Person Tracking in 3D Using Kalman Filtering in Single and Multiple Camera Environments

Bruno Merven

A dissertation submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfilment of the requirements
for the degree of Master of Science in Engineering.

Cape Town, August 2004

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author.... Signed by candidate

Cape Town
20 August 2004

Abstract

We present a multi-camera person tracker solution that makes use of Kalman filtering principles. The tracking system could be used in conjunction with behaviour analysis systems to perform automated monitoring of human activity in a range of different environments. Targets are tracked in a 3-D world-view coordinate system which is common to all cameras monitoring the scene. Targets are modelled as ellipsoids and their colour information is parameterised by RGB-height histograms. Observations used to update the target models are generated by matching the targets in the different views.

3-D tracking requires that cameras are calibrated to the world coordinate system. We investigate some practical methods of obtaining this calibration information without laying out and measuring calibration markers. Both tracking and calibration methods were tested extensively using 6 different single and multiple camera test sequences. The system is able to initiate, maintain and terminate the tracks of several people in cluttered scenes. However, further optimisation of the algorithm is required to achieve tracking in real time.

Acknowledgements

I would like to thank the members in the Digital Image Processing Group at UCT. Thanks to Keith, Markus, Mathew, and Prof. de Jager, and also particularly to Dr. Fred Nicolls, who provided numerous suggestions on my problem. I would also like to give thanks to the members of the Centrum för Bildanalys at the University of Uppsala, Sweden for support and input during the time I spent with them. This research would not have been possible without the financial support given by the National Research Foundation (NRF), and by DeBeers Technology Group (DebTech), to whom I'm grateful. Finally, big thanks to Zia, my house mates Fred NC, Kevin, Bruce and Marc and also my parents for their continuous support and encouragement throughout the writing process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automated visual monitoring systems may be used for a very wide range of applications. Cameras are cheap and versatile and the information content in a video sequence is very high. The main application of visual monitoring is surveillance but more general measurement of human activity such as customer behaviour analysis in shopping malls, perceptual interfaces in intelligent homes and team strategy in sports are other possibilities. An illustration of an automated visual monitoring system is given in figure 1.1. As shown in the figure, the tasks to be performed by such a system can be divided into 'low-level' tasks, which include detection, tracking and camera calibration and 'high-level' tasks, which include behaviour recognition, face recognition and archiving of this high-level analysis. In this thesis we will address only the 'low-level' tasks. The system we present could then be used in conjunction with a 'high-level' system such as one developed by Forbes [10] for the purpose of automated visual person monitoring. In this chapter we define our problem statement in the context of previous work found in the literature and we introduce our chosen approach thus giving a high-level overview of the rest of the thesis.

1.1 Problem definition

The basic requirement for a person tracker for a particular scene monitored by one or several cameras is to be able to detect every person entering the scene and keep track of each of them until they all leave. This task, although trivial for the human eye, is very hard to

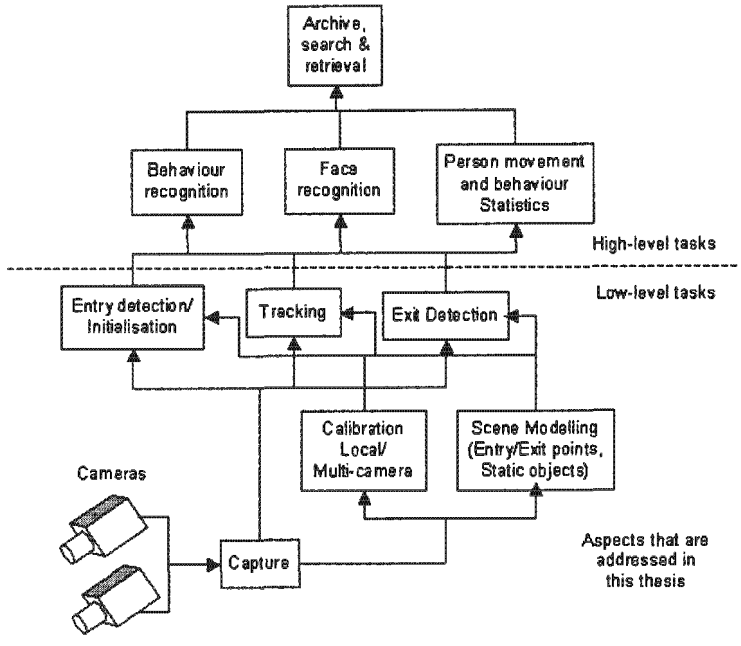


Figure 1.1: The figure shown here shows a full person tracking solution in the context of surveillance or person activity measurement.

automate due to the presence of complexities such as shadows, reflections, changing lighting conditions and occlusions resulting from the interaction of people, static and moving objects. Additional complexities arise in the case of multiple camera configurations where track information has to be shared between different cameras. Tracking of this nature is a highly unconstrained problem. The more a priori information that is incorporated, the more tractable the problem becomes. Two main components can be distinguished in a typical visual tracker. *Filtering*, mostly a top-down process, deals with the dynamics of targets, makes use of scene priors, and evaluates different hypotheses. The other component, *Target Representation* and *Localisation*, is mostly a bottom-up process that has to deal with the changes in the appearance of the target. The way the two components are combined and weighted plays an important role in the robustness of the tracker [7].

1.1.1 Filtering

Filtering almost completely replaces previous rule-based approaches such as ones implemented in [35] simply because they are far more efficient and generally less complex in their implementation. The filtering process is normally formulated through the state space approach for modelling discrete-time dynamic systems [38, 21, 11, 45]. The information characterising the target is defined by the state sequence $\{\mathbf{x}_k\}_{k=0,1,\ldots}$, whose evolution in time is specified by the dynamic equation $\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k)$. The available measurements $\mathbf{y}_k$ are related to the corresponding states by the measurement equation $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{e}_k)$, where $\mathbf{f}_k$ and $\mathbf{h}_k$ are vector-valued, time-varying functions. Each of the noise sequences, $\{\mathbf{v}_k\}_{k=0,1,\ldots}$ and $\{\mathbf{e}_k\}_{k=0,1,\ldots}$ is assumed to be independent and identically distributed (i.i.d.).

The objective of tracking is to estimate the state $\mathbf{x}_k$ given all the measurements $\mathbf{y}_{1:k}$ up to that moment, or equivalently to construct the probability density function (pdf) $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. The theoretical optimal solution is provided by the recursive Bayesian filter which solves the problem in two steps. The *prediction* step uses the dynamic equation and the previously computed pdf of the state at time $t = k-1$ (or initial pdf at $t = 0$) $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ to derive the prior pdf of the current state $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$. Then the *update* step employs the likelihood function $p(\mathbf{y}_k|\mathbf{x}_k)$ of the current measurement to compute the posterior pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.

When the noise sequences are Gaussian and $\mathbf{f}_k$ and $\mathbf{h}_k$ are linear functions, the optimal solution is provided by the Kalman filter ([38], p.142), which results in the posterior also being Gaussian. When the functions $\mathbf{f}_k$ and $\mathbf{h}_k$ are nonlinear, the Extended Kalman Filter (EKF) is obtained by linearisation ([38], p.247). The posterior density in this case is still modelled as Gaussian. An alternative to the EKF is the Unscented Kalman Filter (UKF) [26] which uses a set of discretely sampled points to parameterise the mean and covariance of the posterior density. Kalman filtering was first used for visual tracking by Ayache and Faugeras in 1989 [1] for tracking lines using a camera. Since then various extensions of the filter have shown much success. Zhao and Nevatia [49], Kang and Cohen [20], Comaniciu and Ramesh [7] as well as Piater and Crowley [30], to mention a few, use Kalman filtering for person tracking.

When the state space is discrete and consists of a finite number of states, Hidden Markov Models (HMM) filters [33] can be applied for tracking. This method is implemented by Chen and Rui [6] for visual tracking.

The most general class of filters is represented by particle filters, also called bootstrap filters, which are based on Monte Carlo integration methods. This more general type of filter allows for the state space representation of any distribution and for nonlinear, non Gaussian dynamical and observation models, and process and observations noises. Particle filtering was first introduced in vision as the Condensation algorithm by Isard and Blake [14]. In [28] Nummiaro and Gool present an adaptive colour-based particle filter and compare its performance with a mean-shift tracker and a combination of mean-shift and Kalman filter tracker. Although particle filtering allows for more flexibility it is more difficult to implement. Given a particular tracking problem one has to gauge whether the gained generality is worth the added complexity.

1.1.2 Target Representation and Localisation

The *target representation and localisation* component deals with the measurement process where observations characterised by the pdf $p(\mathbf{y})$, used in the *update* step of the filtering process, are obtained. While *filtering* has its roots in control theory, algorithms for *target representation and localisation* are specific to image processing. For the visual person tracking application targets can be characterised by two main features: their *colour composition* and their *shape and size.*

Targets' Colour Composition

To characterise targets' colour composition, a feature space needs to be chosen. The most common approaches are *colour histograms* [7, 28], *gaussian mixture models* [39] and *appearance models* [49, 34, 27, 15]. Colour histograms are scale and orientation invariant, but lose all spatial information. Gaussian mixture models, like histograms, capture different target characteristics, depending on what features are used, but usually require many parameters to be set (via a training phase) and are complex to implement. In appearance models, target appearance information is stored on a pixel level template, which is then used for matching. Thus appearance models make use of spatial information, but adjusting for scale and orientation changes over time is difficult.

Targets' Shape and Size

Modelling the shape of a non-rigid targets such as humans is not always easy. In many implementations e.g. [35, 20], tracked targets are simply modelled as rectangular bounding boxes in the image view. Another common image view shape model is the ellipse which more accurately accounts for feet and head being narrower parts of the body [28, 7]. Although in many of these implementations, the size of the bounding boxes/ellipses are allowed to vary, it is difficult to accurately explain how they should change. A better alternative is to model targets as 3-D objects. Unless one is trying to recover the exact pose [29, 46] of a tracked person it is not necessary to use a complicated articulated model. Simple shapes such as cylinders [14] or ellipsoids [27] are suitable. To make use of this 3-D information one has to formulate the tracking problem in a 3-D world coordinate system or world-view. Other than explaining how the size and shape of targets in the image varies as they move, a world-view tracker has several additional advantages. It makes it easier to introduce known physical constraints to the dynamic tracking models. Initialisation and termination of tracks can be made more robust if entry/exit points are specified. These points are more easily specified in world coordinates than in image coordinates. 3-D information also greatly simplifies the task of combining measurements obtained from several cameras with overlapping views. However, this approach limits the tracking system to fixed cameras that all have to be calibrated with respect to a common coordinate system. Thus we also address the problem of camera calibration for person tracking applications in this thesis.

Target Localisation

The localisation is performed by comparing target models with image samples to maximise some likelihood (similarity) type function. Comaniciu [7] exploits the smoothness of the similarity function to make use of gradient optimisation to localise targets. Others, like Nummiaro [28], sample images according to the prior distributions of target locations $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and weighs the contribution of each sample according to its likelihood.

The comparison between target models and image samples depends on the chosen target representation. Two methods that were considered are the *Bhattacharyya Coefficient* [18]

and the *Histogram Intersection* method [41]. If $\mathbf{p(y)}$ is the density function of a target candidate at position $\mathbf{y}$ in the image and $\mathbf{q}$ is density of the target model then the measure of distance between the two densities $\rho(\mathbf{y})$ based on the *Bhattacharyya Coefficient* in the chosen feature space $\mathbf{z}$ is as follows:

$$\rho(\mathbf{y}) = \int \sqrt{\mathbf{p_z(y)q_z}} d\mathbf{z}. \tag{1.1}$$

In the histogram formulation the discrete densities $\mathbf{p(y)} = \{p_u(\mathbf{y})\}_{u=1...n}$ and $\mathbf{q} = \{q_u\}_{u=1...n}$ are estimated from the n-bin histograms of the image samples and the target model. The sample estimate of the *Bhattacharyya Coefficient* is then given by:

$$\rho(\mathbf{y}) = \sum_{u=1}^{n} \sqrt{p_u(\mathbf{y})q_u}. \tag{1.2}$$

In the case of the *Histogram Intersection* method, the similarity measure between histograms is given by:

$$\rho(\mathbf{y}) = \sum_{u=1}^{n} \min(p_u(\mathbf{y}), q_u). \tag{1.3}$$

The strength of the *Histogram Intersection* results from the min(...) function, which makes sure that only colours present in the model histogram are matched. The *Bhattacharyya Coefficient* on the other hand has a stronger theoretical foundation, being linked to the Bayes error. It also imposes a metric structure on the distance measure between histograms.

Foreground/Background Segmentation

Foreground/background is typically done by comparing new images as they arrive, to some background or reference model. Images are segmented into foreground and background regions and higher weighting is given to foreground pixels in the image sampling process. The segmentation can be simply performed by taking the difference between sequence images and some reference image or background model [30]. More elaborate methods for obtaining foreground regions are found in [47], where each pixel is modelled as an independent Gaussian mixture model, and in [2], where segmentation is achieved using spatial gradient information. Difficulties arise in the presence of shadows and reflections, moving objects in the background, and varying lighting conditions. Thus the implementation of a robust tracker that relies purely on segmentation information is very difficult.

1.1.3 Calibration methods suited to person tracking applications

Camera calibration in the context of machine vision is the process of determining the internal camera geometric and optical characteristics defined by the intrinsic parameters, and the camera pose (position and orientation) within a world coordinate system, defined by extrinsic parameters. Standard calibration methods based on methods by Tsai [43, 42] are accurate but require the use of calibration points or calibration objects. Calibration points/markers have to be laid out and measured, a process which requires a lot of care. Although methods that make use of calibration objects are suitable for obtaining internal camera parameters, they are usually not for obtaining the camera pose in large fields of view. In the case of a surveillance system covering an entire building where dozens (hundreds) of cameras are installed the use of such calibration methods is a sizeable task which renders a world-view tracker impractical. Auto calibration methods aim to obtain camera parameters without the need for manual procedures or calibration objects, and hence are more suited to person tracking applications.

Jones et al [16] propose a two-stage method to recover calibration parameters for multi-camera configurations automatically. In the first stage, each camera is calibrated to a local ground plane coordinate system. The algorithm makes use of how the size of the segmented images of people in the camera view vary as they walk towards or away from the camera to recover the pitch angle and the focal length to pixel width ratio of the camera, provided the camera height above the ground is known. However, this method assumes shallow camera pitch angle, small roll and pan angles, ignores distortion effects, relies on good segmentation and requires some control over what goes on in the scene during the calibration process. Hence, it is not suited to all camera configurations and video sequences, and alternative semi-automatic methods have to be considered. The second stage of Jones et al.'s method recovers the transformation between the local ground plane coordinate systems by matching tracks obtained in each camera view. This part of the algorithm relies on good local calibration (obtained in the first stage), on a reasonably good monocular world view tracker and also on the different views overlapping. In cases where these conditions cannot be met, semi-automated or manual alternatives have to be considered.

1.2 Overview of Proposed Method

Having presented the various different approaches to the person tracking problem we introduce the approach we have adopted and present in this thesis. Note that this is only an overview; the notions are formally and completely presented in the chapters ahead.

1.2.1 Filtering aspects

We forego the flexibility of particle filtering by assuming simple Gaussian noise sequences, thus adopting the Kalman filter formulation. We model each target as a separate linear model formulated in a world view. The state vector $\mathbf{x}(t)$ follows a transition relationship of the form

$$\mathbf{x}(t) = \mathbf{F}(\Delta t)\mathbf{x}(t - \Delta t) + |\Delta t|\mathbf{v}(t)^1. \tag{1.4}$$

This formulation allows for asynchronous updates of the model. We elaborate further on this choice of formulation in chapter 3.

The observations or measurements are made in the image view. Under perspective projection this measurement process is non-linear. This breaches one of the assumptions of conventional Kalman filtering. We thus adjust for this by performing local linearisation of the measurement process, which results in the Extended Kalman Filter formulation.

1.2.2 Target representation and localisation

For shape representation we model each target as a 3-D ellipsoid with a vertical major axis and feet on the ground plane. To explain the shape and the size of the targets in the image, a projection of the ellipsoid to the image plane can be computed. Under perspective projection, the image of an ellipsoid is actually an ellipse in the image plane.

For colour representation we implement a novel compromise between the colour histogram and the appearance model: a RGB-height histogram. This formulation has the advantage of being size invariant whilst still retaining some spatial information. The *RGB* colour

[1]The temporal indexing notation using t replaces the one using k from the previous section throughout the rest of the thesis. The two notations are related by $t = k\Delta t$.

space was chosen simply because raw image data is in *RGB*, and although slightly better representation (with regards to varying lighting conditions for example) is achieved using *HSV* and *L*a*b* colour spaces, the incurred computational costs in the conversion (from *RGB* to *HSV*/*L*a*b*) is not justified. At initialisation or during the matching process the histogram is populated only by pixels in the foreground regions masked by the expected target position, shape and size in the image (defined by the projected ellipsoid). Foreground regions are obtained using background subtraction in RGB space.

The matching process is performed by sampling the image according to the prior distribution $p(\mathbf{x}(t|t-\Delta t))$ and comparing these samples to a reference target colour model. We found that slightly better performance was achieved when using *Bhattacharyya Coefficient* approach rather than the *Histogram Intersection* approach for histograms comparisons. The best matched sample is then used to define the measurement pdf $p(\mathbf{y})$.

