CREATING A VIRTUAL SLIDE MAP FROM SPUTUM SMEAR IMAGES FOR REGION-OF-INTEREST LOCALISATION IN AUTOMATED MICROSCOPY

by

BHAVIN PATEL

A thesis submitted to the

University of Cape Town in partial fulfillment of the

requirements for the degree of

MSc (Med)

in

BIOMEDICAL ENGINEERING

Faculty of Health Sciences

University of Cape Town

October 2010

Supervisor: Assoc. Prof Tania Douglas

DECLARATION

I, …….., hereby declare that the work on which this dissertation is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

I empower the university to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Signature: ……

Date: …….

ACKNOWLEDGMENTS

I would like to express sincere gratitude to Associate Professor Tania Douglas, my supervisor, for her guidance and support during my Masters thesis.

Thanks, also, to Sriram Krishnan for providing some useful insights and making several suggestions that contributed to the outcome of the project.

I would also like to thank the members of staff of the National Health Laboratory Services at the Groote Schuur Hospital who prepared the sputum slides, and Susan Cooper for her assistance on the microscopy imaging unit in the Department of Human Biology.

Thanks also to the Rethabile Khutlang whose training data and algorithms were used for the segmentation process.

I also wish to thank my parents for tirelessly supporting, enduring and guiding me in their own magical way throughout my studies.

Finally, I would like to thank the NIH and NRF for funding the project, and the members of staff and the students in the Biomedical Engineering division for their support.

ABSTRACT

Automated microscopy for the detection of tuberculosis (TB) in sputum smears seeks to address the strain on technicians in busy TB laboratories and to achieve faster diagnosis in countries with a heavy TB burden. As a step in the development of an automated microscope, the project described here was concerned with microscope auto-positioning; this primarily involves generating a point of reference on a slide, which can be used to automatically bring desired fields on the slide to the field-of-view of the microscope for re-examination. The study was carried out using a conventional microscope and Ziehl-Neelsen (ZN) stained sputum smear slides. All images were captured at 40x magnification.

A digital replication, the virtual slide map, of an actual slide was constructed by combining the manually acquired images of the different fields of the slide. The geometric hashing scheme was found to be suitable for auto-stitching a large number of images (over 300 images) to form a virtual slide map. An object recognition algorithm, which was also based on the geometric hashing technique, was used to localise a query image (the current field-of-view) on the virtual slide map. This localised field-of-view then served as the point of reference.

The true positive (correct localisation of a query image on the virtual slide map) rate achieved by the algorithm was above 88% even for noisy query images captured at slide orientations up to 26°. The image registration error, computed as the average mean square error, was less than 14 pixel^2 (corresponding to 1.02 µm^2 and 0.001% error in an image measuring 1030 x 1300 pixels) corresponding to a root mean square registration error of 3.7 pixels. Superior image registration accuracy was obtained at the expense of time using the scale invariant feature transform (SIFT), with a image registration error of 1 pixel^2 (0.07 µm^2).

The object recognition algorithm is inherently robust to changes in slide orientation and placement, which are likely to occur in practice as it is impossible to place the slide in

exactly the same position on the microscope at different times. Moreover, the algorithm showed high tolerance to illumination changes and robustness to noise.

