[1.2231 0.9342 1.1528]
10	50	52.72	103	1	1505	[1.5385 0.8931 0.8203]
100	50	61.16	102	1	1485	[0.7763 1.2248 0.9373]
1	25	62.1	105	2	1390	[1.7646 0.9891 1.1858]
10	25	70.49	97	2	1520	[1.8534 0.7471 0.9004]
100	25	58.95	102	2	1433	[1.1214 1.5703 0.4662]
1	50	78.49	191	2	1521	[2.0210 1.5175 0.7859]
10	50	76.56	177	2	1409	[1.7518 1.2180 0.6775]
100	50	61.85	175	2	1456	[1.9264 1.3430 0.7552]

β = coefficient of data error for the MLP, Hid. Units = number of hidden units used in the training of the MLP, Mean Error = average error found by subtracting each prediction from the actual value and multiplying by 100 over the size of the test set used (300), Time = time taken to train the network, n = number of loops used to find the best hyperparameter values, σ = the average size of the bounds for all the outputs (average of standard deviations of output samples), Alphas = hyperparameter values found for the corresponding input to hidden unit weights thus showing the importance of the different inputs.

larger than the others. The time column indicates that the networks didn't take too long to train and that if the number of hidden units was doubled so the time to produce a result also doubled (give or take a few seconds). Other values were tried for hidden units and what was also tried was to use more training data to improve the accuracy of the pricing model. It was found that with 1500 training points and 100 hidden units the mean error was much higher than the values found in table I and also took up to 30 minutes to train. Note that to obtain these results the algorithm had to be run several times with the same parameters so that the best results for these parameters could be obtained, this is due to the random nature inherent in the algorithm for training the NN as was found with standard MLP's [7]. The standard deviations found for each network trained are quite large and thus the predictions found by the network are known to be within a range of about R3000 with a certainty of only 68%. Therefore we can only say that we know the price to be within quite a large range (of R3000) and only with a confidence of 68%. The outputs for 50 of the 300 test points used and with the corresponding confidence levels for the 2nd network in table I can be seen in Figure 1.

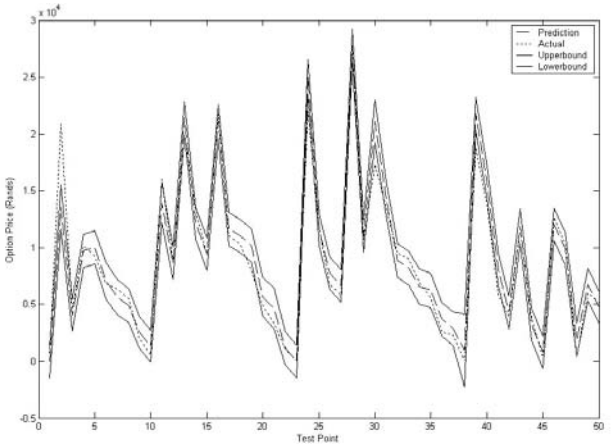


Figure 1: Bayesian NN with ARD Results. Note the upper and lower bounds are solid and notice that they are quite broad.

3.2 Monte Carlo Approach

The data used to train and test the HMC Bayesian NN was the same data as that used for the ARD approach. Here the coefficient of data error value was not experimented with and was rather kept at a value of 10. The number of hidden units was experimented with as well as the number of initial samples rejected and the number of samples in the HMC procedure. The step size was kept constant at 0.002 because it was found that if it was changed to other values bigger or smaller then the threshold (probability used in the rejection criteria) was not a number (NaN in MATLAB®) and so the procedure didn't work very well in these cases. The number of training points used was also 1000 and the number of points used to test the network was 300. The results for the HMC Bayesian NN approach can be seen in table II.