1.2.3 Tracking with multiple cameras

As stated earlier, a world-view formulation of the tracking problem facilitates the task of combining measurements from multiple cameras. Figure 1.2 gives an overview of the system for multiple camera configurations. Each camera view is associated with a different *tracking client*. The world view model in which the tracking takes places exists in a world coordinate system which is independent of different camera views. Cameras are calibrated to this world view so the transformation from world view to image view is always known. Each time a new image is captured and made available to a client foreground regions are identified/segmented using a reference background model. The client then fetches a description of the current targets from the world-view *Server* and the predicted or prior distribution $p(\mathbf{x}(t|t-\Delta t))$ is calculated. The client then tries to match the segmented image data to the targets in the scene. The target *representation and localisation* introduced in the previous paragraph determines how this matching process is performed. At the current stage of implementation each client maintains its own colour model of each target. Finally the client returns to the server the observation obtained from the image and the updated or posterior distribution $p(\mathbf{x}(t|t))$ is calculated.

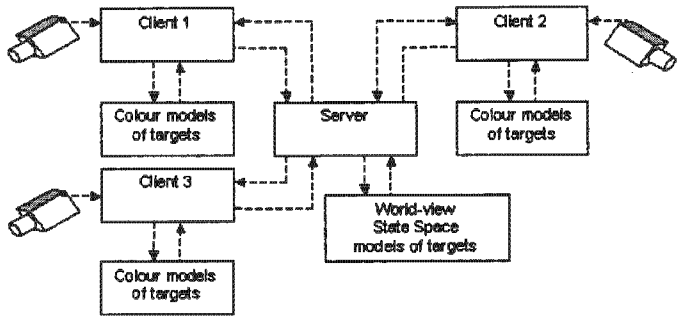


Figure 1.2: Client Server Configuration.

1.3 Datasets

The current implementation of the proposed tracker is too slow to track in real time, so it is tested and evaluated using prerecorded video sequences. The datasets chosen cover a wide range of different camera configurations in an attempt to show the generic nature of the proposed method. Each has its own particular difficulty with regards to both tracking and calibration aspects. The sequences are:

1. The 2-Cam Debtech sequence
2. The 4-Cam DIP sequence
3. The 1-Cam Jammie sequence
4. The Colourful People sequence
5. The PETS2002 sequence
6. The PETS2004 sequence.

1.3.1 2-Cam Debtech Sequence

This dataset is an indoor sequence taken using a set of 2 near-horizontal cameras with overlapping views. Images from both cameras were recorded synchronously at a fixed frame rate. Although the sequence only contains one person, tracking difficulties arise

Figure 1.3: The 2-Cam Debtech sequence.

from partial and complete occlusions occurring in at least one of the views at a time. The calibration of this particular camera configuration can be done using all the methods dealt with in the thesis. Figure 1.3 shows views from each of the cameras used.

1.3.2 The 4-Cam DIP sequence

This dataset is an indoor sequence taken using four ceiling cameras pointing straight down with partially overlapping views. The images from each of the cameras were received asynchronously, each with a time-stamp. The sequence contains three targets and difficulties arise from the numerous occlusions that occur. Another difficulty arises from the fact that the images received are quite severely radially distorted. Since calibration points were available, Tsai's method was used in this case to calibrate the cameras. Figure 1.4 shows shots from each of the cameras used.

1.3.3 The 1-Cam Jammie sequence

This dataset is an outdoor sequence taken using one near-horizontal camera. The difficulty of this sequence arises from the fact that the three tracked people are of very similar colour composition as illustrated in figure 1.5. Calibration for this sequence was performed using Tsai's method for coplanar calibration points and an automatic method proposed in this

Figure 1.4: The 4-Cam DIP sequence

thesis.

1.3.4 The Colourful people sequence

This dataset is an indoor sequence taken using one near horizontal camera. The difficulty of this sequence results from the numerous occlusions that occur from the interaction of the seven people present in the scene at the same time, despite the fact that they are highly colourised. Figure 1.5 shows a particular frame taken from the sequence when all seven people are present in the scene.

1.3.5 The PETS2002 sequence

This dataset was recorded and made available as a standard dataset for the Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS) in 2002. It is a sequence taken with a low quality camera in a shopping mall environment.

1-Cam Jammie Sequence	
Cameras	1
Total tracks	3
Total frames	218
Recorded fps	10
Frame size	360 × 288

Colourful people Sequence	
Cameras	1
Total tracks	7
Total frames	800
Recorded fps	25
Frame size	384 × 288

Figure 1.5: The 1-Cam Jammie and Colourful People sequences.

PETS 2002 Sequence	
Cameras	1
Total tracks	9
Total frames	1050
Recorded fps	15
Frame size	352 x 240

PETS 2004 Sequence	
Cameras	1
Total tracks	4
Total frames	505
Recorded fps	15
Frame size	352 × 240

Figure 1.6: The 2002 and 2004 PETS sequences.

This is probably the most difficult dataset used in this thesis. The main difficulties arise from the poorly defined entry and exit points, the poor image quality and the similitude of the colour composition of people in the scene. Calibration had to be performed manually for this sequence. Figure 1.6 shows a particular frame taken from this sequence.

1.3.6 The PETS2004 sequence

This last dataset was recorded and made available for the PETS 2004 workshop. It was taken from a ceiling camera placed quite high above the ground. The difficulties here arise from the small size of target images and the presence of a patch of sunlight in the middle of

the scene that drastically affects the colour composition of the targets as they pass through it, as shown in figure 1.6.

1.4 Thesis organisation and outline

Chapter 2 describes the measurement process. It deals with our chosen method of target representation for both colour and shape aspects. We explain how 3-D target shape model is projected to its corresponding 2-D image. We describe how the image samples are obtained and compared with the reference colour models.

In chapter 3 we give details of our person tracking algorithm. We describe how tracks are initiated and terminated and we list the assumptions made by the tracking system.

Since the tracking system presented in this thesis relies significantly on the calibration of the different cameras to a common coordinate system we address the problem of camera calibration in chapter 4. We describe the two-stage automatic method based on one by Jones et al. We also suggest but do not discuss in detail some other calibration methods for camera configuration that cannot be calibrated using the automatic method.

In chapter 5, we give an evaluation of the performance of the tracking method on each of the chosen datasets. We detail how the complexity of each of the datasets is computed. The evaluation is performed by comparing tracks estimated using the proposed system to tracks that were generated by hand.

We conclude the main body of the thesis with chapter 6 where we discuss findings and propose some possible extensions the methods presented in this thesis.

In appendix A, we specify the various parameters that we used in the evaluation of the tracking system.

Appendix B describes Tsai's camera calibration method.

Chapter 2

The Measurement Process

This chapter describes the observation or measurement process of obtaining $p(\mathbf{y}(t))$ where $\mathbf{y}(t)$ is related to the state $\mathbf{x}(t)$ by

$$\mathbf{y}(t) = \mathbf{h}_c(\mathbf{x}(t)) + \mathbf{e}(t), \tag{2.1}$$

and where $\mathbf{h}_c$ is the mapping from the state vector (target's position in the world) to the measurement vector (target's position in the image) for camera c and $\mathbf{e}(t)$ is a Gaussian noise. This process has two aspects that need to be addressed. The first deals with the choice of a feature space to characterise targets. For each target a reference *target model* is represented in the chosen feature space. Consider several *target candidates* also represented in the same feature space and obtained from different parts of an image. By computing the similarity between the target model and the selected target candidates (samples from the image) we can deduce the most likely position of the target in the image. The next problem is to decide how to select those candidates from the image (position, size and shape). From a computational cost point of view, an exhaustive search of the whole image is too expensive to be considered. By using the prior distribution $p(\hat{\mathbf{x}}(t|t-\Delta t))$ (constraint on position) and by assuming that targets are ellipsoidal in 3-D space (constraint on shape and size) we show how only a few well picked samples are necessary to accurately locate targets.

We start by introducing our chosen feature space for target representation. The next topic we discuss is how correspondences between the world-view and the image view is done with

a brief introduction to perspective projection. We explain how the world-view model of a target (the ellipsoid) is generated and how we can use perspective projection and the calibration information to compute the corresponding shape of targets in a given camera view. Finally we describe the process of obtaining a measurement from the image using our chosen target representation.

2.1 Target Representation: Colour

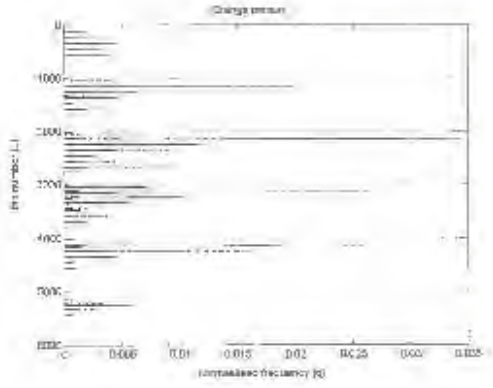
As mentioned in chapter 1, the feature space chosen to represent targets is a compromise between histograms and appearance models. We bin colour information to a $n_R \times n_G \times n_B \times n_z$ histogram where n_R, n_G and n_B are the number of bins for Red Green and Blue values.

The height dimension of the image of targets is discretised into n_z bins. Using this 4-dimensional histogram is effectively the same as modelling targets using n_z ordered $n_R \times n_G \times n_B$ colour histograms, enabling us to make use of some spatial information while retaining the advantages of using histograms. We thus define the discrete pdf's of the target model and a target candidate at position $\mathbf{y}$ as

$$\begin{aligned} \text{target model:} \quad & \mathbf{q} = \{q_u\}_{u=1\ldots n} & & \sum_{u=1}^{n} q_u = 1, \\ \text{target candidate:} \quad & \mathbf{p}(\mathbf{y}) = \{p_u(\mathbf{y})\}_{u=1\ldots n} & & \sum_{u=1}^{n} p_u = 1, \end{aligned}$$

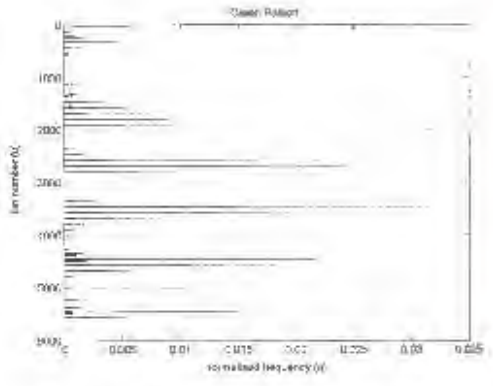
where $n = n_R \times n_G \times n_B \times n_z$. Figure 2.1(a) shows a frame from the *Colourful people* sequence. The histograms (rotated on their side) representation of two of the targets (orange and green person) are shown for $n_R = n_G = n_B = 10$ and $n_z = 6$ in figures 2.1(b) and 2.1(d). The colour of each band in the colour charts shown in figures 2.1(c) and 2.1(e), is the mean of the RGB pdf for each of the height partitions.

(a) Colourful People sequence - frame 125



(b) Histogram for orange person

(c) Colour chart for orange person



(d) Histogram for green person

(e) Colour chart for green person

Figure 2.1: Colour model summary for 4 tracked subjects. Each of the 6 'bands' are coloured with the mean of the colour histogram associated with the corresponding height partition.

2.2 Target Representation: Shape and Size

Before we can discuss how our chosen 3-D world view representation of the targets is projected into an image view we need to introduce a few notions dealing with perspective projection and camera calibration. Good references on this topic include books and notes by Pollefeys [31], Hartley and Zisserman [12] and Birchfield [3].

2.2.1 From the World View to Image View

The world to image transformation function is a non-linear function parameterised by 12 scalars (calibration parameters), 6 extrinsic and 6 intrinsic. These can be usefully combined to form 3 parameters $[\mathbf{R} \quad \mathbf{t}]$ (perspective projection matrix), $\mathbf{S}$ (intrinsic matrix) and κ (radial distortion parameters).

The Perspective Projection Matrix

$[\mathbf{R} \quad \mathbf{t}]$ is a 3×4 matrix known as the perspective projection matrix[1]. The matrix $[\mathbf{R} \quad \mathbf{t}]$ is also referred to as the extrinsic matrix because it holds the camera's extrinsic parameters. These describe the camera pose within the predefined world coordinate system. In other words, they relate the camera reference frame to the world reference frame. $[\mathbf{R} \quad \mathbf{t}]$ is made up of a 3×3 rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$. The matrix $\mathbf{R}$ is itself constructed using the pitch, yaw and roll angles of the camera. The vector $-\mathbf{R}^T\mathbf{t}$ gives the position of the camera in the world reference frame.

The S Matrix

The 3×3 matrix $\mathbf{S}$ describes an affine transformation that scales camera-centred points (in world units) to image points (in image units). It is known as the intrinsic matrix because it holds some of the camera internal parameters, namely

[1] Perspective projection maps a point in $\mathcal{P}^3$ to a point in $\mathcal{P}^2$. A point in projective space of n-dimensions, $\mathcal{P}^n$ is represented by an augmented $(n+1)$ vector of coordinates.

- The lens focal length f.
- The horizontal and vertical pixel dimensions or inter-pixel widths α_j, α_i of the capture element or CCD[2].
- The row and column image centre coordinates (i_0, j_0).
- The skewness of the two image axes, denoted by c.

S is given in terms of these parameters as

$$\mathbf{S} = \begin{bmatrix} f_j^o & c & j_0 \\ 0 & f_i^o & i_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $f_j^o = \frac{f}{\alpha_j}$, $f_i^o = \frac{f}{\alpha_i}$ and c is assumed to be 0.

The parameter $\mathbf{P} = \mathbf{S}\,[\mathbf{R} \;\; \mathbf{t}]$ describes the transformation from the world coordinate system to the undistorted image plane. This is a homogenous transformation so

$$\mathbf{Y}_u = \mathbf{P}\mathbf{X}, \tag{2.2}$$

where $\mathbf{X}$ is the augmented 3-D world point $(X_w, Y_w, Z_w, 1)$ and $\mathbf{Y}_u$ is the augmented 2-D projected image point (X_c, Y_c, Z_c). The undistorted image coordinates $\mathbf{y}_u = (j_u, i_u)$ are obtained by

$$\mathbf{y}_u = (\frac{X_c}{Z_c}, \frac{Y_c}{Z_c}). \tag{2.3}$$

Lens Radial Distortion

There are two types of radial distortion: pincushion and barrel. In the first case, the further a point is from the centre of the image, the more it is distorted away from the centre of the image. In the second case, the opposite happens: the further the point is from the centre of the image, the more it is distorted towards the centre of the image. Barrel distortion is more common than pincushion distortion.

[2]Throughout this thesis we denote row and column pixel coordinates by i and j.

Barrel radial distortion can be modelled as follows:

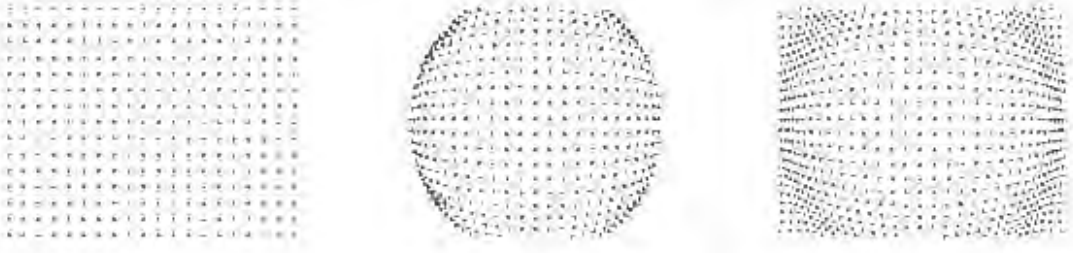
$$x = i_u \left(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + ...\right) \tag{2.4}$$

$$y = j_u \left(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + ...\right) \tag{2.5}$$

where

$$r_u^2 = i_u^2 + j_u^2. \tag{2.6}$$

(i, j) are distorted image coordinates, (i_u, j_u) are undistorted image coordinates and $\kappa_1, \kappa_2, ...$ are the distortion coefficients of the lens. For barrel distortion κ_1 is negative. For most applications it is sufficient to model distortion only with the 1^{st} order distortion coefficient κ_1. There are instances, particularly when important information is contained in the extreme corners of an image with high distortion, when it is necessary to include the 2^{nd} order distortion terms as well. Figure 2.2 illustrates barrel distortion when it is modelled using only κ_1, and then using κ_1 (negative value) and κ_2 (positive value).



(a) Undistorted (b) 1^{st} order distortion (c) 1^{st} and 2^{nd} order distortion

Figure 2.2: Illustration of lens distortion

Note that Tsai [43], amongst others ([31, 48]), models distortion as the inverse of the functions laid out in (2.4) and (2.5). If Tsai's approach is used κ_1 is positive. For our application, Tsai's approach is more computationally expensive hence we adopt the formulation given in 2.4 and 2.5.

2.2.2 The Ellipsoid

An ellipsoid is a second order surface that belongs to a family surfaces referred to as *quadrics*. In $\mathcal{P}^n$, a quadric can be represented by a $(n + 1) \times (n + 1)$ matrix $\mathbf{Q}$ such that

all the points that are elements of the quadric will satisfy:

$$\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0 \tag{2.7}$$

where $\mathbf{X}$ is a $(n+1)$ vector

In the case where $n = 2$, quadrics are called *conics*. Ellipses, parabolas and hyperbolas are referred as conics in projective geometry. A useful property of a quadric such as the ellipsoid is that it forms a 2-D conic under perspective projection transformations. Using ellipsoids to model the shape and size of targets is thus convenient since it results in elliptical person models in the image plane. In the 3-D world view we define an ellipsoid in terms of a centroid, size and orientation. Since we are always going to assume that the tracked subject is a person standing or walking, the orientation is assumed to be vertical at all times. The quadric $\mathbf{Q}$ used to represent such an ellipse size $r_x \times r_y \times r_z$ with centroid at (X_Q, Y_Q, Z_Q) shown in figure 2.3, is constructed as follows :

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{r_x^2} & 0 & 0 & \frac{-X_Q}{r_x^2} \\ 0 & \frac{1}{r_y^2} & 0 & \frac{-Y_Q}{r_y^2} \\ 0 & 0 & \frac{1}{r_z^2} & \frac{-Z_Q}{r_z^2} \\ \frac{-X_Q}{r_x^2} & \frac{-Y_Q}{r_y^2} & \frac{-Z_Q}{r_z^2} & \frac{X_Q^2}{r_x^2} + \frac{Y_Q^2}{r_y^2} + \frac{Z_Q^2}{r_z^2} \end{bmatrix}. \tag{2.8}$$

Figure 2.3: Ellipsoid used to model shape of person

2.2.3 From an Ellipsoid to an Ellipse

An ellipsoid is a particular configuration of a quadric represented in homogeneous coordinates by a symmetric 4×4 matrix $\mathbf{Q}$. The points in space that are inside the ellipsoid satisfy

$$\mathbf{X}^T\mathbf{Q}\mathbf{X} > 0, \tag{2.9}$$

where $\mathbf{X} = (X, Y, Z, 1)^T$ is the 3-D homogeneous coordinates of points in the world view.