CONTENTS

LIST OF FIGURES

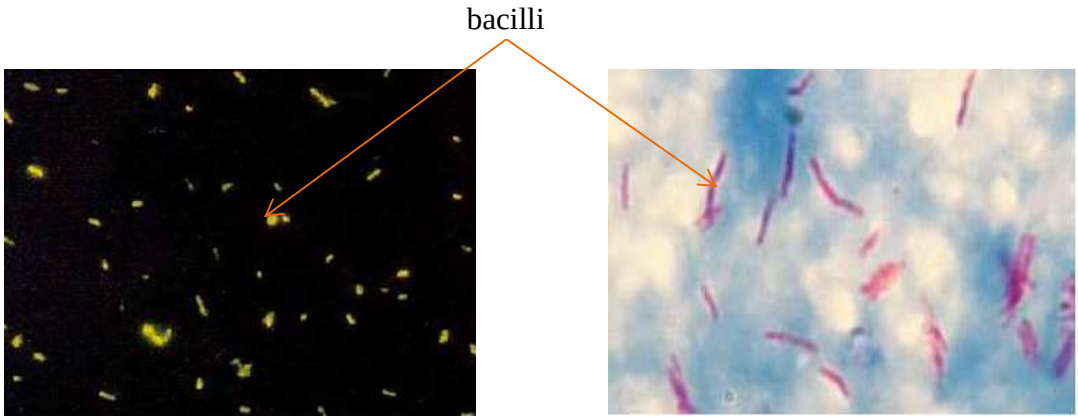
LIST OF TABLES

1. Introduction

The World Health Organisation (WHO) estimates that 1.7 million people are killed by tuberculosis (TB) every year; about three people each minute (World Health Organisation 2008). TB is an airborne infectious disease that spreads easily in densely populated areas with poor sanitation. In some countries the spread has been exacerbated by co-infection with HIV/AIDS, which weakens the immune system. Both the highest number of deaths and the highest mortality per capita are in the Sub-Saharan Africa region mainly due to the persistence of HIV/AIDS in the region (World Health Organisation 2007).

TB is caused by an infection with *Mycobacterium tuberculosis*, which is a human-residing bacterium. This bacterium is transmitted through the air in the form of microscopic droplets that are expelled by a person with active tuberculosis. This expulsion may occur during coughing, sneezing or speaking. Although the droplets dry out quickly, the bacteria remain airborne for several hours and when inhaled by a normal person, result in TB infection (Todar 2005).

Early detection of tuberculosis is critical for the initiation and monitoring of treatment and for epidemiological tracking, and therefore is essential for disease control. In high TB prevalence countries and in low- and middle- income countries, the most effective method of detecting TB is by direct sputum smear microscopy owing to the relatively cheap equipment (Steingart et al. 2006). *Mycobacterium tuberculosis* expectorated in sputum can be seen in clusters or individually in stained sputum smears under a microscope. Presently, there are two staining methods: auramine staining which is used in fluorescence microscopy; and Ziehl-Neelsen (ZN) staining which is used in bright field microscopy. In ZN-stained sputum smears, acid-fast bacilli stain red against a blue background. Bacilli have a waxy coating which absorbs the red of the Ziehl-Neelsen carbol fuchsin; the background is stained blue by a methylene blue counterstain. With the auramine staining method, the bacilli become fluorescent against a black background. Figure 1.1 shows example images of sputum samples stained by the two methods.



(a) Image of an auramine-stained sample (b) Image of a ZN-stained sample

Figure 1.1: Staining methods for sputum smears.

Although fluorescence microscopy of auramine stained sputum specimens for TB screening offers greater sensitivity than bright field microscopy of ZN stained specimens, the latter is the method of choice in developing countries due to its low cost and simplicity (Hänscheid 2008). An automated microscope is being developed in the MRC/UCT Medical Imaging Research Unit at the University of Cape Town in an effort to ease the workload of laboratory technicians screening sputum smears for TB in countries with a heavy burden of TB. An automated microscope for TB detection would include a motorised stage, an image capture unit and algorithms for auto-focusing, segmentation, classification and auto-positioning. These components would allow automatic detection of *Mycobacterium tuberculosis* bacilli in sputum smears.