As can be seen from table II, the networks took quite sometime to train with 1000 training points. It was attempted to try fewer points for training but just reduced the performance of the network significantly. What was also attempted was to use more hidden units to train the network but this proved to increase the amount of time required to train the network with no improvement in the error analysis. Note that the algorithm for each result in table II was found by training the same network only once. It didn't have to be run several times. The process of training networks in this was still random but the seed used for the random number generator was the same every time and so there was no difference between the results of two networks that were trained with the same parameters. The

standard deviations found by each network trained are significantly smaller than that found by the Gaussian approach with ARD. Therefore the predictions of the network are known with a confidence of also 68% to be within a certain range but the range is much smaller and at best the range was R802.84. The outputs for 50 of the 300 test points used and with the corresponding confidence levels for the 1st Network in table II can be seen in Figure 2.

TABLE II
HMC RESULTS

Rej.	Max Error (%)	Mean Error (%)	Samp.	Time (s)	σ	Hidden Units
100	5241	76.07	100	259	445.95	10
100	5990	95.68	100	444	502.72	20
100	4372	82.76	100	816	699.36	40
100	4378	98.31	400	648	468.71	10
100	5212	77.92	400	1114	575.67	20
100	6719	98.26	400	2104	814.61	40
200	5662	79.21	100	390	401.42	10
200	7618	103.42	100	665	684.75	20
200	4021	91.80	100	1227	680.83	40
200	3849	92.04	400	777	472.08	10
200	4093	78.29	400	1322	591.20	20
200	5836	78.53	400	2451	722.30	40

Rej. = number of samples to be rejected initially (at the start of the Markov chain), Max Error = Maximum error between the actual output and that predicted by the network in the 300 point test set used, Mean Error = average error of the size of the test set used (300), Samp. = number of samples in the HMC method, Time = time taken to train the network, σ = the average size of the bounds for all the outputs (average of standard deviations of output samples), Hidden Units = number of hidden units used in the MLP.

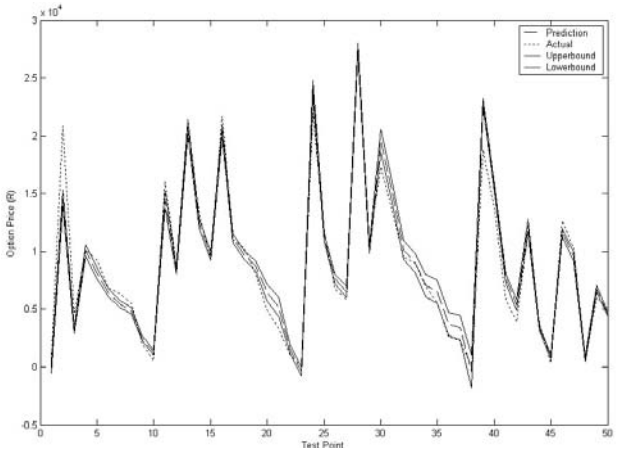


Figure 2: Bayesian NN with HMC results. Note that the bounds are solid and notice that they are not as broad as in the ARD approach.

4. Comparison of Bayesian Techniques with Standard Multi-Layer Perceptrons and Support Vector Machines

From the results obtained for the standard MLP and SVM [7], it must be said that the Bayesian techniques applied to NN's didn't provide any improvements. In fact mathematically they are said to be the same as standard NN's but the advantage they bring is the actual confidence levels. With regards to the ARD approach, the best level of mean error was found to be 53% which is very close to the 51% found by the standard MLP trained before. The amount of time taken to train the network was much more than that found by the standard MLP as was to be expected due to the extra functions being utilized in the Bayesian approach due to the approximations inherent in the technique. Compared to SVM it was faster than the 7 minutes taken to train an SVM network but the results were significantly poorer because the average error found by the SVM network at best was 34.4%.

With regards to the HMC approach the best value found for average error over the test set was found to be 76.07%. HMC is mathematically supposed to provide the same results as standard MLP's but it didn't in this case. This is probably because not enough samples were taken when obtaining a prediction. With there being 400 samples the network took up to 40 minutes to train and so for the purposes of this study what was considered to be more interesting is the fact that HMC provided a much narrower band of confidence than that found by the Gaussian approach with ARD. The band produced by the HMC approach was R804.84 which is significantly better than the R3000 found by the ARD approach. Therefore even though the error found by the HMC approach was found to have at best an average of 76.07% we know that the price given by the network is known to be within a band of R804.84 with a confidence of 68%. A drawback is of course the time taken to train the network using HMC. It takes very long but is still more useful than standard MLP's and MLP's with the ARD approach.