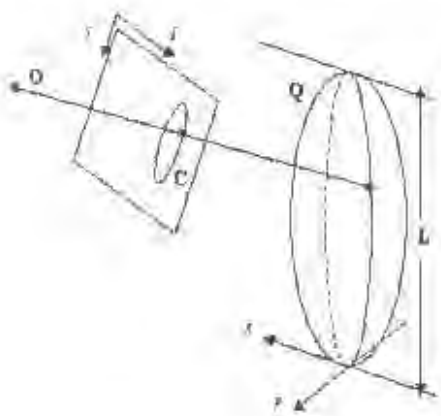


Figure 2.4: A quadric Q with its projection C on the image plane

It is shown in [40] that for a normalised projective camera $\mathbf{P}_o = \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix}$, the profile of a quadric

$$\mathbf{Q}_o = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c \end{pmatrix}$$

is a conic $\mathbf{C}$ described by

$$\mathbf{C} = c\mathbf{A} - \mathbf{b}\mathbf{b}^T. \tag{2.10}$$

Hence the points $\mathbf{Y}$ in the image that lie inside the projected ellipse satisfy

$$\mathbf{Y}^T\mathbf{C}\mathbf{Y} > 0, \tag{2.11}$$

where $\mathbf{Y}$ is the homogeneous undistorted pixel coordinates of points in the image space. To obtain the image $\mathbf{Q}_i$ of a quadric $\mathbf{Q}_w$ in an arbitrary projective camera $\mathbf{P} = \mathbf{S}[\mathbf{R} \quad \mathbf{t}]$,

one has to first compute $\mathcal{H}$ such that $\mathbf{P}\mathcal{H} = \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}$. $\mathcal{H}$ can be calculated using the following reasoning.

Let $\mathbf{X}_w$ be a point in the world coordinate system and $\mathbf{X}_n$ be the corresponding point in the normalised coordinate system determined by $\mathbf{P}$. The image point of $\mathbf{X}_w$ in homogeneous coordinates is

$$\mathbf{Y} = \mathbf{P}\begin{pmatrix}\mathbf{X}_w \\ 1\end{pmatrix} \tag{2.12}$$

Similarly, the image of the same point projected from the normalised coordinate system is

$$\mathbf{Y} \quad \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{X}_n \\ 1\end{pmatrix} \tag{2.13}$$

We want $\mathcal{H}$ so that

$$\begin{pmatrix}\mathbf{X}_w \\ 1\end{pmatrix} \quad \mathcal{H}\begin{pmatrix}\mathbf{X}_n \\ 1\end{pmatrix} \tag{2.14}$$

Since the last row of $\mathcal{H}$ will be $\begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}$,

$$\mathcal{H} \quad \begin{pmatrix}\mathcal{H}_{11} & \mathbf{h} \\ \mathbf{0}^T & 1\end{pmatrix} \tag{2.15}$$

Letting $\mathbf{P} = \begin{pmatrix}\mathbf{P}_{11} & \mathbf{p}\end{pmatrix}$, it can be shown that

$$\mathbf{P}_{11}\mathbf{h} + \mathbf{p} = \mathbf{0} \tag{2.16}$$

and

$$\mathbf{P}_{11}\mathcal{H}_{11} + \mathbf{p} = \mathbf{I} \tag{2.17}$$

Once $\mathcal{H}$ is found, the normalised quadric $\mathbf{Q}_n$ is calculated as follows

$$\mathbf{Q}_n = \mathcal{H}^T\mathbf{Q}_w\mathcal{H} \tag{2.18}$$

The projected conic $\mathbf{C}$ can then be calculated using 2.10. Figure 2.4 illustrates the mapping of a world view quadric $\mathbf{Q}$ (ellipsoid) to a conic $\mathbf{C}$ (ellipse) in the image plane.

2.3 Target Localisation

Having explained our chosen target formulation we now describe how we use it to obtain a measurement from the image. Since the computation involved in the histogram representation and matching is quite substantial, we want to keep the number of image samples required to find $\mathbf{y}$ as low as possible. The adopted target localisation process is given in figure 2.5.

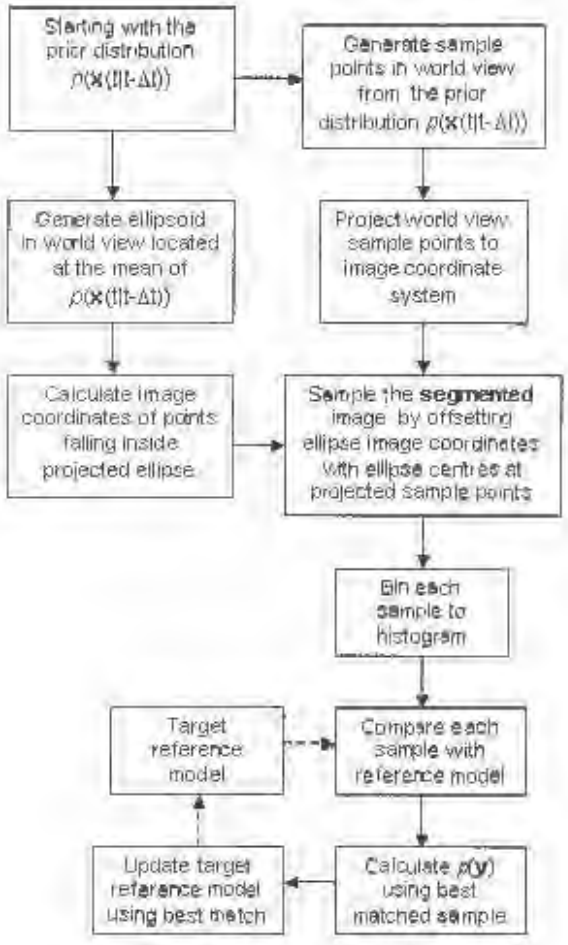


Figure 2.5: Target localisation process overview.

Sample generation

We generate m sample points $\mathbf{x}_s$ according to the world view prior distribution of the target $p(\tilde{\mathbf{x}}(t|t-\Delta t))$. Figure 2.6(a) shows a top-view of the world coordinate system where 4 targets are present. The ellipse shows a line of equal probability for the world view prior distribution and the +'s show the world view sample points that were drawn from that distribution.

Projection of sample points

We proceed by projecting these sample points to the image view by using the transformation described in the previous section (equations 2.3, 2.4, 2.5 and 2.10) to obtain the set of points $\{\mathbf{y}_s\}_{1:m}$. An illustration of this process is shown in figure 2.6(b). The +'s show the $\{\mathbf{y}_s\}_{1:m}$ values for each targets and the ellipses show the lines of equal probability for the prior distribution projected onto the image.

Generation of 3-D ellipsoid

In this step we generate the 3-D ellipsoid quadric matrix $\mathbf{Q}$ at centred at world coordinate $(\hat{x}(t|t-\Delta t), \hat{y}(t|t-\Delta t), r_{\bar{Z}})$ and dimension $r_x \times r_y \times r_z$ using equation 2.8.

Computing image points that fall inside projected ellipse

We calculate $\mathbf{C}$ from $\mathbf{Q}$ using equation 2.10. By applying equation 2.11 we obtain undistorted image points that fall inside the projected ellipse described by $\mathbf{C}$. After re-scaling using equation 2.3 these points can be applied to equations 2.4 and 2.5 to obtain the distorted image points that fall inside the ellipse. We call this set of points $\mathbf{y}_c$.

Sampling of image

By centering the cluster of points $\mathbf{y}_c$ at locations $\{\mathbf{y}_s\}_{1:m}$ we obtain m different elliptical image samples. An illustration of this is given in figure 2.6(c) where elliptical shaped

samples are taken from the segmented image.

Binning of each sample

For each sample taken, we bin the colour-height information to obtain m candidate distributions $\{\mathbf{p}_u(\mathbf{y}_s)\}_{1:m}$. Note that only foreground pixels are considered in this process.

Similarity Measure

The similarity or likelihood measure between the model distribution $\mathbf{q}$ and a candidate distribution $\mathbf{p}(\mathbf{y}_s)$ is obtained using the discrete version of the Bhattacharyya Coefficient defined by:

$$\rho(\mathbf{y}_s) = \sum_{u=1}^{n} \sqrt{p_u(\mathbf{y}_s) q_u}. \tag{2.19}$$

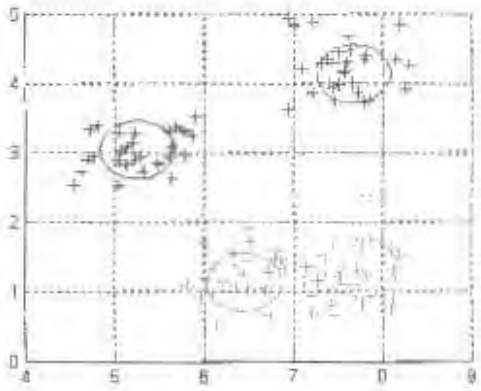
Figure 2.6(e) shows the colour model matching output $\rho(j, i)$ surface for the white target model reference in the neighbourhood of the white and pale blue targets. In figure 2.6(f) the response ρ is shown for samples that were taken (shown by the inverted red triangles).

Calculation of measurement distribution $p(\mathbf{y})$

Since the measurement distribution $p(\mathbf{y}$ is Gaussian it can be expressed simply by a mean vector $\mathbf{y}$ and a covariance matrix $\mathbf{N}_y$. The mean vector $\mathbf{y}$ is approximated to $\{\mathbf{y}_s\}_{max}$ where $\{\rho(\mathbf{y}_s)\}_{max} = \max(\{\rho(\mathbf{y}_s)\}_{1:m})$. Figure 2.6(d) shows the sample which gave the best match. The covariance matrix $\mathbf{N}_y$ is a function of $\{\rho(\mathbf{y}_s)\}_{max}$. The higher $\{\rho(\mathbf{y}_s)\}_{max}$, the better the match and so the lower the uncertainty should be. $\mathbf{N}_y$ is thus given by

$$\mathbf{N}_y = \frac{1}{\rho(\mathbf{y})} \begin{pmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_j^2 \end{pmatrix}, \tag{2.20}$$

where σ_i^2 and σ_j^2 are the row and column uncertainties, which are fixed throughout the tracking process.

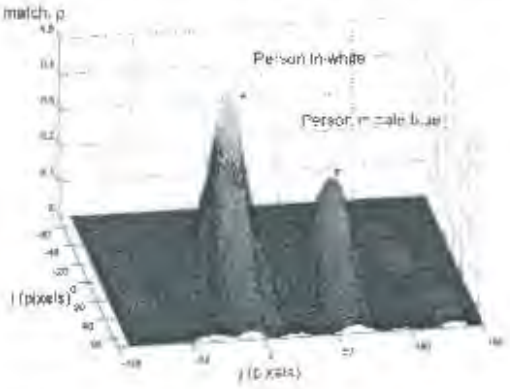


(a) Sample points in world view for 4 targets.

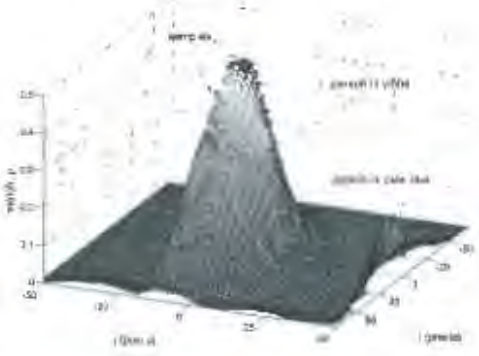
(b) Projected sample points in segmented image view.

(c) Ellipse shaped samples taken of the white target. Ellipses are centred at the projected samples points

(d) Best candidate for white target.



(e) Surface plot of the match ρ in the vicinity of the white target.



(f) Close up of surface plot. The red triangles show each of the samples that were taken and their corresponding match ρ

Figure 2.6: Sampling the segmented image in the target localisation process.

Colour model update

The colour reference model is initialised using only one frame (see next chapter section 3.5), so there is always some probability that important colour features might have been missed on initialisation. Also, light variations can alter the colour features of tracked subjects quite drastically, especially in outdoor scenes. To overcome these effects we slowly adapt the colour model of each of the tracked subjects over time as done in [28]. The update of the target reference model is implemented by mixing the reference model with a small part of the best candidate model using the equation

$$\mathbf{q}_k = (1 - \lambda_c)\mathbf{q}_{k-1} + \lambda_c\{\rho(\mathbf{y}_s)\}_{max}, \tag{2.21}$$

λ_c is a learning rate parameter.

Chapter 3

The Person Tracking Algorithm

The previous chapter described the observation process that is executed each time a new frame is available to the system. In this chapter we describe how we use this to track targets in the chosen world-view coordinate system. We start with an overview of the algorithm before describing in detail some of the more important components namely state transition, state update, foreground update and track initialisation, and track termination.

3.1 Algorithm Overview

Figure 3.1 gives an overview of the tracking algorithm. The algorithm makes a number of assumptions about the cameras, the scene and the targets. It assumes that cameras are fixed, that they are calibrated to a unique world coordinate system and that there is some overlap between the views. Entry and exit points of monitored scenes are assumed to have been specified beforehand. Targets to be tracked are people of average size walking or standing on a horizontal ground plane. No slopes or steps are taken into account, although if one were to construct a detailed description of such instances, then the system could quite easily be adapted to cope with them.

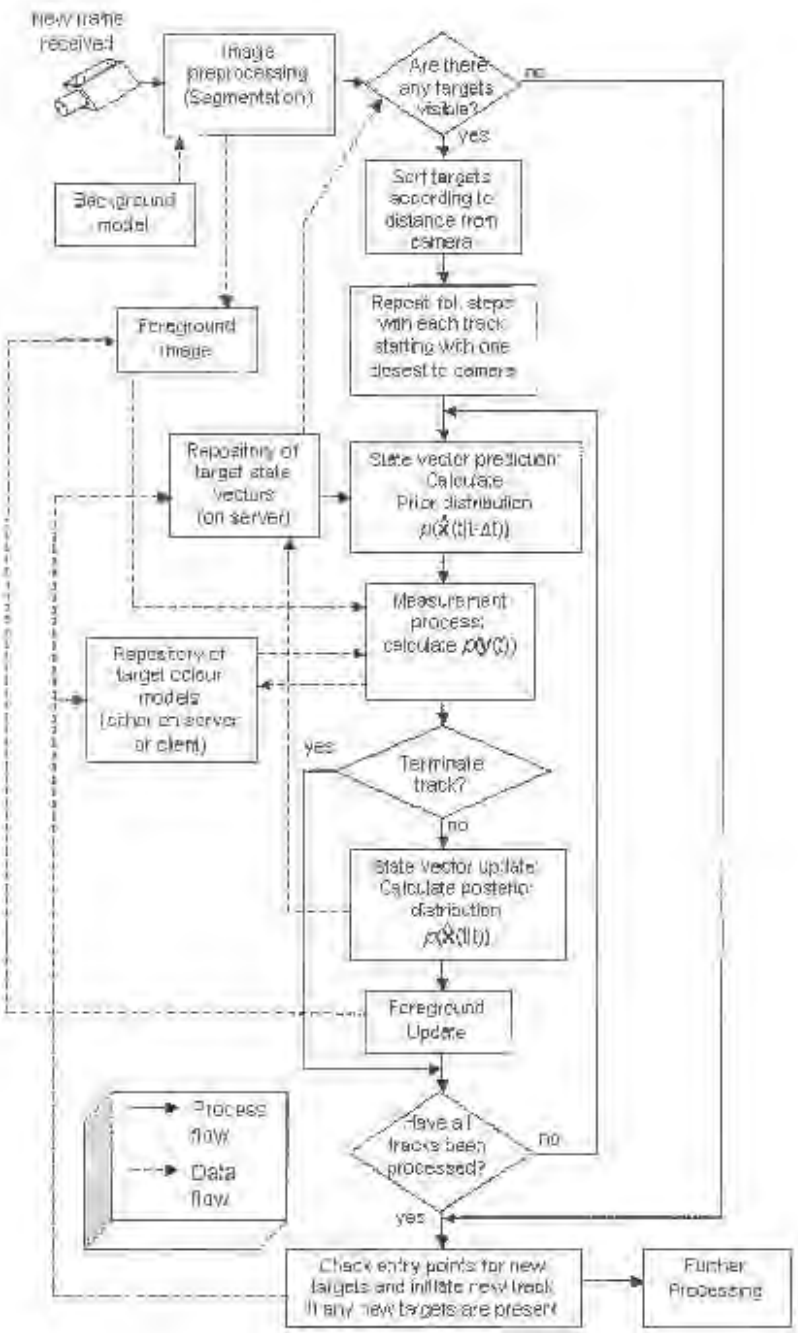


Figure 3.1: Tracking algorithm overview

3.2 Image Preprocessing

The fact that we are using fixed cameras allows us to perform foreground/background segmentation at relatively low computational costs. This step considerably reduces the amount of image pixels to be processed as well as provides further constraints on the measurement process. We demonstrate in chapter 5 that the tracking process is not seriously affected by poor segmentation but does suffer if no segmentation is performed. This justifies our choice for a simple segmentation algorithm summarised as follows:

The background model is simply the image of the monitored scene when it contains no targets. A difference D is calculated according to

$$D = |I - B| \tag{3.1}$$

over each pixel, where I is the current image and B is the background image in RGB coordinates. D values are then simply thresholded to mark foreground regions.