To compare the image quality and bacillus detection accuracy at different settings of a microscope or across different microscopes, performance in capturing and analyzing the same field in a slide may be tested and compared. Manually locating the same field of a slide on a microscope with different settings would be a time consuming, laborious and poorly repeatable task. Microscope auto-positioning would save operators' time and provide greater accuracy and repeatability. The core component in microscope auto-positioning is finding a reliable point of reference on the slide which can then be used to bring desired fields-of-interest to the field-of-view of the microscope. This can be achieved using a virtual slide map which is constructed by combining digital images from different fields on a slide and therefore is a digital replication of the actual slide (Dee et al. 2003, Begelman et al. 2006). Localisation of the current field-of-view (FOV)

involves determining its co-ordinates on the virtual slide map and can be formulated as an object recognition task. The object recognition algorithm needs to be robust to changes in slide orientation and placement (i.e. needs to be invariant to similarity transformation), which are likely to occur in practice as it is impossible to place the slide in exactly the same position on the microscope at different times. Once the FOV is localised on the virtual slide map, it can be used as a point of reference and the coordinates of a desired field with respect to this reference can be determined on the virtual slide map and fed to the motors of the XY stage to bring that field to the field-of-view of the microspore, thus achieving microscope auto-positioning.

There are several instances in which previously scanned, imaged, and/or analyzed slides must be physically re-examined by an operator. For example, in the analysis of pathology slides, a slide is pre-examined by a cytotechnician. The cytotechnician locates and marks the regions-of-interest (ROIs) with a pen for further examination by an expert pathologist. Marking by this approach may be time consuming and inaccurate since it is difficult to mark features under high magnification. Additionally, the expert pathologist is faced with the task of manually re-positioning to these regions-of-interest. This process is usually also time consuming and laborious and hence an expert can only review and diagnose a limited number of slides per day (Begelman et al. 2006, Doerrer 2007). Microscope auto-positioning using virtual slide maps would enable the cytotechnician to easily make accurate ‘virtual marks’ on regions-of-interest on a virtual slide map which can be easily and automatically re-located, hence saving the expert pathologist time and effort. This would allow the diagnosis of a higher volume of slides per day. Figure 1.2 shows an overview of the pathologist’s examination routine (Begelman et al. 2006).