In conclusion the best NN method was found to be the SVM method because it produced the best error analysis results and even though it took 7 minutes to train it is worth using in the future. But it must be said that Bayesian NN's do produce confidence levels for the outputs which is still a serious advantage over standard NN's when pricing options. This is because what can be done is to say that a price is provided with this degree of confidence and thus we can then see the implications of adding a bit to the price because we know the confidence or subtracting from the price. Based on this we can see that optimally a Bayesian SVM approach would be favorable and this could be further researched.

5. Conclusion

The algorithm that worked the best for the option pricing problem is the SVM algorithm. It produced the best error analysis results even though it takes a bit longer to train than standard MLP NN's and Bayesian MLP NN's with ARD. What can be attempted in the future is to use some optimization approach (such as Particle Swarm Optimization or Genetic Algorithms) to obtain the optimum number of weights and values for other parameters so that the best Bayesian NN can be found. This may prove to be very computationally intensive and may take a very long time especially with the HMC approach with Bayesian NN's. Bayesian techniques can be very powerful and should be experimented with further so that the best parameters for them can be found but at first hand it has been found that the best performing NN is the SVM. The HMC Bayesian approach provides the best confidence levels and maybe a combination of these confidence levels with SVM can be attempted in some manner.

REFERENCES

[1] I. T. Nabney, *NETLAB: Algorithms for Pattern Recognition.* London, Great Britain: Springer-Verlag, 2003, pp. 325-365.

[2] J. C. Hull, *Options, Futures and Other Derivatives, 5th Edition.* Upper Saddle River, New Jersey, U.S.A.: Prentice Hall, 2003, pp. 6-10.

[3] F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities," *Journal Political Economy*, vol. 81, pp. 637–659, 1973.

[4] J. C. Hull, *Options, Futures and Other Derivatives, 5th Edition.* Upper Saddle River, New Jersey, U.S.A.: Prentice Hall, 2003, pp. 234-257.

[5] R. A. Jarrow, and S. M. Turnbull, *Derivative Securities, 2nd Edition.* U.S.A.: South-Western College Publishing, 2000, pp. 15-20.

[6] R. A. Jarrow, and S. M. Turnbull, *Derivative Securities, 2nd Edition.* U.S.A.: South-Western College Publishing, 2000, pp. 175-202.

[7] M.M. Pires and T. Marwala, "American Option Pricing Using Multi-Layer Perceptron and Support Vector Machine", in *Proc. IEEE Conference on Systems, Man and Cybernetics*, The Hague, October 10-13 2004, to be published.

[8] I. T. Nabney, *NETLAB: Algorithms for Pattern Recognition.* London, Great Britain: Springer-Verlag, 2003, pp. 156-157.

[9] I. T. Nabney, *NETLAB: Algorithms for Pattern Recognition.* London, Great Britain: Springer-Verlag, 2003, pp. 300-307.

[10] T. H. Mirer, *Economic Statistics and Econometric, Third Edition.* U.S.A: Prentice Hall, Inc., 1995, pp. 209-218.

POSTER ABSTRACTS

Appropriate baseline values for HMM-based speech recognition
Etienne Gouws, Kobus Wolvaardt, Neil Kleynhans, Etienne Barnard

A number of issues related to the development of speech recognition systems with Hidden Markov Models (HMMs) are discussed. A set of systematic experiments using the HTK toolkit and the TIMIT database are used to elucidate matters such as the number of mixtures to use for a particular training set size, the utility of various feature sets, the value of triphone modelling, etc. These results suggest guidelines, which will be useful for those who wish to develop speech-recognition systems in new languages.