3.3 State Representation and Transition

For each person being tracked, the system uses a separate single world-view model. This model describes the x and y position and velocity (a 4-D state vector $\mathbf{x} = (x, y, \dot{x}, \dot{y})^T$), together with a measure of the uncertainty in this vector (a 4×4 diagonal covariance matrix $\mathbf{N}_x$) in the chosen 3-D world coordinate space. This contrasts with the *Bramble* [14] implementation, where all the target states are parameterised by a single state space model. This one state space model formulation allows for occlusions to be handled implicitly. In our implementation we have to handle occlusions explicitly. This is achieved by processing each of the separate target models in order according to their distances to the cameras (or depth) starting with the closest one, and then modifying the image (as described later in this chapter) so that the influence of targets that might be occluding other targets is reduced.

The individual target model $\mathbf{x}$ follows a transition relationship of the form

$$\mathbf{x}(t + \Delta t) = \mathbf{F}(\Delta t) \cdot \mathbf{x}(t) + |\Delta t|\mathbf{v}(t) \tag{3.2}$$

where

$$\mathbf{F}(\Delta t) = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

and where Δt is the time that has elapsed since the model was last updated, and $\mathbf{v}(t)$ is a Gaussian noise sequence. Note that the uncertainty grows with time between observations. Also, since we are scaling the uncertainty by the modulus of Δt, we allow negative values of Δt. This allows observations to be made out of sequence, which could easily occur when tracking with multiple cameras.

Given an initial or a previous estimate of the state vector at time $t-\Delta t$, namely $\hat{\mathbf{x}}(t-\Delta t|t-\Delta t)$ with associated uncertainty $\mathbf{M}(t-\Delta t|t-\Delta t)$, the predicted state and associated uncertainty at time t are given by

$$\hat{\mathbf{x}}(t|t-\Delta t) = \mathbf{F}(\Delta t) \cdot \hat{\mathbf{x}}(t-\Delta t|t-\Delta t) \tag{3.3}$$

$$\mathbf{M}(t|t-\Delta t) = \mathbf{F}(\Delta t)\mathbf{M}(t-\Delta t|t-\Delta t)\mathbf{F}^{T}(\Delta t) + |\Delta t|\mathbf{N}_v(t). \tag{3.4}$$

3.4 State Update

The update step can be summarised as follows:

> Given an observation $\mathbf{y}(t)$, the predicted state vector $\hat{\mathbf{x}}(t|t-\Delta t)$, and the respective uncertainties $\mathbf{N}_y(t)$ and $\mathbf{M}(t|t-\Delta t)$, make an optimal estimate of the location $\hat{\mathbf{x}}(t|t)$ and its associated uncertainty $\mathbf{M}(t|t)$.

This is done using the Kalman filter formulation:

$$\hat{\mathbf{x}}(t|t) = \hat{\mathbf{x}}(t|t-\Delta t) + \mathbf{K}(t)[\mathbf{y}(t) - \mathbf{h}(t, \hat{\mathbf{x}}(t|t-\Delta t))] \tag{3.5}$$

$$\mathbf{M}(t|t) = \mathbf{M}(t|t-\Delta t) - \mathbf{K}(t)\mathbf{H}(t)\mathbf{M}(t|t-\Delta t). \tag{3.6}$$

Since $\mathbf{h}(t, \mathbf{x})$ is non-linear, $\mathbf{H}(t)$ is calculated by locally linearising $\mathbf{h}$ at $\mathbf{x} = \hat{\mathbf{x}}(t|t-\Delta t)$:

$$\mathbf{H}(t) = \left.\frac{\partial \mathbf{h}(t, \mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(t|t-\Delta t)} \tag{3.7}$$

Thus

$$\mathbf{H}(t) = \begin{pmatrix} \frac{\partial i}{\partial x} & \frac{\partial i}{\partial y} & 0 & 0 \\ \frac{\partial i}{\partial x} & \frac{\partial i}{\partial y} & 0 & 0 \end{pmatrix} \tag{3.8}$$

The Kalman gain is calculated as follows:

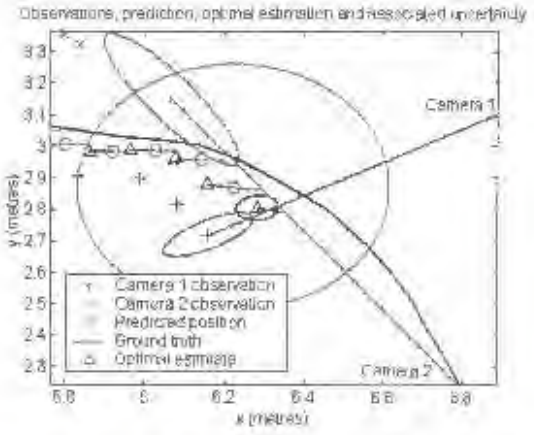
$$\mathbf{K}(t) = \mathbf{M}(t|t-\Delta t)\mathbf{H}^T(t) \cdot [\mathbf{H}(t)\mathbf{M}(t|t-\Delta t)\mathbf{H}^T(t) + \mathbf{N}_p(t)]^{-1}. \tag{3.9}$$

Figure 3.2 gives an overall picture of the update step. Frames taken from the '2-Cam Debtech' sequence where one person is tracked by two cameras with overlapping views are shown. Figures 3.2(b) and 3.2(c) show simultaneous views from camera 1 and camera 2. Figure 3.2(a) shows a top-view of the world coordinate system at that same instant. The 4 ellipses show lines of equal probability of distributions. The large grey (more circular) ellipse is the predicted distribution (or ptior) $p(\tilde{\mathbf{x}}(t|t-\Delta t))$, the elongated red ellipse shows the measurement $p(\mathbf{y}_1)$ obtained from camera 1 the elongated purple ellipse shows the measurement $p(\mathbf{y}_2)$ obtained from camera 2 and finally the smaller blue ellipse shows the estimated position (or posterior) of the position of the target $p(\tilde{\mathbf{x}}(t|t))$. The small grey circles, red +'s, purple crosses and blue triangles represent previous predictions, observations and estimates. The black line shows the ground truth that was defined manually.

3.5 Foreground Update and Initialisation

The updated world position of a target is used by the system to mask out foreground regions associated with the target. This improves the subsequent localisation of other targets that are further away from the camera especially in the event of an occlusion. Figure 3.3(b) shows a scene from the 'Colourful People' sequence with 4 targets being tracked and a new person having just entered the scene. Figure 3.3(c) shows the foreground image after the target closest to the camera has been masked out. The measurement processes for subsequent targets, which are partially occluded, are thus not influenced as much by that first target.

When foreground regions near entry points are not accounted for by any of the currently tracked targets, a new track is initialised. In other words, initialisation of a new track is triggered if at a predefined entry point the foreground pixel count of an ellipse-shaped sample is above a certain threshold T_{init}. Figure 3.3(a) shows entry/exit points for the



(a) World view of estimate with observations from 2 cameras

(b) Camera 1

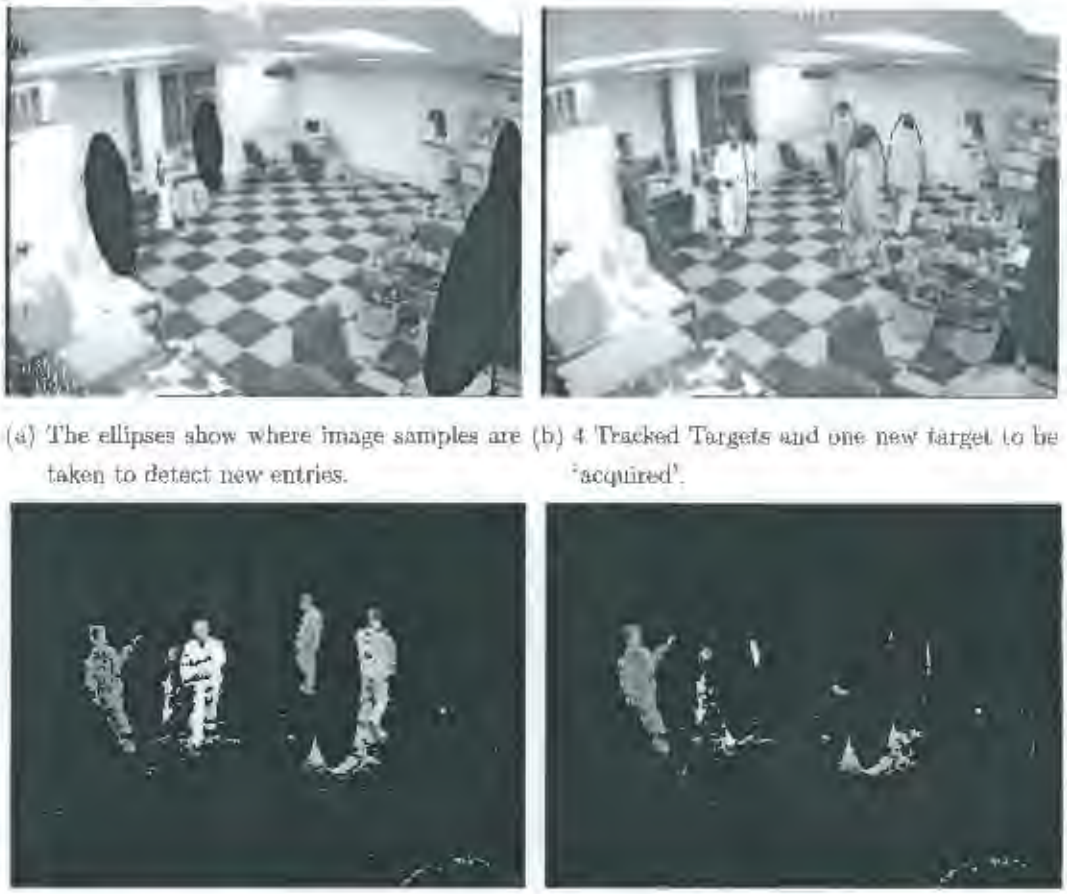
(c) Camera 2

Figure 3.2: Simultaneous views from camera 1 and 2 at that particular instant with estimated position of ellipsoid projected back onto the image

'Colourful People' sequence. The unaccounted for pixels near the entry point as shown in Figure 3.3(d) are used to initiate a new track.

Once detected the new target is tagged, a new state vector containing its position and velocity is generated on the server and a colour reference model for the target is defined and distributed to all clients. This simplistic approach to initialisation was implemented at the very last stages of the project and works well on the datasets presented in this thesis. However, it needs to be improved further. For instance, should a person entering the scene be occluded by another person already in the scene or entering at the same time, his/her

entry may not be detected by this approach.



(a) The ellipses show where image samples are taken to detect new entries.

(b) 4 Tracked Targets and one new target to be 'acquired'.

(c) Foreground model after closest target is masked out.

(d) Foreground model after all 4 tracked subjects are masked out.

Figure 3.3: Foreground model update and initialisation.

The updated foreground is also a useful mask for the background model update. This is implemented by a similar equation to the one used for the colour model update:

$$B(t) = (1 - \lambda_b \Delta t) B(t - \Delta t) + \lambda_b \Delta t F I(t), \tag{3.10}$$

where B is the background image (RGB), F is the mask (binary) made up of the projected ellipsoids at the estimated locations of the tracked subjects, I is the current image (RGB), and λ_b is the learning rate parameter. A background update at every frame slows the tracking process down. Thus depending on how much lighting variation one expects, the background may be updated at regular time intervals. The background update step was not

implemented when testing the algorithm with the selected sequences because they were too short to contain any drastic lighting conditions that would have affected the segmentation.

3.5.1 Termination of Track

When a target leaves the scene we expect observations with low quality of match ρ_{max}. The main difficulty is to decide whether the poor quality of the observations are due to the target having indeed left the scene or whether the target is simply being occluded temporarily. Figure 3.4 shows a typical response of the colour match variable ρ_{max} through a particular person in a tracking sequence. As one can see, ρ_{max} is as low during occlusion as it is when the target exists, so it is not a sufficient indication that the target has indeed exited. Fortunately we know the world view locations of exit points and using this extra information we can more robustly terminate tracks. Shown on the same axes in the figure, is the variable d_{min}, which is the world view distance of the estimated position of the tracked person to the nearest exit point. Occlusions tend to be short, whereas as a target exit results in a more sustained low ρ_{max}. Hence a track is terminated if the following conditions are simultaneously met:

1. $d_{min} < T_{d_{min}}$ and
2. $\text{mean}(\rho_{max}(t - T_t), ..\rho_{max}(t)) < T_\rho$.

where $\text{mean}(\rho_{max}(t - T_t), ..\rho_{max}(t))$ is the average value of the best match obtained over the last T_t seconds,and $T_{d_{min}}$ and T_ρ are threshold values for distance to closest exit point and value of best match.

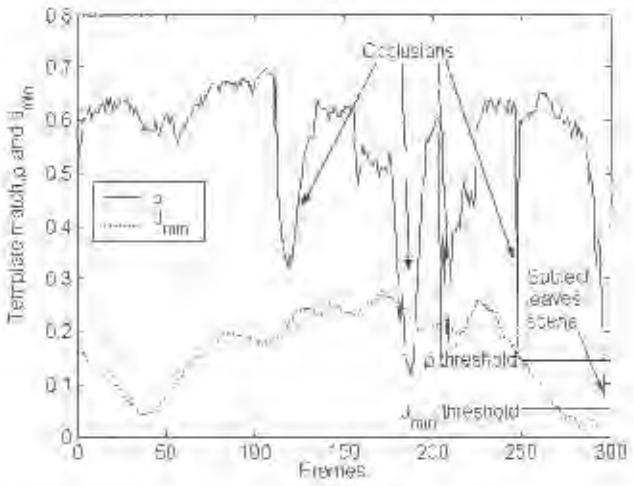


Figure 3.4: Plot of matching response ρ_{max} and d_{min} for orange person in 'Colourful people Sequence' for 300 frames. The distance d_{min} has been scaled to fit the axes, so a value of 0.1 actually represents 1.0 metres.

Chapter 4

Camera Calibration Suited to Person Tracking Applications

In this chapter we present a camera calibration solution for person tracking applications. Our approach is based on a 2-stage method proposed by Jones et al. [16]. In the first stage the method uses observed image size variations of objects obtained from a sequence of images to automatically recover the local ground-plane transformation, by making some assumptions about the camera and the monitored scene. In the second stage, the transformation between these local ground planes is recovered. For cases where assumptions made by the automatic method are breached we propose some adaptations that require some level of user/operator intervention. We end this chapter by briefly describing an approach that makes use of visual cues present in the scene, based on a method by Tsai [43] that also proves to be of some practical use.

4.1 Local Ground Plane Calibration

To recover the local ground plane transformation automatically using the proposed approach we need 2 things. First, we need a model that explains how the size (or height) of the projected 2-D image of an object varies with its vertical position in the image in terms of camera parameters. Then we need a process for observing and recording this size

variation given a video sequence or set of images taken using the cameras that are to be calibrated.

4.1.1 The Projected Object Height Model

It is shown in [16] that the height h of the image of an object of height H located at image row coordinate i can be related to camera parameters ϕ, t_z, f_i^v and i_0 by

$$h = \frac{\cos\phi\sin\phi((f_i^v)^2 - (i_0 - i)^2) + f_i^v(i_0 - i)(\cos^2\phi - \sin^2\phi)}{f_i^v(t_z/H - \cos^2\phi) + (i_0 - i)\cos\phi\sin\phi}, \tag{4.1}$$

where ϕ is the pitch angle of the camera, t_z is the height of the camera above the ground, f_i^v is the focal length to pixel width ratio and i_0 is the row coordinate of the optical centre. The camera parameters ϕ, t_z, f_i^v and i_0 are sufficient to describe the local image to ground plane transformation if a simplified camera model is used. Figure 4.1 shows an illustration of this simplified model where the following assumptions are made:

- The pan and roll angles of the camera θ and ψ are both very small or equal to zero.
- The origin of the ground plane coordinate system is directly below the optical centre of the camera.
- The column pixel width is equal to the row pixel width $\alpha_j = \alpha_i$.
- The optical centre (j_0, i_0) is assumed to be the image centre.
- Lens radial distortion effects are ignored.

Jones et al. [16] further assume that the projected 2-D image of an object varies linearly with its vertical position in the image, from zero at the horizon (at row coordinate i_h) to a maximum at the at the bottom row of the image. In other words, they assume that the (i, h) relationship given in equation 4.1 is linear. They however recommend that precautions be taken when making this assumption for steep camera angles. Figure 4.2 gives a plot of the projected height h versus the vertical image position i. From the plot we can see that indeed the relationship deviates more and more from linearity as ϕ is decreased.