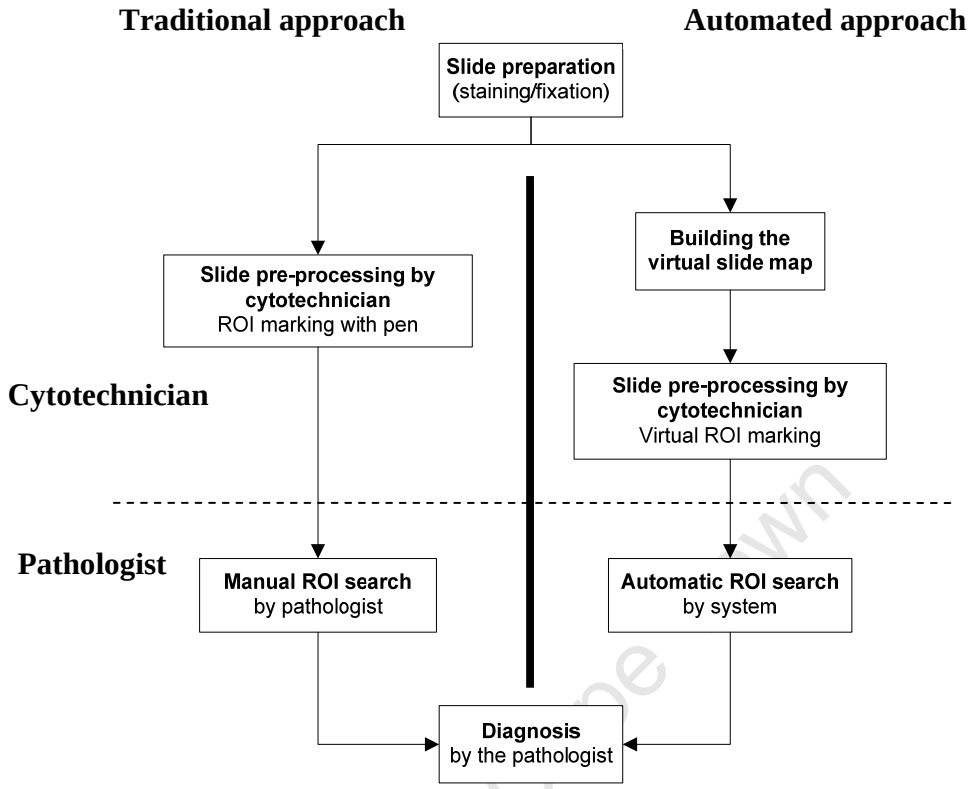


Figure 1.2: Overview of the pathologist's examination routine (Begelman et al. 2006)**.**

The aim of automation in the context of TB screening is to speed up the screening process and to prevent errors due to boredom and fatigue i.e. automation would help improve efficiency and accuracy.

The project described here represents a step in the development of an automated microscope for TB detection and was concerned with the construction of virtual slide maps and the development of object recognition algorithms for use in auto-positioning. The investigations were carried out on ZN-stained sputum smears using a conventional microscope. Since the XY stage movement in a conventional microscope is manually performed – the XY stage is not motorised – the desired fields cannot be accurately brought to the field-of-view once their coordinates relative to the point-of-reference are determined. Therefore, the project scope was limited to finding a point of reference on the slide, which is the core component in microscope auto-positioning. Although a

conventional microscope was used for this study, the ultimate goal is to use the algorithms in automated microscopy:

- to reposition to regions-of-interest of a previously scanned slide for reviewing or re-examination e.g. as shown in Figure 1.2;

- as a research tool to optimise the image quality and detection accuracy of the microscope by capturing and analysing the same field in a slide with different microscope settings;

- as a practical tool to assist an operator to reposition to a field-of-interest allowing him/her to manually verify the output of detection algorithms.

1.1 Objectives

The objectives of the study were to:

- Manually and automatically stitch images of different fields in ZN-stained sputum smear slides to construct a virtual slide map suitable for object recognition and hence for auto-positioning in bright field microscopy for TB detection.

- Develop object recognition algorithms suitable for auto-positioning in bright field microscopy for TB detection.

- Perform field-of-view localisation tasks on the virtual slide maps to validate the quality of the constructed virtual slide maps and to validate the performance of the object recognition algorithm.

The purpose of image stitching was to automatically combine the manually acquired images of the different fields of a slide in order to construct a digital replication of the slide - virtual slide map - while retaining the resolution at which the images were acquired. Object recognition algorithms were required to localise the current field-of-view on the virtual slide map so that it could be used as a point of reference allowing desired regions on the slide to be brought to the field-of-view of the microscope. Hence

both the virtual slide map and the object recognition algorithms would be used in achieving microscope auto-positioning.

1.2 Plan of Development

Chapter 2 reviews the literature on automated TB microscopy and explores methods that may be applicable to auto-positioning in TB microscopy.

Chapter 3 describes the materials used in the study.

Chapter 4 details the methods that were used for the *offline pre-processing stage* for object recognition in auto-positioning.

Chapter 5 details the methods that were used for the *online localisation stage* for auto-positioning and also presents the performance assessment techniques that were employed.

Chapter 6 details the procedure for automatically stitching TB images to construct virtual slide maps.

Chapter 7 summarises the methods chapters 4, 5, and 6.

Chapter 8 presents the results of construction of virtual slide maps using image stitching and the results of various object recognition tasks.

Chapter 9 summarises and provides detailed discussions on the results obtained and the performance of the algorithms. It also describes how the methods and algorithms developed can be used for auto-positioning in automated TB microscopy.

Chapter 10 presents the overall conclusions drawn from the study. Recommendations for future work are also made.

2. Literature review

No literature was found directly relating to auto-positioning in TB microscopy. Therefore, this chapter reviews methods that have been proposed for auto-positioning in microscopy in general and explores methods that may be applicable to TB microscopy. Because the work presented in this report is intended for use in automated TB microscopy, and because many of the microscope auto-positioning methods are built around automated microscopy, the relevant literature on automated microscopy for TB detection is first presented.