Comparative study of Hidden Markov Model and Neural Network on Speech recognition portion of the speech translation system between English and Sepedi
Machaba Machaba, Francis Izeze Ndamutsa, and Tshilidzi Marwala

Language statistics in South Africa show a clear need for the translation between English speech and native South African languages. In this paper, approaches of using Hidden Markov Model (HMM) and Neural Network for implementing the speech recognition portion of a computer based speech translation system between English and Sepedi are compared. An isolated-word, single-speaker speech recognition system with a vocabulary size of 10 words was designed and implemented for each of the Neural Network and Hidden Markov Model methods. An accuracy of 94% and 99.33% respectively was achieved.

Evolutionary Optimisation Methods for Template Based Image Registration
Lukasz A Machowski, Tshilidzi Marwala

This paper investigates the use of evolutionary optimisation techniques to register a template with a scene image. An error function is created to measure the correspondence of the template to the image. The problem presented here is to optimise the horizontal, vertical and scaling parameters that register the template with the scene. The Genetic Algorithm, Simulated Annealing and Particle Swarm Optimisations are compared to a Nelder-Mead Simplex optimisation with starting points chosen in a pre-processing stage. The paper investigates the precision and accuracy of each method and shows that all four methods perform favourably for image registration. SA is the most precise, GA is the most accurate. PSO is a good mix of both and the Simplex method returns local minima the most. A pre-processing stage should be investigated for the evolutionary methods in order to improve performance. Discrete versions of the optimisation methods should be investigated to further improve computational performance.

Land Cover Mapping: Exploring Support Vector Machines
Gidudu Anthony and Heinz Ruther

Remote Sensing is a major source of land cover information due to its ability to acquire measurements of land surfaces at various spatial and temporal scales. The process of identifying land cover on a satellite image is refereed to as image classification. This research investigates the application of Support Vector Machines (SVM), heralded as one of the state of the art classifiers in machine vision, to satellite image classification. The ongoing research indicates that SVMs present a viable approach to the extraction of surface cover information from satellite images. Although linear SVMs are known to be inferior to polynomial and radial basis function SVMs, they do seem to perform surprisingly well when the classification results are evaluated on a class-by-class basis.

Verification Procedures in a Medical Imaging Application
Neil Muller, Evan de Kock, Ruby van Rooyen

At iThemba Labs, we are constructing a patient positioning system for proton therapy. This system will use stereo vision techniques to accurately position the patient relative to the beam line. To do this, we need to be sure that the vision system is functioning correctly. In addition to various procedural checks, we implement a variety of software tests to try and detect errors in the vision system.

Optimising input windows for the Prediction of stock-market indices
B. Leke Betechuoh, T. Marwala

Recent computational intelligence methods such as neural networks have been applied in the field of prediction. In this paper, we propose methods that optimally select the time-window required for the prediction of future stock prices. These methods are implemented in two ways: (1) employing a polynomial approximation to compute the required optimal input time-window; and (2) reformulating the Bayesian neural network architectures to discretely select the optimal input time-window. The results from both the polynomial approximation and Bayesian framework show that 7 days are needed to predict the index average for the next five days.

An Implementation of a Isizulu Text to Speech System
Julia M Majola, Daniel J Mashao

IsiZulu language is the most spoken home language in South Africa. Like many other aspects of life in third world, it lags in its technology interaction. This paper discusses the implementation of an isiZulu concatenative text to speech (TTS) system. There are several kinds of concatenative TTS systems, ranging from the natural sounding full word systems to more flexible phone based general-purpose systems that unfortunately sound like a computer. In this paper, we discuss an implementation of a general-purpose diphone based isiZulu TTS system.

Texture Detection for Eyelash Segmentation in Iris Images
Asheer K. Bachoo , Jules R. Tapamo

The idea of using the distinct spatial distribution of patterns in the human iris for person authentication is now a widely developing technology. Current systems rely on a set of basic assumptions in order to improve the accuracy and running time of the recognition process. The advent of a robust system implies a viable solution to a number of general problems. This paper focuses on a common yet dif_cult problem - the segmentation of eyelashes from iris texture. Tests give promising results when using co-occurrence matrix approach.