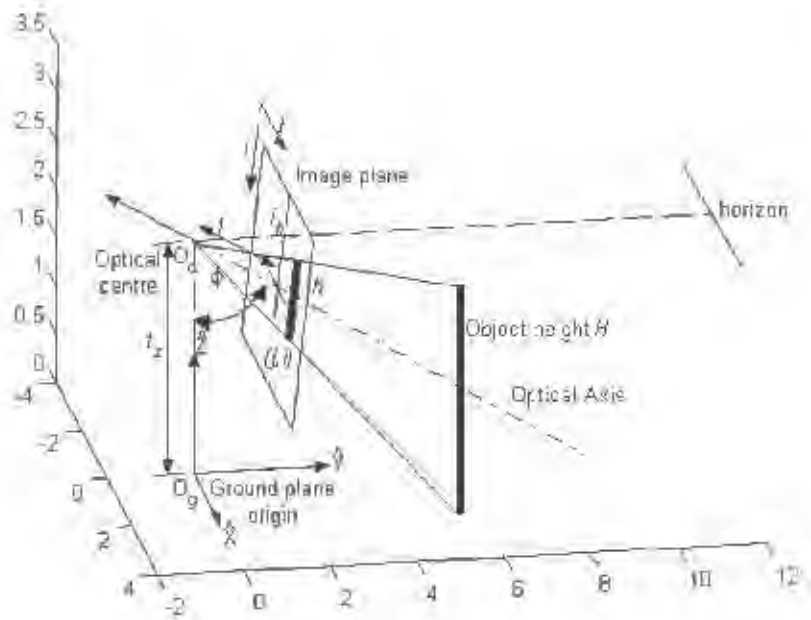


Figure 4.1: Simplified Camera model

The linear relationship is expressed as follows:

$$h = \gamma(i - i_h), \tag{4.2}$$

where γ is the *height expansion rate* and i_h is the pixel row coordinate of the horizon. By recording how the height h of the image of an object varies with its vertical position i in the image over a number of frames, the values of γ and i_h can be recovered.

If an object of height H is placed (upright) on the ground plane at the point where the projection of the optical axis intersects the ground plane, i will be equal to i_0. We denote the image height of this object by $h(i_0)$. We can find $h(i_0)$ by substituting i by i_0 in equation 4.1 and simplifying:

$$h(i_0) = \frac{f_i^v H \cos\phi \sin\phi}{t_z - H\cos^2\phi}. \tag{4.3}$$

We can also find $h(i_0)$ using the linearised height model given in equation 4.2:

$$h(i_0) = \gamma(i_0 - i_h). \tag{4.4}$$

The pitch angle ϕ is directly related to the horizon parameter i_h, i_0 and f_i^v by

$$(i_0 - i_h) = f_i^v \cot\phi. \tag{4.5}$$

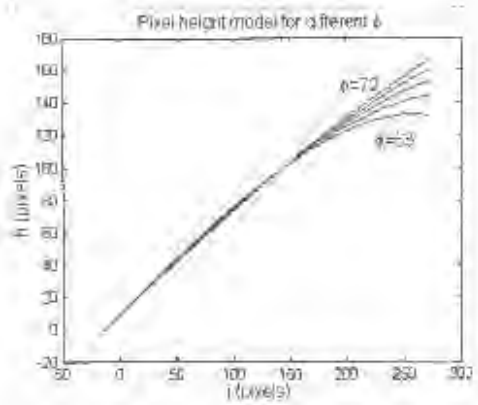


Figure 4.2: Height of image of object h variation with row coordinate of image of object i for different camera pitch angles ϕ.

Substituting $(i_0 - i_h)$ from equation 4.5 in equation 4.4, equating to 4.3 and simplifying yields:

$$\sin^2\phi = \frac{\gamma(t_z - H)}{H(1-\gamma)}. \tag{4.6}$$

We thus have a function that relates the height expansion rate γ to the pitch angle ϕ, the camera height t_z and the height of the object H. If we know H and t_z and γ we can calculate ϕ using equation 4.6. The expansion rate γ can be obtained by making a suitable number of (i, h) observations. The process of recording (i, h) observations is a critical aspect of this method since it relies on information contained in video sequence images and is described in the next section.

If we use n objects of different height H we get a different expansion rate γ for each object. Substituting each of the γ and H values for the different objects in equation 4.6 we get n equations relating ϕ to t_z. Let

$$\Gamma = \frac{1-\gamma}{\gamma} \tag{4.7}$$

and

$$\eta = \frac{1}{H}. \tag{4.8}$$

We can rearrange equation (4.6) and express it in terms of Γ and η as follows

$$\Gamma = \frac{t_z}{\sin^2\phi} \cdot \eta - \frac{1}{\sin^2\phi} \tag{4.9}$$

By applying linear regression to the set of points (η, Γ) we can solve for ϕ and t_z. Once ϕ is known, f_i^a is calculated using equation 4.5. We thus have a method to recover the 3

required camera parameters ϕ, t_z, f_i^a if we know the height H of each object in the scene and the corresponding expansion rate γ.

Figure 4.3(a) shows a frame taken from the 1-Cam Jammie dataset containing 3 people of different heights. It also illustrates an example of an (i, h) observation. Figure 4.3(b) shows a plot of how the height h of the image of the 3 people in the scene vary with their vertical position in the image i over several frames, as they move towards and away from the camera. For each person, a different expansion rate γ is obtained by linear regression. Note the position of i_h just above the top of the image, where all three regressed lines intersect. Figure 4.3(c) shows a plot of Γ vs η for the 3 values of γ, from which we can infer the camera parameters ϕ and t_z.

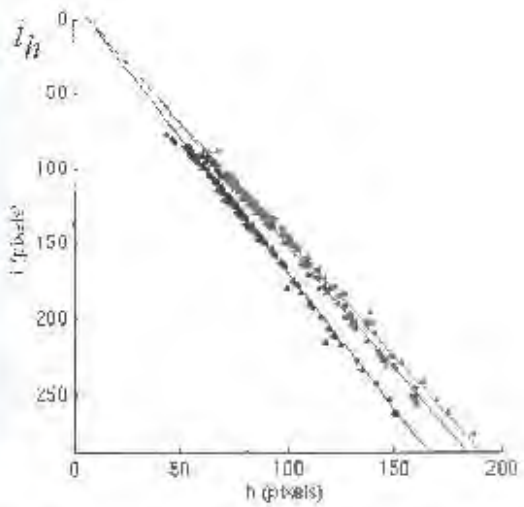
4.1.2 Learning the Height Model Automatically

The linear height model expressed in equation 4.2 can be learnt from the scene automatically by accumulating (i, h) object observations. This is achieved using a motion detection process (or segmentation) to extract components of connected components of moving pixels (or blobs). The bounding box $(i_{min}, i_{max}, j_{min}, j_{max})$ of each segmented blob generates a height $h = i_{max} - i_{min}$ and a row position $i = i_{max}$.

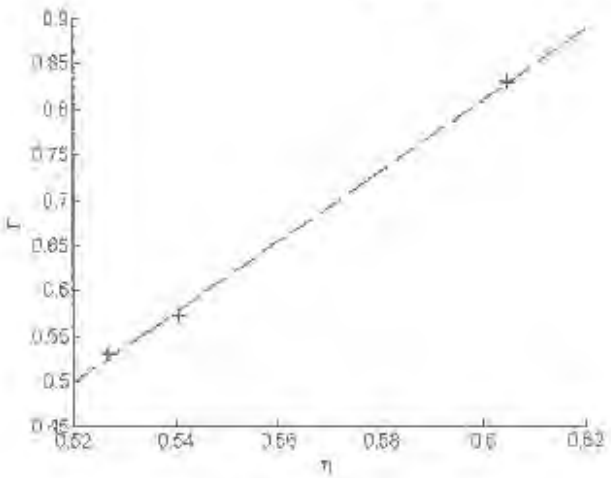
Figure 4.4 shows the screen view of the operator interface for the rudimentary blob tracker that was implemented to automatically record (i, h) observations. Since no calibration information is available, only 2-D image information can be used. Track initialisation, target representation and localisation used in this blob tracker is based on principles similar to those used in the person tracker described in chapter 2 and 3. We make use of no filtering and the tracker relies quite heavily on good segmentation. Any complexities arising from occlusions or shadows and reflections are not handled well. We thus assume that during the calibration process, the operator will have some control over what goes on in the scene. For instance, only one person need to be in the scene at a time.

In an attempt to improve the quality of segmentation the blob tracker includes a background model update feature as well as a shadow identification feature. The background update is achieved in a similar fashion to the colour model update presented in chapter 2, where the reference background model is mixed with a small part of each new frame

(a) Frame taken from 1-Cam Jammie sequence.



(b) Measurements of (i, h) for 3 people of different heights



(c) Plot of Γ vs η.

Figure 4.3: Calibration using automatic method on '1-Cam Jammie' dataset.

processed. The shadow identification process is based on work by Cucchiara [9], where shadow pixels are identified by the following criteria:

- $\tau_{V1} \leq \frac{V_F}{V_B} \leq \tau_{V2}$,
- $S_F - S_B \leq \tau_S$ and
- $\min(|H_F - H_B|, |H_F - H_B|) \leq \tau_H$,

where H, S and V are the hue saturation and intensity values associated with each pixel being tested. The subscript F indicates that the pixel belongs to the identified foreground image and B indicates that it belongs to the background model.

4.1.3 Obtaining the Height Model manually

In cases where the operator has little control over what goes on in the scene or where good segmentation is not achievable, (i, h) observations have to be made manually. Figure 4.5 shows the operator interface that was implemented for manually recording (i, h) observations.

4.1.4 Manually Adjusting the Local Calibration

There are camera configurations that are not suited to using the calibration method that was just presented. Such configurations include:

- Cameras with considerable pan and roll angles (as in the PETS 2002 sequence).
- Cameras with steep pitch angles (as in the PETS2004 sequence).
- Cameras with very wide angles or high distortion coefficients (as in the 4-Cam DIP sequence).

Figure 4.6 shows the operator interface that was implemented for manually finding local ground plane calibration parameters. The frame shown is taken from the PETS 2002

Figure 4.4: Operator interface for the blob tracker. The top left window shows the latest received image and 3 tracked targets and their bounding boxes. The window below shows the (h, i) data that was collected. The other 4 windows show (clockwise from top-left): the current background image, the foreground regions with shadow, the foreground regions with shadows masked out and foreground regions (including shadow regions) that will be masked out in the background update process.

Figure 4.5: User interface for obtaining (i, h) data manually. The operator is simply requested to point and click on the feet and head of an object/person of known height H. A collection of (h, i) points are obtained for as many frames as the operator judges necessary.

sequence. The camera used to capture this sequence has a non-zero pan angle and quite considerable distortion hence cannot be calibrated using the approach based on Jones et al.'s method. The calibration procedure is quite simple. The operator makes an initial guess of what the camera parameters are. Then by trial and error, he adjusts each of the parameters is adjusted until satisfactory results are obtained. Trials are evaluated visually by projecting ellipses (calculated using the trial parameters) onto the image of people in the scene. The accuracy of this approach depends on the operator skill level. It is however still less labour intensive than using hand-measured calibration points.

Figure 4.6: User interface for performing local ground plane calibration manually.

4.1.5 Local Ground Plane Camera Pose Recovery

In some instances, the operator may be able to calibrate a camera for internal parameters before it is deployed to a monitored scene. This would typically be done using a calibration method such as the one proposed by Zhang [48] that makes use of a calibration object. Since f_i'' is known only the camera pose (parameterised by $\mathbf{t}_z$, ϕ and maybe θ, ψ) need to be estimated when the camera is deployed. Whether one resorts to using the automatic or the manual methods described above, more accurate results are generally obtained

4.2 Registering Multiple Cameras

The second stage of the calibration method recovers the transformation between the local ground planes of different cameras by matching tracks obtained from each of the cameras.

4.2.1 Automatic Approach

We assume a starting point of 2 cameras for which the local image to ground plane calibration parameters are known. The generalisation of the method to systems with more than two cameras is then relatively simple. The ground plane coordinate systems of temporally synchronised observations of the same 3-D object are related by a rotation $\mathbf{R}_g(\beta)$ and translation $\mathbf{t}_g$:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{R}_g(\beta)\mathbf{x}_2 + \mathbf{t}_g, \\ \dot{\mathbf{x}}_1 &= \mathbf{R}_g(\beta)\dot{\mathbf{x}}_2. \end{aligned} \tag{4.10}$$

where β is the angle between the two cameras, and $\mathbf{x}_1$, $\dot{\mathbf{x}}_1$ and $\mathbf{x}_2$, $\dot{\mathbf{x}}_2$ are positional and velocity estimations of an objects measured in the local ground plane coordinate systems of two cameras c_1 and c_2 respectively. Given a pair of observations $\mathbf{x}_1(t)$, $\dot{\mathbf{x}}_1(t)$ and $\mathbf{x}_2(t)$, $\dot{\mathbf{x}}_2(t)$ at time t, the transformation estimates may be defined as:

$$cos(\beta(t)) = \frac{\dot{\mathbf{x}}_1(t) \cdot \dot{\mathbf{x}}_2(t)}{|\dot{\mathbf{x}}_1(t)| \cdot |\dot{\mathbf{x}}_2(t)|} \text{ and} \tag{4.11}$$

$$\mathbf{t}_g(t) = \mathbf{x}_1(t) - \mathbf{R}_g(\beta(t))\mathbf{x}_2(t) \tag{4.12}$$

We make the assumption that only one target is tracked by the two cameras at a time during the observation process and the observations are temporally synchronised. In other words correspondences of the data from cameras c_1 and c_2 are known. After collecting a sufficient number of observations (say from $t = 0$ to $t = T$), we find the angle β by taking the mean all the observed values. In other words

$$\beta = mean\{\beta(t)\}_{t=0,1,\ldots,T} \tag{4.13}$$

The translation $\mathbf{t}_g$ is then calculated as follows:

$$\mathbf{t}_g = mean\{\mathbf{t}_g(t)\}_{t=0,1,\ldots,T} \tag{4.14}$$

Let $[\mathbf{R}_1 \quad \mathbf{t}_1]$ and $[\mathbf{R}_2 \quad \mathbf{t}_2]$ be the local-ground-plane transformations for cameras c_1 and c_2. Once β and T are calculated, $[\mathbf{R}_1 \quad \mathbf{t}_1]$ can be transformed so that tracking takes place in a common ground coordinate system:

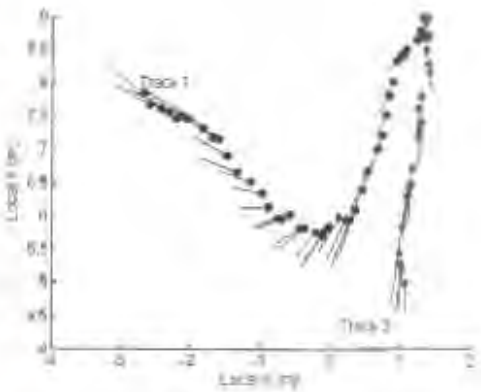
$$[\mathbf{R}_1^{new} \quad \mathbf{t}_1^{new}] = \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_g(\beta) & \mathbf{t}_g \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.15}$$

Figure 4.7 illustrates the process described in this section. Figure (a) shows frame 125 from the 2-Cam Debtech sequence where one target is being tracked. Figure (b) shows tracker position and velocity estimates for sequence frames where the target is present in both camera views. The estimates have been grouped into 2 separate tracks, coloured differently to help with visualisation. Figures (c) and (d) show the same as (a) and (b) but for camera 2. Figure (e) shows a histogram of $\beta(t)$ values that were obtained using equation 4.11. Figure (f) shows the tracks obtained from each view plotted on the same ground plane coordinate system.

4.2.2 Manual Approach

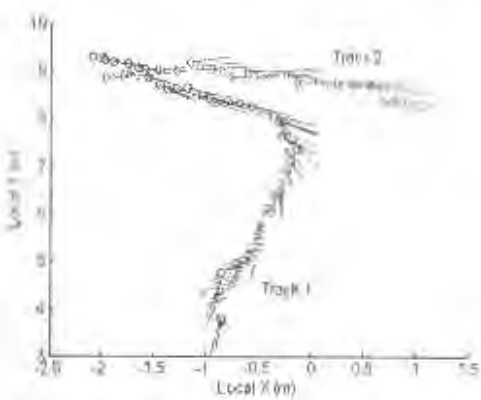
An alternative approach to using monocular tracking is to manually select points and vectors present in both views. This yields more accurate results in instances where poor monocular tracking results are available. Figure 4.8 shows how a collection of 8 manually selected points and vectors were used to obtain the transformation between 2 local ground planes for the 2-Cam Debtech sequence.