2.1 Automated microscopy for TB detection

Automated microscopy comprises several components including auto-focusing, image capture, segmentation, feature extraction, classification and auto-positioning. Auto-focusing ensures that the microscope automatically and correctly focuses on a slide under a microscope. Image capture acquires an image of the slide for processing. Image segmentation involves the separation of the candidate bacillus objects from each other and from the background in an image. Feature extraction is a process in which characteristic features of the objects-of-interest are extracted. Image segmentation and feature extraction are significant intermediate steps in image analysis for automated microscopy as the features extracted from the segmented objects are used in the classification process and also to create invariant descriptions of individual images for the auto-positioning process. Classification is the process in which a classifier is used to determine whether or not the extracted objects are indeed bacilli. Auto-positioning is the process of automatically moving to regions-of-interest on a slide and allows automatic navigation on the slide under a microscope.

2.1.1 Auto-focusing

In order to auto-focus, the microscope needs to be equipped with a motorized z-drive. Auto-focusing is a crucial step in automated microscopy as subsequent image processing greatly depends on it. Images that are well focused are expected to yield better results in segmentation, classification and auto-positioning. A well-focused image has the best

average focus over an entire field of view, even though objects often reside at multiple foci in thick sample slides (Osibote et al. 2010).

Auto-focusing algorithms establish a correspondence between the level of the microscope stage with respect to the slide, and the value of a focus measure. The focus measure is an indication of how in-focus an image is and it is commonly based on the illumination gradient of an image (He et al. 2003). Focus measures are typically Gaussian in shape, with the maximum of the curve corresponding to the position of focus as shown in Figure 2.1 (Russell 2006).

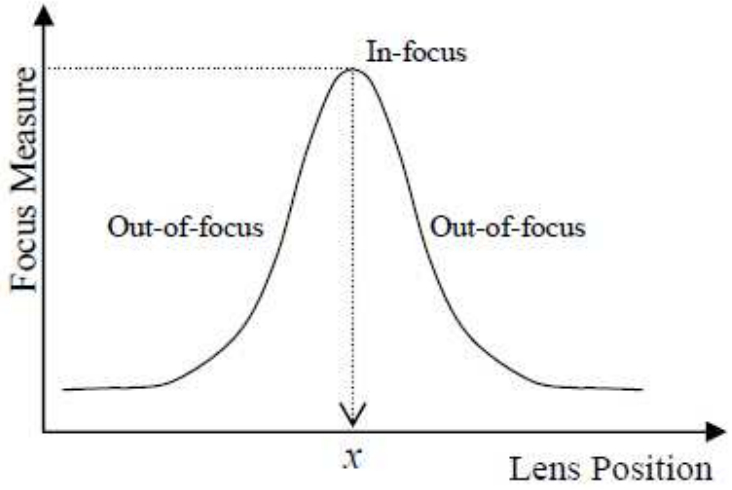


Figure 2.1: An example of an ideal focus measure function (Russell 2006)**.**

Russell (2006) experimented with auto-focusing algorithms using three different focus measures, namely the energy of the image Laplacian, the variance of the log-histogram and the first order Gaussian derivative, on Ziehl–Neelsen stained sputum smears. They found that the processing time of the energy of the image Laplacian focus measure was the least and that it worked well on slides with content of medium and low density.

Osibote et al. (2010) compared several existing auto-focus measures for digital microscopy to determine which one performs best for bright field microscope imaging of ZN-stained sputum smear slides. The methods they investigated included normalized variance, the Brenner gradient, the sum-modified Laplacian, the energy of the Laplacian

of the image, Vollath's F4 and Tenengrad's algorithm. They concluded that Vollath's F4 method performed best.