Towards Implementing a Text-to-Speech System for Cellular Phones for Blind Users
Lehlohonolo Mohasi, Daniel Mashao

Use of cellular phones has become popular in our everyday lives. This popularity has been brought on by the increasing standard in communication technology, speech technology being part of this. In this paper, we aim to show how speech technology can be used to improve productivity in the use of cellular phones. Text-to-speech (TTS), as an element of speech technology, can be used on cellular phones to enhance communication for blind users. Thus, for our project, we will be implementing TTS on cellular phones. It must be noted that as the project is in its infancy, no results are available as yet.

Financial Forecasting using Conventional and Bayesian trained Neural Networks
Trevor Malcolm Ransome, Kam Hay Claren Chan, Tshilidzi Marwala

This paper presents and compares two neural network (NN) architectures that can be used for financial forecasting, namely the multi-layer perceptron (MLP) and the radial basis function (RBF). These networks are trained using conventional and Bayesian training methods. The testing of the architectures performance was done by trying to predict the future five day Dow-Jones Industrial Index average given the previous seven day closing values. It was found that the RBF network provided the best general solution in the least amount of time. However, if time is not an issue, the hybrid Monte Carlo trained Bayesian neural network having a threshold value of 0.5 provides a better solution compared to the RBF network with respect to the number of trends predicted correctly.

A framework for bootstrapping morphological decomposition
L. Joubert, V. Zimu, M. Davel and E. Barnard

The need for a bootstrapping approach to the morphological decomposition of words in agglutinative languages such as isiZulu is motivated, and the complexities of such an approach are described. We then introduce a generic framework which can be employed for this task, and show a number of simple examples of its use for the decomposition of words in isiZulu. Initial thoughts on the process of rule induction are discussed.

Digital Watermarking for Tamper Detection
Jeremy Thurgood and Roger Peplow

This paper describes digital watermarking for use in authentication and tamper detection and presents a work in progress semi-fragile watermarking algorithm embedded in the wavelet domain. The embedded watermark is generated from contentbased image parameters and is robust to JPEG and SPIHT lossy compression methods while remaining fragile to malicious image manipulations.

Evaluation of speaker adaptation algorithms
Ofentse Noah and Daniel Mashao

This paper aims to evaluate the performance of two speaker adaptation algorithms for speech recognition systems. While Speaker Independent (SI) systems perform well, Speaker Dependent (SD) systems have desirable speaker-specific features that make them perform better than SI systems. It is therefore desirable to have an SD system for a speaker. The algorithms require different amounts of data in order to achieve adaptation on speech recognition systems. One algorithm uses transformation, while the other uses a weighting formula in order to adapt the speech recognition systems. In the evaluation we shall build several speech recognition systems. We shall test these speech recognition systems using various subjects. The main aim of this evaluation is to compare the time it takes the systems to adjust to the training data for each algorithm.

Formal specification of extraction of spatio-temporal semantics in automated surveillance and traffic monitoring
Johan K¨ohler , Jules R. Tapamo

The extraction of semantics from a sequence of images relies on a suitable conceptual model of the objects in the scenes in order to bridge the gap between low-level image features and highlevel descriptions. In this paper we outline a framework for extraction of semantics from traffic and surveillance images using a region-based representation of moving and stationary objects in a scene incorporating context layers to capture semantics. We express some situations in the traffic and surveillance domains to illustrate the potential of the framework.

Overview of MPEG-7 - the Multimedia Content Description Interface
JS van der Merwe, HC Ferreira, WA Clarke

This paper provides an overview of the MPEG-7 standard and also serves as a guide to anyone considering implementing the standard. It explores the different parts that make up the standard and provide readers from different multimedia processing backgrounds a glimpse of the possibilities that may lead to future research or further implementation of current research. There is also a discussion of some of the challenges that may be faced in implementing the standard.