(a) Tracking in camera 1 local coordinate system

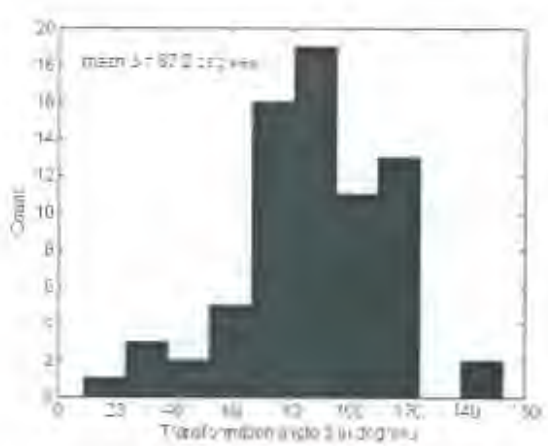


(b) Selected tracks (position and velocity) for tracking camera 1

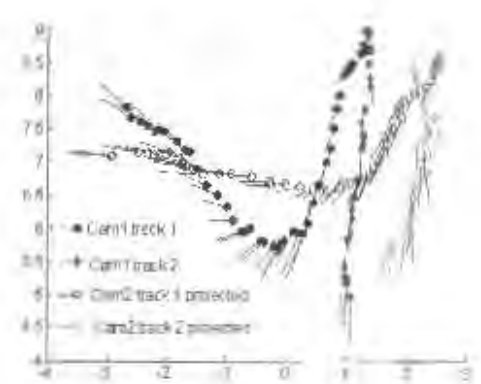
(c) Tracking in camera 2 local coordinate system



(d) Selected tracks (position and velocity) for tracking in camera 2 local coordinate system



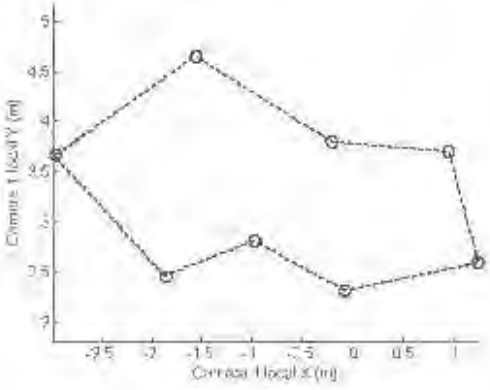
(e) Histogram plot of transformation angle inferred from tracks



(f) Selected tracks from camera 2 local coord. system projected onto camera 1 local coord. system

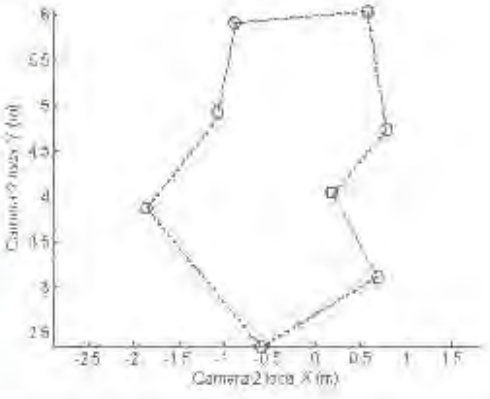
Figure 4.7: Obtaining local to local ground plane transformation using automatically obtained tracks

(a) Hand selected points and vectors from camera 1's view.



(b) Selected points and vectors projected to camera 1's local coordinate system.

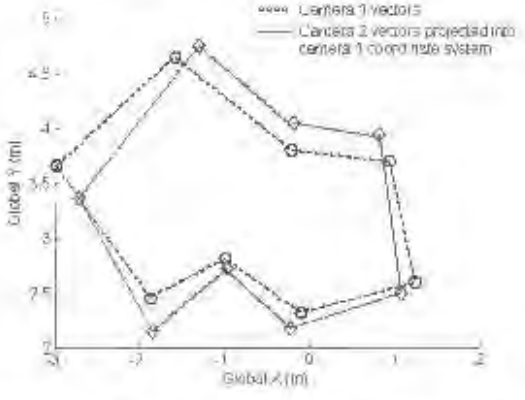
(c) Hand selected points and vectors from camera 2's view



(d) Selected points and vectors projected to camera 2's local coordinate system.

Recovered Transformation

mean β = 85.5 degrees



(f) Selected points and vectors in common ground coordinate system.

Figure 4.8: Obtaining local to local ground plane transformation using manually selected points and vectors.

(a) 1-Cam Jammie sequence (b) PETS 2004 sequence

Figure 4.9: Calibration using Tsai's calibration method that makes use of coplanar calibration points.

4.3 Calibration Using Co-planar Calibration Points

Often one finds that the scene to be monitored contains strong visual cues such as tiled floors or other regular patterns. Camera calibration using these visual cues is sometimes possible and easier than using the methods presented in the previous sections, and so should also be considered. The approach we present is based on a method by Tsai [43] that makes use of co-planar calibration points. The procedure is quite simple. The operator selects a ground plane coordinate system origin. Then calibration points in the field of view of the cameras to be calibrated are selected in a way that their positions relative to the chosen origin can be determined (knowing the dimensions of the tiles for example). Tsai's method (described in appendix B) makes use of the ground coordinates of the calibration points and their corresponding image coordinates to recover camera parameters (including distortion).

Figure 4.9 shows a frame taken from the 1-Cam Jammie sequence and one taken from the PETS 2004 sequence. Both scenes contain floor tiles. In the first case, the dimensions of the tiles were known, in the second, they had to be guessed. The red dots show the points that were chosen as calibration points. The blue diamonds and black crosses show initial and final re-projection estimates made using calculated calibration parameters.

Chapter 5

Results

Performance evaluation of image tracking systems has become a topic of interest as commercial systems are slowly being introduced into society. The performance of a tracker is difficult to measure as ground truth is not easy to generate or obtain. Also, the level of perceptual complexity of tracking problems can vary enormously. Black and Ellis [4] recently presented some work on tracking performance evaluation. They developed quite a sophisticated method that makes use of pseudo-synthetic sequences of controllable levels of perceptual complexity. No attempt was made to replicate this here, it being beyond the scope of our work. However, we try to adhere to some of the propositions made in [4] in our definition of a perceptual complexity metric for the datasets as well as performance metric for evaluating the performance of the person tracker. After discussing the tracking performance of the tracker we perform a simple evaluation of the calibration methods discussed in the previous chapter and how calibration quality affects tracking results. We end this chapter with some preliminary investigations conducted on the tracker with regards to image size and processing speeds.

5.1 Perceptual complexity metric

Each of the datasets used to evaluate the person tracker presented in this thesis presents different difficulties. The complexity metric that is used to quantify the difficulty level of

each of the datasets used to evaluate the tracker is defined as follows[1]:

$$PC = w_1 OC + w_2 CS + w_3 QI + w_4 NE, \tag{5.1}$$

where $w_1 = 0.3$, $w_2 = 0.3$, $w_3 = 0.3$, $w_4 = 0.1$ and

- CS quantifies the colour similarity of tracked subjects.
- OC quantifies the occlusion complexity defined as follows.

$$OC = \frac{1}{NF} \sum_{k=1}^{K} OE_k \times OD_k$$

  where NF is the total number of frames in sequence, K is the number of occlusions, OE_k and OD_k is the extent and duration of occlusion k.

- NE is the number of entry/exit points. This measure gives an indication of how many entry points a scene has. A sequence with access through only one narrow door will have a lower value for NE than a scene with a wide corridor leading into it.
- QI quantifies the quality of images of the sequences. Some sequence images have more noise than others and some sequences are captured using high distortion lenses which also adds to the complexity of the sequence.

Table 5.1 summarises the complexity metric of the 6 datasets used in this thesis. The data sets are sorted in order of complexity starting with the most complex sequence, namely the PETS 2002 sequence.

5.2 Performance of Tracking System

Four main aspects of the tracking system are evaluated. The first one relates to how well the system initiates new tracks. The second aspect relates to how well the system tracks through occlusions. The third aspect evaluates the tracking accuracy and finally, we evaluate how well the tracker detects the exit of a target from a scene. The experiments

[1]Some of the datasets used to evaluate the tracker were also used by Price [32], so the metric presented here is the result of our joint effort.

Dataset	*OC*	*CS*	*QI*	*NE*	*PC*
PETS 2002	0.27	.85	0.80	0.60	0.64
4-Cam Dip	0.46	0.78	0.80	0.10	0.62
Colourful People	0.85	0.68	0.20	0.30	0.55
1-Cam Jammie	0.18	0.96	0.20	0.70	0.47
PETS 2004	0.02	0.75	0.50	0.60	0.44
2-Cam Dcbtech	0.15	0.00	0.20	0.10	0.14

Table 5.1: Perceptual complexity summary for the 6 datasets.

used to evaluate the tracking performance were carried out using a similar procedure for all 6 datasets used. Raw image data (either video sequence file or time-stamped image files) and parameters that are specific to the dataset such as calibration information, entry and exit points, image file(s) information are used as input to the tracking system. The other non-dataset specific tracking system parameters are given in appendix A. The output of the system is a data file that contains the estimated x and y position and velocity (state vector $\hat{\mathbf{x}}$) of each target for each of the frames processed. The world-view estimated positions are projected to image view and displayed as shown in figures 5.1 and 5.2 throughout the processing phase. This facilitates the evaluation of occlusion resolution as well as track initialisation and termination capabilities of the system.

Figures 5.1 and 5.2 show the tracking system at work on each of the test datasets. The red ellipses around the targets are constructed using the projected estimated target world-view positions at particular instants during the tracking process. The small coloured crosses show where the most recent target observations were made in the image. Figure 5.1(a) illustrates some of the features of the tracking system. The top left diagram shows the world-view target position as estimated using two simultaneous views from two different angles also shown in the figure. The colour charts (similar to the ones we used in chapter 2) show the reference and best candidate colour models at frame 50 for each of the cameras/clients. The charts on the bottom left show historical match quality ρ for each camera up to frame 50. The target at that instant is occluded from the view of camera 1. Note how this is reflected by the low match quality ρ at that instant. Figures 5.1(b) and 5.1(c) illustrate an example of tracking through an occlusion in the 1-Cam Jammie sequence. Figure 5.2(a) shows the 3 targets being tracked in the 4-Cam DIP sequence

Note the high distortion present at the image corners. Figures 5.2(b) and 5.2(c) show two frames from the Colourful people sequence. Figure 5.2(d) shows a frame from the PETS 2002 sequence. Note the distortion and the reflections present in the image. Figure 5.2(e) shows a frame taken from the PETS 2004 sequence. Note the big patch of sunlight on the lower-left corner of the image.

5.2.1 Track Initialisation

The *Track Initialisation* performance is evaluated using a detection rate TDR_{in} and a false alarm rate FAR_{in} metric defined as follows:

$$TDR_{in} = \frac{\text{Total True Positives}}{\text{Total number of entries}} \tag{5.2}$$

and

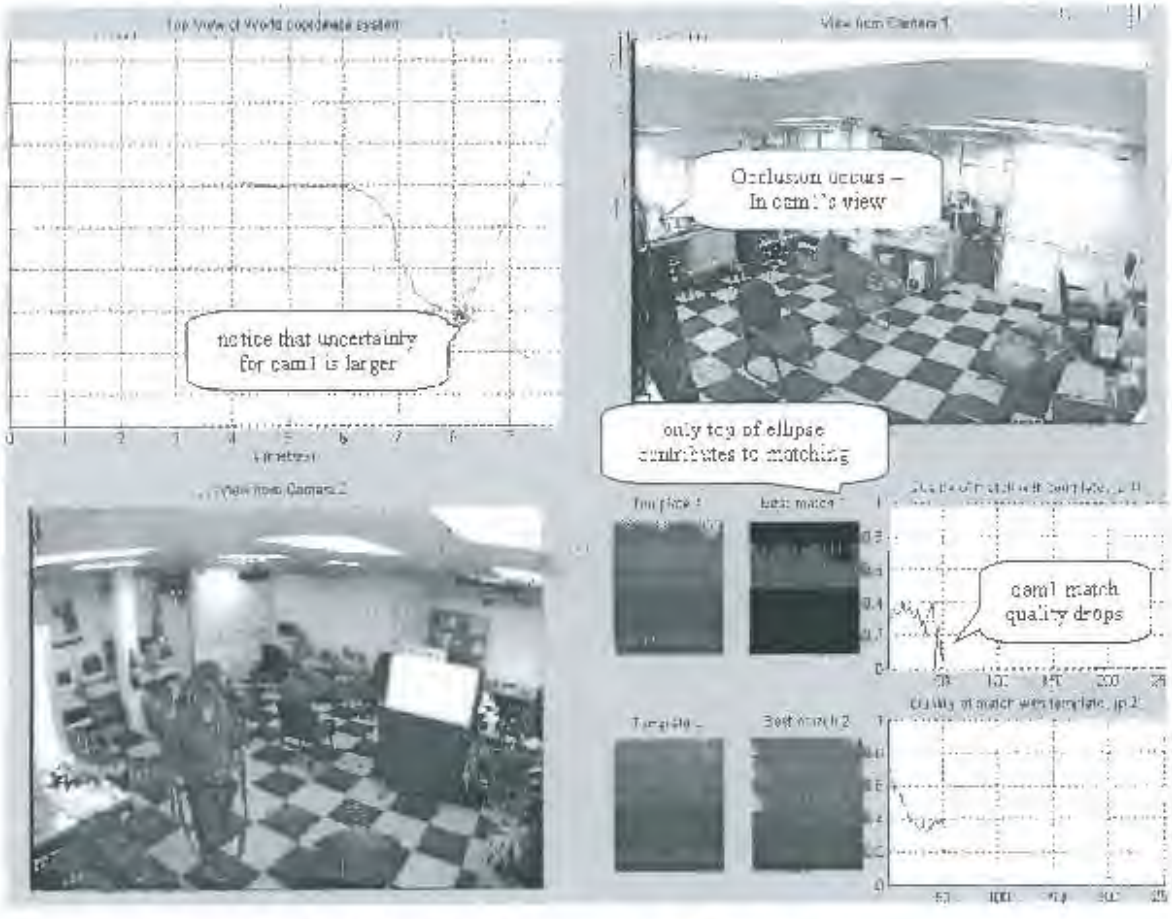
$$FAR_{in} = \frac{\text{Total False Positives}}{\text{Total number of entries}}. \tag{5.3}$$

Table 5.2 gives the TDR_{in} and FAR_{in} results for 5 datasets. The number of entries NE and perceptual complexity PC metrics are also given. The 4-Cam DIP sequence datasets is not listed in the table because the initialisation is this case was done manually. This is because targets were already present in the scene when the sequence capture was initiated. The tracker performed well for the first 3 sequences but in the PETS 2002 sequence, 2 out of the 9 tracks were not detected and 1 track was falsely initialised. The initialisation failures are due to two new targets arriving at the scene at the same time, one occluding the other. The one false alarm results from a new target being detected at the wrong entry point. In the PETS 2004 sequence one of the tracks was not detected. This is because the new target enters at a point in the image where segmentation quality is poor. These failures identify the initialisation aspects of the tracker that need further refinement.

5.2.2 Tracking Trough Occlusion

A lost track index LT is used to evaluate how well the tracker is able to track through occlusions. This index is defined as follows:

$$LT = \frac{\text{Lost Tracks}}{\text{Total Tracks}}. \tag{5.4}$$



(a) 2-Cam Dobrech sequence frame 50 — occlusion

(b) 1-Cam Jammie sequence frame 59 — just before occlusion.

(c) 1-Cam Jammie sequence frame 69 — just after occlusion.

Figure 5.1: Tracking through occlusion.

(a) 4-Cam Dip sequence - t=21.5s.

(b) Colourful People sequence - frame 140.

(c) Colourful People sequence - frame 448.

(d) PETS 2002 sequence - frame 673 (1346)

(e) PETS 2004 sequence - frame 290 (580).

Figure 5.2: More tracking illustrations.

Dataset	NE	PC	TDR_{av}	FAR_{av}
2-Cam Debtech	0.10	0.14	1.00	0.00
1-Cam Jammie	0.70	0.47	1.00	0.00
Colourful People	0.30	0.55	1.00	0.00
PETS 2002	0.60	0.64	**0.78**	**0.10**
PETS 2004	0.60	0.44	**0.75**	0.00

Table 5.2: Track initialisation performance values.

Table 5.3 gives the LT results for all 6 datasets as well as some of complexity metrics defined in the previous section. Two tracks were lost or confused during the processing the PETS 2002 sequence. This sequence was given quite high OC and CS complexity ratings due to the high number of occlusions occurring between targets of very similar colour compositions. The difficulties presented by this sequence give an indication of the limits of the tracking method presented in this thesis. Further improvements to the way targets are modelled, both from an appearance and behaviour point of view, would be required to overcome these difficulties.

Dataset	OC	CS	PC	LT
2-Cam Debtech	0.15	0.00	0.14	0.0
4-Cam Dip	0.46	0.78	0.62	0.0
1-Cam Jammie	0.18	0.96	0.47	0.0
Colourful People	0.85	0.68	0.55	0.0
PETS 2002	0.27	0.85	0.64	**0.2**
PETS 2004	0.03	0.75	0.44	0.0

Table 5.3: Occlusion handling performance of tracker.

5.2.3 Tracking Error

The average image-view tracking error per target, per frame OTE_{IV} (%) is calculated as follows:

$$OTE_I = \frac{100}{TF\sqrt{r^2+c^2}} \sum_k \sqrt{(i_k - \bar{i}_k)^2 + (j_k - \bar{j}_k)^2}. \tag{5.5}$$

where r and c are the row and column dimensions of the image, $(\tilde{j}_k, \tilde{i}_k)$ are the ground truth image coordinates obtained manually and $(\hat{j}_k, \hat{i}_k)$ are the estimated image coordinates of the targets at frame k. The image error is given as it facilitates the comparison of performance of datasets with large fields of view and ones with smaller fields of view. Ground truth in all cases was obtained by manually specifying the position of each of the targets present in the scene for a selected number of sequence frames using point and click method. The average world-view tracking error per target, per frame OTE_W is calculated in a similar manner.

$$OTE_W = \frac{1}{TF} \sum_k \sqrt{(\hat{x}_k - \tilde{x}_k)^2 + (\hat{y}_k - \tilde{y}_k)^2}. \tag{5.6}$$

where TF is the total number of frames for which ground truth was defined (this was done by projecting the image ground truth coordinates), $(\tilde{x}_k, \tilde{y}_k)$ are the ground truth world-view coordinates and $(\hat{x}_k, \hat{y}_k)$ are the estimated world-view coordinates.

Table 5.4 gives the world-view and image view average tracking errors for the 6 datasets and the calibration method used and the perceptual complexity. Here again the tracking performance is lowest for the PETS 2002 sequence both in the image-view and world view. The the high world-view tracking error for the 1-Cam Jammie sequence is explained by the relatively large field of view over which targets were tracked. The image-view error for this sequence is comparable to the rest of the datasets.