2.1.2 Image segmentation

Not all the information contained in images captured from a microscope is useful for image processing. Image segmentation can be used to extract only the useful information (Veropoulos 2001). Image segmentation is the process of partitioning the image into non-intersecting regions such that each region is homogenous and the union of any two adjacent areas is not homogenous (Pal and Pal 1993). Put simply, it involves the separation of objects in the image from each other and the background. For TB microscopy, it involves extracting candidate bacillus objects (Figure 1.1) from the background. The segmented bacillus objects represent the useful information that is carried forward to higher level processing such as classification and auto-positioning.

Several methods (Veropoulos et al. 1999, Forero et al. 2006) have been proposed to automatically segment tuberculosis bacilli from auramine-stained sputum smear images. Khutlang et al. (2010) compared pixel classifiers for segmentation of *Mycobacterium tuberculosis* in images of ZN-stained sputum smears which were obtained using a bright field microscope at 100x objective magnification. They used manually segmented bacilli as a gold standard, along with the Hausdorff distance and the modified Williams index, to compare the classifiers. The quadratic and Bayes' classifiers individually performed the best in terms of bacillus pixels correctly classified; 88.39% of bacillus pixels correctly classified and 38.08 % incorrectly classified. Using a combination of Bayes', quadratic and logistic linear classifiers improved the number of correctly classified pixels slightly, with 89.38% of bacillus pixels correctly classified and 39.52% incorrectly classified.

Segmentation of images generally generates some artefacts – objects that are not of interest. Filters that employ features characteristic to the objects-of-interest, may be used to filter objects. Khutlang et al. (2010) and Forero et al. (2006) filtered segmented sputum smear images to decrease the number of non-bacillus objects and hence further

decrease the amount of irrelevant information in segmented images. Khutlang et al. (2010) used an area based filter to reject objects with area that did not fall within threshold limits. Due to the long and thin shape of TB bacilli, Forero et al. (2006) employed an eccentricity filter in addition to an area filter, to eliminate objects with eccentricity lower than the threshold value. The threshold values were empirically determined. Additional features may be used to optimize the filtering process and hence further eliminate image information that may be irrelevant in further processing.

2.1.3 Feature extraction

To understand and interpret an image, object/region extraction and description is essential. This stage involves detecting features and generating numeric feature vectors or syntactic description words which characterise the properties of described features (Veropoulos 2001, Roth and Winter 2008). Some features such as size and shape properties are extracted from the segmented images while others such as texture and colour features can be extracted directly from colour or grey scale images. This step is necessary for several processes in automated microscopy including filtering of segmented objects, classification and auto-positioning.

To provide robustness, extracted features need to be invariant to various changes to provide robustness against image variations. In microscopy, image variations arise due to imperfect slide placement during re-loading resulting in geometric changes such translation, scaling, and rotation. Bacilli may occur in an image in any orientation. Additionally, differing lighting conditions result in illumination changes in images. Hence the extracted features need to be invariant to both geometric changes and photometric changes. This section explores and presents various features that can be extracted for the filtering process.

2.1.3.1 Boundaries and boundary descriptors

The boundary of an object is an important feature as it may be used to describe several aspects of the object. It is generally extracted only after segmentation. Boundary

descriptors can be employed for filtering (Forero et al. 2006). Typically, useful boundary descriptors for TB bacilli in segmented images include:

Boundary length - as the name suggests, this is simply the length of the boundary of the object. It is the simplest boundary descriptor.

Diameter - the diameter of the boundary can be used as a descriptor if it is unique.

Major axis, minor axis and eccentricity - The major axis is the longest diameter of the object while the minor axis is the diameter perpendicular to it. Both are boundary descriptors. The ratio of the length of the major axis to that of the minor axis is called the eccentricity (Gonzalez et al. 2004). Due to the long and thin shape of TB bacilli, eccentricity is an appealing descriptive feature.

Area enclosed by boundary - the area of an object can be simply defined as the number of pixels/sub-pixels contained within the object's boundary.