Stock market prediction using evolutionary neural networks
Taryn Tim, Mutajogire Mukono, Nkamankeng Nkamngang, Tshilidzi Marwala

A method based on neural networks and genetic algorithms is proposed to accurately predict the average weekly stock market returns of the Dow Jones index. The relationship between the input and output variables is modeled using a multilayer perceptron (MLP) that is trained using the scaled conjugate gradient method. A genetic algorithm is used to select the optimum number of past day's closing prices as inputs, as well as the MLP architecture. The accuracy in the buy or sell decisions based on this network ranges between 78 and 80%. The trading strategy using the neural network yielded profits at least 5 times greater than that of the simple buy and hold strategy.

Face recognition using eigenfaces and the CrCb colour space
Neil Muller

Most research on face recognition has focused on using the grey-scale intensity images for recognition. This is sensitive to changing lighting conditions. Colour is increasingly being used as a method for overcoming this sensitivity to lighting. In this paper, we investigate a simplistic method of using purely chromatic information for recognition and evaluate its performance relative to the standard eigenface technique on the XM2VTS database.

VTIMIT: The Vodacom speech corpus
Daniel J. Mashao and Nicholas Zulu

This paper describes the creation of the cellular speech database based on the TIMIT database. Speech databases play an important role in research on speech technology. They enable researchers to compare results of different algorithms and implementations. Due to the popularity of mobile platforms databases that are created on these networks are increasingly important for design of speech applications. A cellular CTIMIT database was created in response to this need. The CTIMIT database, has however, two major weaknesses; it is not complete and it was created over an analogmobile network that is no longer in widespread use. In this paper we present a new speech database that is complete (in TIMIT sense) and based on the worldsmost popular cellular coding technology. We give results of comparing a speaker identification task on the new database and other TIMIT derived databases. The results show that is still noisier than the TIMIT database but also easier to recognise speakers than on the NTIMIT database. This could be because no additional noises were added other than the GSM network.

Formulation of a Hidden Markov Model to Learn The Motion Patterns in People's Day to Day Activities

Lynn Sitzer, Fred Nicolls and Gerhard De Jager

The ability to detect irregular patterns in people's behaviour, or detecting that a particular person's behaviour does not match up to a certain predefined model, is very useful in surveillance applications. An aerial view of a person working at their desk is captured from an uncalibrated camera. The actual state of the person is unknown and we will use Hidden Markov Models to model the scenario. Segmentation of the captured video sequence is achieved by subtraction of a generic background image. The features extracted from the image data are chosen so that they differ from person to person due to the variability in motion and physical appearance. A HMM will be trained up for a particular person's motion.

The contour tracking of a rugby ball: An application of particle filtering

Tersia Janse van Rensburg, M.A. van Wyk, Marco van der Schyff, Johan Smit

This article discusses the principles of particle filtering and contour tracking as a method to track an object in an image sequence. The tracking of a rugby ball, as a practical example for applying these techniques, is discussed.

Three-dimensional finite difference time domain modelling of borehole radar in mining applications

P. K. Mukhopadhyay, M.R. Inggs and A.J. Wilkinson

Imaging subsurface orebodies in three-dimension (3-D) with high resolution (_ 1m) is of high importance in the mining industry. A borehole radar is an electromagnetic (EM) tool for detecting electrical discontinuities in the rock formation with high resolution. often the host rock medium through which the wave propagates and the target orebody have complex electromagnetic properties. Modelling EM wave propagation in 3-D is important for understanding the physics of observed responses, and for providing insight into data processing and interpretation. The finite difference time domain (FDTD) method is a useful numerical method suitable for modelling EM wave propagation through complex media. A 3-D FDTD code in a Cartesian coordinate system has been written and implemented by using uniaxial perfectly matched layer (UPML) for boundary absorber. This code has been used to simulate the EM field responses of a dipole source in a lossy medium. The code has been applied to some subsurface problems, such as layered sediments EM wave propagation and reflection from a geological reverse fault. Near surface sediments often form layered structures. When the wavelength of the propagating pulse is similar to the thickness of the host layer, crosswell radar traces start propagating as a *guided wave.* A borehole has been included in the simulation and the borehole mud has been modelled as wet clay. The borehole mud and the size of the borehole causes a significant effect both on the amplitude and signature of the radar traces.