Dataset	Calibration	PC	$OTE_W(m)$	$OTE_I(\%)$
2-Cam Delstech	Tsai	0.14	0.23	2.2
4-Cam Dip	Tsai	0.62	0.20	2.6
1-Cam Jammie	Automatic	0.47	1.02	4.8
Colourful People	Tsai	0.55	0.69	4.8
PETS 2002	Manual	0.64	1.10	8.3
PETS 2004	Tsai	0.44	0.26	1.6

Table 5.4: Tracking error summary.

5.2.4 Track Termination

The *Track Termination* aspect is evaluated in a similar manner to the *Initialisation* aspect using a detection rate TDR_{out} and a false alarm rate FAR_{out} defined as follows:

$$TDR_{out} = \frac{\text{Total True Positives}}{\text{Total number of exits}} \tag{5.7}$$

and

$$FAR_{out} = \frac{\text{Total False Positives}}{\text{Total number of exits}} \tag{5.8}$$

Table 5.5 gives the TDR_{out} and FAR_{out} results for 5 datasets. The number of entries NE and perceptual complexity PC metrics are also given. Again, the 4-Cam DIP sequence datasets is not listed in the table because the termination of tracks were specified manually. This aspect of the system works very well on the datasets used. Two exits were falsely detected in PETS2002 sequence. This is due again to occlusions occurring near exit points. The proposed approach for track termination needs further refinement to cope with this scenario.

Dataset	NE	PC	TDR_{out}	FAR_{out}
2-Cam Debtech	0.10	0.14	1.0	0.0
1-Cam Jammie	0.70	0.47	1.0	0.0
Colourful People	0.30	0.55	1.0	0.0
PETS 2002	0.60	0.64	1.0	**0.2**
PETS 2004	0.60	0.44	1.0	0.0

Table 5.5: Tracking system's ability to detect target exit.

5.3 Tracking Performance and Segmentation Quality

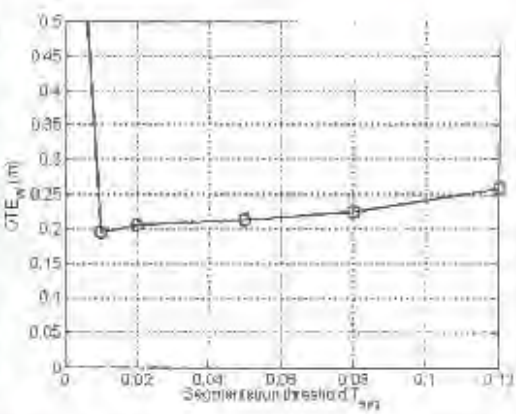
To investigate how the segmentation quality affects tracking performance we tested the tracking system (with manual initialisation) on a sequence of 150 frames from the '2-Cam Debtech' dataset for varying segmentation threshold values T_{seg}. The sequence contains one instance of simultaneous complete occlusion of the target in one view and partial occlusion in the other. Figures 5.3(a), 5.3(b) and 5.3(c) show images for T_{seg} values of 0.12 (over-segmented), 0.08 (segmented) and 0.02 (under-segmented).

The chart in figure 5.3(d) shows a plot of the tracking error OTE_W for different values of T_{seg}. The tracking error remains fairly constant, improving slightly as we reduce the segmentation threshold but rises significantly (track is lost) if we make the threshold zero. We can ascribe the slight improvement as we lower the threshold value to the fact that more information is available for the matching process when the image is under-segmented. The loss of robustness when no segmentation is performed is explained by the increased difficulty in the're-acquisition' of the target after the occlusion occurred.

(a) T_{seg}=0.12 — Over-segmented image.

(b) T_{seg}=0.08 — Segmented image.

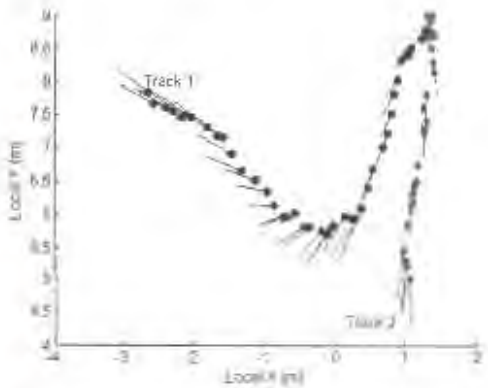
(c) T_{seg}=0.02 — Under-segmented image



(d) Chart of tracking error for different segmentation threshold values.

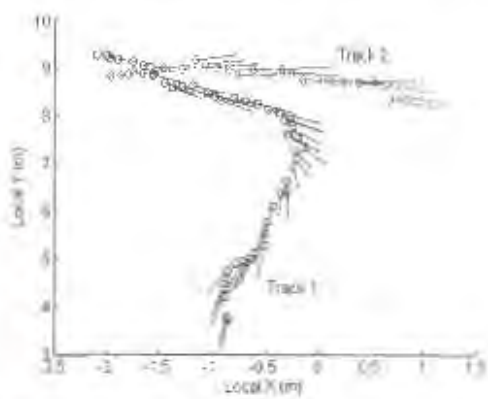
Figure 5.3: Tracking performance and segmentation quality.

(a) Tracking in camera 1 local coordinate system

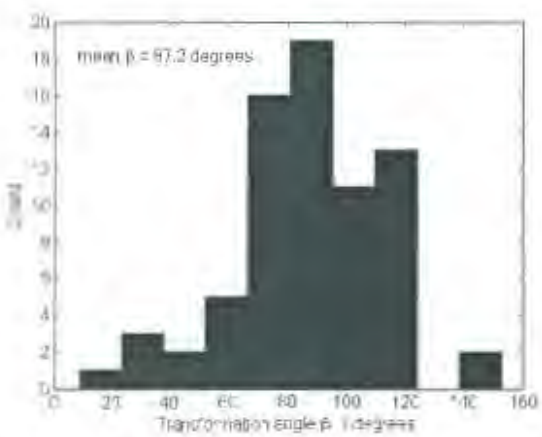


(b) Selected tracks (position and velocity) for tracking camera 1

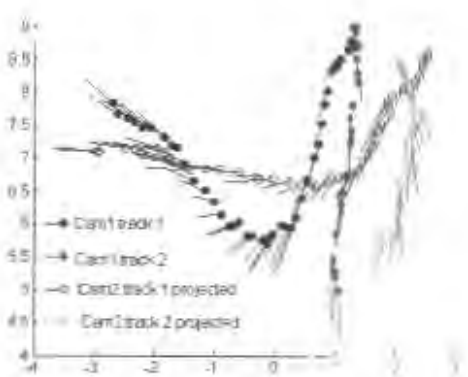
(c) Tracking in camera 2 local coordinate system



(d) Selected tracks (position and velocity) for tracking in camera 2 local coordinate system



(e) Histogram plot of transformation angle inferred from tracks



(f) Selected tracks from camera 2 local coord. system projected onto camera 1 local coord. system

Figure 4.7: Obtaining local to local ground plane transformation using automatically obtained tracks

We make the assumption that only one target is tracked by the two cameras at a time during the observation process and the observations are temporally synchronised. In other words correspondences of the data from cameras c_1 and c_2 are known. After collecting a sufficient number of observations (say from $t = 0$ to $t = T$), we find the angle β by taking the mean all the observed values. In other words

$$\beta = mean\{\beta(t)\}_{t=0,1,\ldots,T}. \tag{4.13}$$

The translation $\mathbf{t}_g$ is then calculated as follows:

$$\mathbf{t}_g = mean\{\mathbf{t}_g(t)\}_{t=0,1,\ldots,T}. \tag{4.14}$$

Let $[\mathbf{R}_1 \quad \mathbf{t}_1]$ and $[\mathbf{R}_2 \quad \mathbf{t}_2]$ be the local-ground-plane transformations for cameras c_1 and c_2. Once β and T are calculated, $[\mathbf{R}_1 \quad \mathbf{t}_1]$ can be transformed so that tracking takes place in a common ground coordinate system:

$$[\mathbf{R}_1^{new} \quad \mathbf{t}_1^{new}] = \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_g(\beta) & \mathbf{t}_g \\ 0 & 1 \end{bmatrix} \tag{4.15}$$

Figure 4.7 illustrates the process described in this section. Figure (a) shows frame 125 from the 2-Cam Debtech sequence where one target is being tracked. Figure (b) shows tracker position and velocity estimates for sequence frames where the target is present in both camera views. The estimates have been grouped into 2 separate tracks, coloured differently to help with visualisation. Figures (c) and (d) show the same as (a) and (b) but for camera 2. Figure (e) shows a histogram of $\beta(t)$ values that were obtained using equation 4.11. Figure (f) shows the tracks obtained from each view plotted on the same ground plane coordinate system.

4.2.2 Manual Approach

An alternative approach to using monocular tracking is to manually select points and vectors present in both views. This yields more accurate results in instances where poor monocular tracking results are available. Figure 4.8 shows how a collection of 8 manually selected points and vectors were used to obtain the transformation between 2 local ground planes for the 2-Cam Debtech sequence.

5.4 Tracking Performance and Image Size

Real-time processing is not achieved at the current stage of development of the tracking system. The system tracks a single person, from two views at 2.4 frames a second in a Matlab implementation. This result is achieved on a Pentium 2.4 GHz with an image size of 384×288 with and ellipses containing 1000-4000 pixels (roughly 1-4% of the total image area). The tracking process bottleneck is the histogram representation of the ellipse-shaped samples. As we increase the number of views and/or number of subjects to be tracked, the processing speed goes down quite dramatically even though no further time is spent on segmentation. Some preliminary experimentation was done using down-sampled images. Table 5.6 shows the tracking error and the speed performance (in frames per second) for the 2-Cam Debtech sequence. It appears that effective tracking could be performed at much higher frame rates using quarter and 16^{th}-sized images without considerably compromising the tracking performance.

	Full image	**1/4 image**	**1/16 image**
$OTE_W(m)$	0.98	1.20	1.76
fps	2.4	4.8	7.0

Table 5.6: Comparison of calibration results for 1-Cam Jammie dataset.

5.5 Assessment of Calibration methods

Two datasets were used to evaluate and compare the calibration methods presented in the previous chapter. Table 5.7 gives the calibration parameters obtained for the two cameras used to capture the 1-Cam Debtech sequence. Recall that f_i^α is the focal length pixel width ratio, t_z is the height of the camera above the ground, ϕ is the pitch angle (where a $\phi = 0$ means the camera is pointing straight up), β is the angle between the local ground plane coordinate systems and $|\mathrm{t}_g|$ is the distance between the origins of the local ground plane coordinate systems. In this sequence, the height of the cameras above the ground cannot be found using the automatic method as only one target is present. Note that the automatic method gives quite poor results in comparison to the other methods for

this dataset. This is due mainly to the poor segmentation quality achieved during the recording of (i, h) observations. Also the limited range of data points obtained, especially for camera 1, where most of the target movement is transverse. Also shown in the table is the tracking accuracy when each of the calibration methods was used. As expected, the tracker performs badly when cameras are poorly calibrated.

	Actual	**Automatic**	**Manual**	**Tsai**
Camera 1: f_i^α	281	318	245	275
$\mathbf{t}_z(m)$	2.40	—	2.40	2.40
ϕ	73°	80°	76°	74°
Camera 2: f_i^α	276	305	249	271
$\mathbf{t}_z(m)$	2.40	—	2.40	2.40
ϕ	70°	73°	77°	79°
β	84.0°	87.2°	85.5°	85.0°
$\lvert\mathbf{t}_g\rvert(m)$	6.2	7.9	5.5	6.0
$OTE_W(m)$	0.19	1.37	0.52	0.23
$OTE_I(\%)$	2.2	9.4	4.5	2.9

Table 5.7: Comparison of calibration results for 2-Cam Debtech dataset.

Calibration parameters obtained for the camera used in the 1-Cam Jammie sequence are shown Table 5.8. Calibration data was obtained by filming each of the 3 targets separately. Segmentation in this case was good and a suitable range of (i, h) observations was obtained. Hence the calibration results obtained using the automatic method are very good.

	Actual	**Automatic**	**Tsai**
f_i^α	474	470	483
$\mathbf{t}_z(m)$	2.68	2.65	2.79
ϕ	70°	72°	72°
$OTE_W(m)$	0.98	1.20	1.76
$OTE_I(\%)$	4.2	4.8	7.1

Table 5.8: Comparison of calibration results for 1-Cam Jammie dataset.

Chapter 6

Conclusions

In this chapter we summarise the tracking system presented in this thesis. We discuss the strengths and the weaknesses and potential further improvements of each aspect of the system. We also briefly discuss our findings on calibration methods suitable for person tracking applications.

6.1 The Tracking System

The way tracking is performed in our system can be summarised as follows.

- The world-view shape of the targets is assumed to be Ellipsoidal.
- The colour information on each target is parameterised by a 4-dimensional RGB-height histogram.
- Each time a new frame is received by a tracking client (associated with each camera) it is segmented into foreground and background regions and the following steps are executed:
  - The world-view state of all targets being tracked is fetched from the server. It is used to predict world-view position using a simple constant acceleration dynamic model.

- The predicted world-view position is then used together with camera calibration information to match targets within the foreground regions of the image.
- The match results are then used to update the world-view target states using the Extended Kalman Filter formulation.

World-view Tracking

World-view tracking as opposed to image-view tracking offers a number of advantages. A dynamic model with various physical constraints is more sensible, constraints on the expected shape and size of targets in the camera views are more easily imposed and the definition of a common coordinate system in the case of multi-camera tracking configurations is made simpler.

Client Server Architecture

The modular client-server architecture used for sharing target data is very versatile. The system can easily be expanded to large-scale implementations involving 100s of cameras without the need for complicated rule-based system with numerous data interconnections. Tracking clients do not interact directly with each other and so need not be synchronised. Should one client/camera become temporarily unavailable, the system can still function provided there is enough overlap between views.

Filtering

The Extended Kalman Filter lends itself very well to asynchronous and synchronous observations from multiple cameras. This formulation allows quite complex fusion of prior knowledge and observations from different devices without the use of any complicated rule-based architecture. As shown in the previous chapter, the tracker performs very well when scenes are not overly cluttered with targets of similar colour composition. More robust tracking in cluttered scenes would require a more sophisticated way of representing targets, modelling their behaviour and handling occlusions. More general noise assumptions when formulating the tracking problem, using for example a multiple hypothesis Kalman

Filter, or alternatively a particle filter could be explored and potentially be implemented without major modifications of the current system to improve the filtering process.

Shape Representation of Targets

The ellipsoid model is simple and suitable for representing the shape of a person of average size standing or walking. The sitting position is not currently handled explicitly but on the sequences tested (the 4-Cam DIP sequence includes the tracking of a person initially seated at a desk), does not seem to cause difficulties. Should the tracking of other targets whose size and shape differ considerably from that of a standing human (wheelchair, car, animal) new shape models would have to be defined. This would then also require some further heuristics for classifying targets when they enter a monitored scene.

Colour Representation and Matching

The colour-height histogram representation is promising. It is able to differentiate between different targets quite successfully at a relatively low computational cost, even in cases of very poorly segmented images. However, it needs further refinements in order to be used more effectively in the case of ceiling cameras. The Bhattacharyya Coefficient seems to be suitable for comparisons of histogram models. Potential improvements could be obtained by considering more robust methods such as the *EarthMover's Distance* in [36].

Foreground/background Segmentation

We cannot avoid the segmentation step as the initialisation process depends on it and the robustness of the tracker is affected by it. However, very good tracking performance is achieved even with very poor segmentation. Thus, only simple segmentation methods need to be considered for the tracking system presented here.

Initialisation of Tracks

The simplistic approach used for initialising tracks in this system proves quite promising. Track initialisation fails when two targets enter the scene at the same time. Explicit handling of this event seems to be the only way around this problem. The current implementation also does not attempt to recover a lost track. Re-initialisation is something that would still need to be dealt with.

Termination of Tracks

The handling of targets leaving the scene seems satisfactory for the test datasets considered. Again, this aspect can be made more robust by including further heuristics.

Processing Speed

The system tracks a single person in one view at roughly 5 frames a second. Some preliminary experimentation was done using down-sampled images and it appears that effective tracking could be performed at higher frame rates using smaller images without considerably compromising the tracking performance. No real effort has been made so far to optimise the implementation, which is at this stage completely done in Matlab. Some effort needs to be put into evaluating how much information the tracker actually does need from the images for robust tracking.

6.2 Calibration Methods suited to Person Tracking

The tracking system relies heavily on good calibration of the cameras to a common coordinate system. As demonstrated in the previous chapter, better calibration leads to better tracking performance and poor calibration makes the fusing of multiple observations difficult. The calibration requirement is a considerable restriction especially in large-scale implementations. We address this by exploring a few practical methods of obtaining the calibration information without the need for time-consuming manual measurement of calibration points.

The automatic method

A 2-stage automatic calibration method based on a method by Jones et al. [16] was implemented and tested. The first stage recovers the local ground plane calibration parameters by observing size variations of images of targets as they move towards and away from each of the cameras. The second stage finds the correspondence between each of the local ground plane coordinate systems by matching tracks observed in each view. The method proves to be very useful, sometimes yielding better results than Tsai's method, but is however not suitable for all camera configurations:

- *Lens Distortion*: The method assumes a simplified distortion-free camera model. Should cameras to be calibrated have high levels of distortion, the method will not yield accurate results. However, it would not be too difficult to incorporate prior knowledge about the extent of distortion in the calibration process.

- *Camera Pose*: The method assumes that the height variation is linear. For shallow pitch angles this assumption is valid. However, the steeper the angle the more the variation deviates from linearity, and so the less accurate the calibration becomes. The method makes use of only one vanishing point out of a possible 3. This only allows the recovery of the pitch angle, so cameras with substantial roll and yaw cannot be calibrated using this method. A method by Zhao [24] uses similar principles method and claims to recover all 3 vanishing points. However, this was noticed too late to be included in this work and can only be recommended for future investigation.