Compactness - Compactness is a measure of how closely the shape of the object approaches a circle; a circle being the most compact region in a Euclidean space. It is the ratio of the square of the boundary length to the area of the object (Forero et al. 2006). Compactness is a relevant feature since TB bacilli are long and thin.

Colour - When stained with the ZN method, TB bacilli stain red and the background blue. Therefore colour features can be used in both segmentation and classification to distinguish TB bacilli in images of ZN-stained sputum smears (Khutlang et al. 2010).

2.2 Microscope positioning and currently used solutions

When a slide is viewed under a microscope, only a small field of the smear is seen through the microscope's eye piece. A smear is made up of hundreds of fields depending on the objective being used. The field that is currently seen through the eye piece is called the field-of-view (FOV). In automated microscopy, the FOV is the field that is

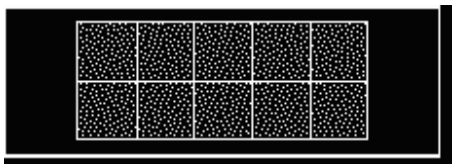
being seen by the microscope camera, which is usually smaller than that seen directly through the eye piece.

Microscope positioning is the process of bringing a desired region on the slide to the FOV of the microscope. This can be a time consuming and difficult task if manually performed. Existing techniques that enhance microscope positioning use specially manufactured slides. These include Field Finder slides (Electron Microscopy Science 2010) and CellFinder slides (Microlab 2010).

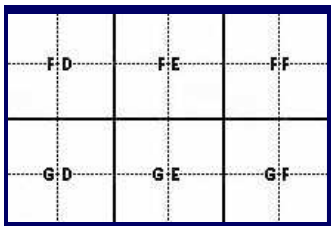
The Field Finder has a precision rectangular-coordinate grid pattern and each of the squares is marked with a letter and a number. When a given FOV on a slide, *S*, is to be marked as a region-of-interest, its coordinates are obtained by replacing the slide by a Field Finder slide and reading off the co-ordinates, *B3* for example, appearing on the FOV and storing them. The stored coordinates can then be used to relocate that region-of-interest when the slide, *S*, is re-loaded at a different time. This repositioning process is conducted as follows:

- Place the Field Finder slide on the slide holder of the microscope and manually move the XY stage so as to bring the square reading *B3* to the field-of-view of the microscope. Since the squares are marked in a particular order, moving from a given square to a desired square is simplified.
- Once the square *B3* is centered, replace the field finder with the slide *S*.
- The field-of-interest will now appear in the field-of-view of the microscope and has been relocated.

In the case of CellFinder, the slide consists of micro-patterns including squares, lines, dots and letters as seen in Figure 2.2 (Microlab 2010). In a similar process as that described above, the CellFinder allows marking of and repositioning to the region-of-interest.



CellFinder microscope slide 76x26 mm
with 10 CellFinder-cells of 10x10 mm



Section shows ~1% of a CellFinder-square
Each coded square is 400x400 µm
Character codes are 40 µm high
Reference dots are ~10 µm
Magnification ~75x

Figure 2.2: CellFinder Slides (Microlab 2010)**.**

These methods only slightly speed up the repositioning of the microscope to previously marked regions-of-interest as there is still a considerable amount of manual input. Additionally, they require precise slide placement during the marking and repositioning processes. Since slide placement is done manually, significant errors are likely to be introduced (Begelman et al. 2006).

Auto-positioning could greatly speed up microscope positioning and could also provide repeatability and higher accuracies.

2.3 Microscope auto-positioning

Auto-positioning in microscopy can be defined as the ability of the microscope to automatically bring a desired region on the slide to the field-of-view of the microscope.
In order to achieve auto-positioning, the following need to be known:

i) Point of reference

ii) The co-ordinates of the field-of-interest relative to the point of reference.