- *Segmentation*: The method relies quite heavily on good segmentation to record the height variations. Observations should not be made in complex sequences where poor segmentation is achieved.

The manual method

Some of the constraints of the automatic method can be overcome by allowing the operator to intervene manually. A Matlab interface was developed to facilitate manual recording of height observations as well as fine tuning of the calibration parameters. Better calibration can be achieved if intrinsic parameters are found in advance (before the cameras are

installed).

The Method by Tsai

Camera calibration using visual cues found in the scene is sometimes possible and so should also be considered. An method based on one by Tsai [43] that makes use of co-planar calibration points determined by patterns found in the scene such as floor tiles of known size is suitable as this approach also foregoes the time consuming manual laying and measuring of markers.

Appendix A

Tracking System Parameters

The system was designed to be as general as possible but it is impossible to perform tracking without specifying certain parameters that do affect the tracking performance. These parameters are summarised as follows:

- *Ellipsoid Parameters* (r_x, r_y, r_z)
  - r_z, the ellipsoid semi-major axis length is set to 0.90 metres.
  - r_x and r_y are set as a fraction of the height to $\frac{r_z}{3.5}$.
- *Dynamic model noise covariance matrix* $(\mathbf{N}_x)$.
  This sets the uncertainty of the dynamic model. No deterministic approach to select this parameter exists so it can only be determined through experimentation. No attempt was made to find the optimal N because it would be different for each tracking scenario. However it was found that

$$N_x = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\dot{x}} & \sigma_{x\dot{y}} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\dot{x}} & \sigma_{y\dot{y}} \\ \sigma_{\dot{x}x} & \sigma_{\dot{x}y} & \sigma_{\dot{x}}^2 & \sigma_{\dot{x}\dot{y}} \\ \sigma_{\dot{y}x} & \sigma_{\dot{y}y} & \sigma_{\dot{y}\dot{x}} & \sigma_{\dot{y}}^2 \end{bmatrix} = \begin{bmatrix} 0.03 & 0 & 0 & 0 \\ 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0.03 & 0 \\ 0 & 0 & 0 & 0.03 \end{bmatrix}$$

  works well on most of test the sequences proposed in this thesis.
- *Measurement noise covariance matrix*$(\mathbf{N}_y)$.
  This sets the uncertainty of our measurement in image space. Again, there is no

other way to set this parameter other than by experimentation. It was found that

$$\mathbf{N}_y = \begin{bmatrix} \sigma_j^2 & \sigma_{ji} \\ \sigma_{ij} & \sigma_i^2 \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

works well on all test sequences.

- *Segmentation.*
  - T_{seg}, the threshold on the difference image, is set to 0.08.
- *Colour models.*
  - n_h, the height histogram bins is set to 6.
  - n_R, n_G, n_B, the number of RGB colour bins, are each set to 10.
  - λ_c, the colour model learning rate parameter, is set to 0.0001.
  - n_s, the number of samples taken per tracked subject per frame, is set to 20.
- *Initialisation of track.* T_{init}, the ratio of foreground pixels in ellipse to total pixels in ellipse, is set to 0.6.
- *Termination of track.*
  - $T_{d_{min}}$, the distance of the tracked subject to the closest exit point is set to 0.5m.
  - T_t, the time for which average of ρ must be taken is set to 2 sec.
  - T_ρ, the threshold value for $\hat{\rho}$, is set to 0.15.

Ideally, no 'tuning' should be required from the values given above. As mentioned before, no extensive experimentation was performed to find how much these parameters change the performance of the tracking; they were tuned so that the tracker worked with all test sequences. Thorough experimentation is left as future work.

Appendix B

Tsai's Camera Calibration Method

Introduction

This method requires the image $\mathbf{m} = (j, i)^T$ and world coordinates $\mathbf{M} = (X_w, Y_w, Z_w)^T$ of a minimum number of points (*calibration points*) well scattered around the camera view. For non-coplanar points, the method requires a minimum of 7 points and the case of coplanar points a minimum of 5 points. The method for co-planar points offers the big advantage that it is often easier to obtain world coordinates of points lying in the same plane. This method is particularly useful when one does not know the exact world coordinates of the calibration points and the observed scene has some strong coplanar visual cues such as floor tiles. The only parameter that needs to be guessed then is the length and width of the floor tiles. The disadvantage is that it is difficult to get points that cover large parts of the image and so calibration results tend to be poorer.

The calibration process is broken down into the following steps.

1. Setup linear equations relating $\mathbf{m}_{hu}$ to M and camera parameters (except for κ).
2. Compute the magnitude of t_y.
3. Find the sign of t_y.
4. Determine s and t_x.
5. Compute R.

6. Compute f_i^α and t_z.

7. Optimiser run to estimate κ, refine t_z and f_i^α and $(j_0, i_0)^T$.

Linear equations relating $\mathbf{m}_u$ to M and camera parameters (except for κ)

The undistorted projected image points $\mathbf{m}_{hu}$ can be expressed in terms of the projection matrix $\mathbf{P} = \mathbf{S}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$ and the world coordinates M using

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} sf_i^\alpha & 0 & j_0' \\ 0 & f_i^\alpha & i_0' \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \tag{B.1}$$

where (j_0', i_0') is the estimated optical centre $(\frac{c}{2}, \frac{r}{2})$, c and r the number of columns and rows of pixels in the image.
The undistorted image coordinates $\mathbf{m}_u$ can then be expressed using (2.3) to give:

$$j_u = \frac{x_c}{z_c} = sf_i^\alpha \frac{(r_1 X_w + r_2 Y_w + r_3 Z_w + t_x)}{(r_7 X_w + r_8 Y_w + r_9 Z_w + t_z)} + j_0 \tag{B.2}$$

$$i_u = \frac{y_c}{z_c} = f_i^\alpha \frac{(r_4 X_w + r_5 Y_w + r_6 Z_w + t_y)}{(r_7 X_w + r_8 Y_w + r_9 Z_w + t_z)} + j_0 \tag{B.3}$$

By letting $(j_d, i_d)^T = (j_u - j_0, i_u - i_0)^T$ and equating the denominators of (B.2) and (B.3) one obtains the equation:

$$j_d(r_4 X_w + r_5 Y_w + r_6 Z_w + t_y) = si_d(r_1 X_w + r_2 Y_w + r_3 Z_w + t_x) \tag{B.4}$$

For n calibration points we can set up n linear equations with
$(a_1, a_2, ...a_7) = (\frac{sr_1}{t_y}, \frac{sr_2}{t_y}, \frac{sr_3}{t_y}, \frac{t_x}{t_y}, \frac{r_4}{t_y}, \frac{r_5}{t_y}, \frac{r_6}{t_y})$ as unknowns, by dividing B.4 through by t_y.

$$\begin{bmatrix} i_{d1}X_{w1} & i_{d1}Y_{w1} & i_{d1}Z_{w1} & i_{d1} & j_{d1}X_{w1} & j_{d1}Y_{w1} & j_{d1}Z_{w1} \\ i_{d2}X_{w2} & i_{d2}Y_{w2} & i_{d2}Z_{w2} & i_{d2} & j_{d2}X_{w2} & j_{d2}Y_{w2} & j_{d2}Z_{w2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ i_{dn}X_{wn} & i_{dn}Y_{wn} & i_{dn}Z_{wn} & i_{dn} & j_{dn}X_{wn} & j_{dn}Y_{wn} & j_{dn}Z_{wn} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_7 \end{bmatrix} = \begin{bmatrix} j_{d1} \\ j_{d2} \\ \vdots \\ j_{dn} \end{bmatrix} \tag{B.5}$$

Given enough calibration points we can use this set of linear equations to compute $(a_1, a_2, \ldots a_7)$. A solution to the above is not always guaranteed, especially in the case of high distortion

cameras. One way to overcome this problem is to remove calibration points that are furthest away from the centre of the image for this step, and re-include them in the optimiser runs where distortion is taken into account. Alternatively, one could guess a value for distortion and 'undistort' the image coordinates before solving (B.5).
For the co-planar case Z_w is assumed to be 0. This causes some of the elements of the above matrix to fall away and only leaves us with 5 unknowns.

The magnitude of t_y

Since $r_4^2 + r_5^2 + r_6^2 = 1$, t_y for the non coplanar case can be calculated as follows

$$t_y = \pm \frac{1}{\sqrt{a_5^2 + a_6^2 + a_7^2}}, \tag{B.6}$$

and for the coplanar case:

$$t_y = \pm \sqrt{\frac{\mathcal{B} - \sqrt{\mathcal{B}^2 - 4\mathcal{A}}}{2\mathcal{A}}}, \tag{B.7}$$

where $\mathcal{A} = (a_1 a_5 - a_2 a_4)^2$ and $\mathcal{B} = (a_1^2 + a_2^2 + a_4^2 + a_5^2)$.

The sign of t_y

The sign of t_y is found by projecting a point the coordinates of a point whose image is in a known quadrant of the frame assuming t_y to be positive. If the point is projected onto the expected quadrant then $sign(t_y) = +1$ otherwise $sign(t_y) = -1$.

s and t_x

In the coplanar case, s is initially assumed to be 1 or to any other better approximation of its correct value. In the case of internal parameters already known, there is of course no need to compute s. In the non coplanar case, since $r_1^2 + r_2^2 + r_3^2 = 1$,

$$s = \sqrt{a_1^2 + a_2^2 + a_3^2}|t_y|. \tag{B.8}$$

t_x is simply

$$t_x = a_3 t_y \tag{B.9}$$

Rotation matrix R

For the non coplanar case, the first two rows of **R** are calculated as follows:

$$\begin{aligned} r_{1\ldots3} &= \frac{a_{1\ldots3}t_y}{s} \\ r_{4\ldots6} &= a_{4\ldots6}t_y \end{aligned} \tag{B.10}$$

For the coplanar case $r_{1,2,4,5}$ are calculated as above and $r_3, 6$ are calculated as follows:

$$\begin{aligned} r_3 &= \sqrt{1 - r_1^2 - r_2^2} \\ r_6 &= \sqrt{1 - r_4^2 - r_5^2} \end{aligned} \tag{B.11}$$

In both cases the third row of **R** is determined from the outer or cross product of the first two rows using the orthonormal property of a rotation matrix.
For the coplanar case, if f_i^α calculated in the next section (equation B.12) is negative, then the signs of $r_{3,6,7,8}$ must be changed.

f_i^α and t_z

f_i^α and t_z are obtained as the solution of the following linear equations.

$$\begin{bmatrix} r_4X_{w1} + r_5Y_{w1} + r_6Z_{w1} + t_y & -i_{d1} \\ r_4X_{w2} + r_5Y_{w2} + r_6Z_{w2} + t_y & -i_{d2} \\ \vdots & \vdots \\ r_4X_{wn} + r_5Y_{wn} + r_6Z_{wn} + t_y & -i_{dn} \end{bmatrix} \cdot \begin{bmatrix} f_i^\alpha \\ t_z \end{bmatrix} = \begin{bmatrix} (r_7X_{w1} + r_8Y_{w1} + r_9Z_{w1})i_{d1} \\ (r_7X_{w2} + r_8Y_{w2} + r_9Z_{w2})i_{d2} \\ \vdots \\ (r_7X_{wn} + r_8Y_{wn} + r_9Z_{wn})i_{dn} \end{bmatrix} \tag{B.12}$$

In the case where intrinsic parameters are already known, we have a system of linear equations with only one unknown, namely T_z.

Optimiser runs

At this stage we are left with distortion coefficients and the optical the centre, as well as s in the coplanar case, to determine. This could be done as suggested by Tsai [43] using one run of any standard optimising scheme. Better results are achieved using 3 separate

runs using a multi-dimensional unconstrained algorithm such as the Levenberg-Marquadt with error vector as follows:

$$Error = \begin{bmatrix} j_1 & j_2 & \cdots & j_n \\ i_1 & i_2 & \cdots & i_n \end{bmatrix} - \begin{bmatrix} j_{proj1} & j_{proj2} & \cdots & j_{projn} \\ i_{proj1} & i_{proj2} & \cdots & i_{projn} \end{bmatrix}. \tag{B.13}$$

In the first run f_i^a, t_z and κ are the only inputs to the optimiser. In the second run all the parameters estimated so far are refined. It is important to note that the nine elements of $\mathbf{R}$ cannot be directly used as inputs to the optimiser since $\mathbf{R}$ has to retain its orthogonality. Hence $\mathbf{R}$ must be parameterised either using Euler angles or quaternions. In the third and final run the image centre, $(j_0, i_0)^T$ is also included.

Bibliography

[1] N. Ayache and O. Faugeras. Maintaining representations of the environment of a mobile robot. *Int. Journal of Robotics and Automation*, 5(6):804–819, 1989.

[2] M. Bichsel. Segmenting simply-connected moving objects in a static scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16.

[3] S. Birchfield. An introduction to projective geometry. March 1998.

[4] J. Black and T. J. Ellis. Multi camera image measurement and correspondence. *The Journal of the International Measurement Confederation*, 35(1):61–71, 2002.

[5] B. Bose and E. Grimson. Ground plane rectification by tracking moving objects. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, October 2003.

[6] Y. Chen, Y. Rui, and T. Hunag. Jpdaf-based hmm for real-time contour tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii*, volume 1.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(5):564–575, 2003.

[8] A. Criminisi, I. Reid, and A. Zisserman. Single view geometry. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[9] R. Cucchiara, C. Grana, and A. Prati. Detecting moving objects and their shadows. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, June 2002.

[10] G. Forbes. The automatic detection of patterns in peoples movements. Master's thesis, University of Cape Town, March 2002.

[11] D. Forsyth and J. Ponce. *Computer Vision A Modern Approach.* Prentice Hall, 2003.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry.* Cambridge, 2 edition, 2003.

[13] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[14] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *IEEE International Conference on Computer Vision*, volume 2, pages 34–41, Vancouver, British Columbia, Canada, 2001. IEEE Computer Society, Los Alamos CA.

[15] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. In *CVPR*, volume 1, pages 415–422, 2001.

[16] G. A. Jones, J. Renno, and P. Remagnino. Auto-calibration in multiple-camera surveillance environments. In *IEEE International Workshop on Visual Surveillance*, 2002.

[17] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *SPIE*, volume 3068, pages 182–193, 1997.

[18] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15.

[19] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–46, 1960.

[20] J. Kang, I. Cohen, and G. Medioni. Soccer player tracking across uncalibrated camera streams. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, October 2003.

[21] S. Kay. *Fundamentals of Statistical Signal Processing Estimation Theory.* Prentice Hall, 1993.

[22] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 2000.

[23] M. Louw, F. Nicolls, and G. de Jager. Multi-blob particle filter. In *Thirteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2002.

[24] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *International Conference on Pattern Recognition*, 2002.

[25] Maybeck. *An Stochastic Models, Estimation and Control.* New York Academic Press, 1979.

[26] R. Merwe, A. Doucet, N. Freitas, and E. Wan. The unscented particle filter. Technical report, Cambridge University Engineering Department, 2000.

[27] F. Nicolls. Multiple camera person tracking: a world-centric formulation. Technical report, University Cape Town, June 2002.

[28] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 2002.

[29] J. Pers, M. Bon, and S. Kovaix. Errors and mistakes in automated player tracking. In *Sixth Computer Vision Winter Workshop*, 2001.

[30] J. H. Piater and J. L. Crowley. Multi-modal tracking of interacting targets using gaussian approximations. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, December 2001.

[31] M. Pollefeys. *3D Modeling from Images.* Katholieke Universiteit Leuven, June 2002.

[32] M. Price, F. Nicolls, and G. de Jager. Using colour features for video-based tracking of people in a multi-camera environment. In *Fourteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2003.

[33] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, volume 77.

[34] J. R. Renno, P. Remagnino, and G. A. Jones. Learning surveillance tracking models for the self-calibrated ground plane. *Acta Automatica Sinica - Special Issue on Visual Surveillance of Dynamic Scenes*, 2003.

[35] F. Robertson. Segmentation in image sequences: Tracking human figures in motion. Master's thesis, University of Cape Town, June 2001.

[36] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *IEEE International Conference on Computer Vision*, 1998.

[37] N. T. Siebel and S. J. Maybank. Real-time tracking of pedestrians and vehicles. In *IEEE International Workshop on Visual Surveillance*, 2001.

[38] T. Sodestrom. *Discrete-time Stochastic Systems.* Springer, 2 edition, 2002.

[39] C. Stauffer, K. Tieu, and L. Lee. Robust automated planar normalization of tracking data. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, October 2003.

[40] B. Stenger, P. Mendonca, and R. Cipolla. Model-based 3-d tracking of an articulated hand. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2.

[41] J. Swain, M and H. Ballard, D. Colour indexing. *International Journal of Computer Vision*, 7.

[42] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, 1998.

[43] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, pages 323–344, 1987.

[44] D. R. Wehner. *High Resolution Radar.* Artech House, Norwood, MA 02062, 1987.

[45] G. Welch and G. Bishop. *An Introduction to the Kalman Filter.* University of North Carolina at Chapel Hill, 2001.

[46] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 780–785, 1997.

[47] M. Xu and T. J. Ellis. Tracking occluded objects using partial observation. *Acta Automatica Sinica - Special Issue on Visual Surveillance of Dynamic Scenes*, pages 370–380, 2003.

[48] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22.

[49] T. Zhao, R. Nevatia, and F. Lv. Segmentation and tracking of multiple humans in complex situations. In *International Conference on Computer Vision and Pattern Recognition*, 2001.