A point of reference is critical as the microscope needs to know exactly where it is before any stage movement to a desired location can be performed. A simple method is to use a motorized stage with a ‘home’ position, which acts a point of reference. The coordinates of the region-of-interest on the slide can be obtained relative to this home position. These co-ordinates can be read from the step counters of the x and y motors

and stored. During the repositioning phase, the slide is reloaded onto the microscope stage and the stage is automatically driven to the home position. The stored coordinates are then used to drive the motorised stage to the desired position, bringing the ROI to the field-of-view of the microscope (Graham and Cook 1985). This method however requires highly repeatable and precise repositioning of the stage's home position relative to the optical axis and equally precise repositioning of the slide on the stage's slide holder on re-loading. These requirements arise due to the open loop nature of the method, i.e. it lacks some form of feedback to the system on the location of the objective. In practice, this results in excessively demanding tolerances on mechanical components and electrical drive circuitry and hence high hardware cost (Graham and Cook 1985).

A closed loop approach to microscope auto-positioning would allow the microscope to automatically determine its position relative to the slide and hence would be more robust, improve accuracy and reduce cost of mechanical and electrical components. In order to reach this goal a virtual slide map may be used.

2.3.1 Auto-positioning as an object recognition task

Auto-positioning can be achieved using a virtual slide map which is a digital replication of a slide as seen under a traditional microscope. It is viewed on computer screen using viewer software and, with a pan and zoom viewer, it can emulate viewing the glass slide under a microscope (Dee et al. 2003).

Creating a virtual slide typically requires a microscope attached with a digital camera, a motorised XY stage and a computer with appropriate communication to the microscope. The microscope scans the slide by moving over its surface while capturing images of adjacent fields. The virtual slide is then created by combining these images (referred to as tiles). Since the relative positions of the individual tiles on the virtual slide are known, the virtual slide created is in fact a map of the actual slide (Begelman et al. 2006).

To create a good quality virtual slide map, i.e. perfect replication, of the actual slide surface, the images captured during the scanning process should be well-focused. Since the focus plane does not remain the same, especially if smears on the slide are thick and cover a large area, re-focusing of the FOV, after every shift to an adjacent field and before capturing its image, is often required. Typically, an auto-focusing algorithm is integrated with the microscope to perform this task to speed up the scanning process. In its absence, manual focusing can be used but this would be time consuming and laborious.

Once the virtual slide map is created, it can be used by the system to auto-position to regions-of-interest (ROIs) when a slide is reloaded onto the microscope slide holder. By using a virtual slide map, any given field-of-view can act as a point of reference, as its location in the virtual slide map can be determined. The co-ordinates of the ROI relative to the current FOV can be computed using the virtual slide map and these can be fed to the motorised stage to drive the slide so as to bring the ROI to the field-of-view of the microscope.

The localisation of the current FOV on the virtual slide map essentially involves matching the FOV image to the entire virtual slide map. This process can be simplified by breaking down the virtual slide map into smaller portions and then finding the correspondence between the FOV image and these smaller portions. Hence, microscope auto-positioning can be formulated as an object-recognition task. The current FOV acts as the query image and the portions represent models to which the query image is compared to find the matching model. This is referred to as model-based object recognition (Wolfson and Rigoutsos 1997, Lifshits et al. 2004).

Model-based recognition comprises broadly two stages; namely an *offline pre-processing stage* and an *online localisation stage*. The *offline pre-processing stage* involves the appropriate pre-processing of models and their storage in a suitable database. This stage is highly time consuming since every single model image is processed, but this time is of little significance as execution takes place offline. The

online localisation stage involves finding the best matching model to the query image. Less time is taken by this stage and it is this low online complexity that determines the actual time taken for object recognition (Begelman et al. 2006).

The co-ordinates, on the virtual slide map, of the current FOV are simply those of the matching model (Begelman et al. 2006, Lifshits et al. 2004). Lifshits et al. (2004) and Begelman et al. (2006) decomposed the virtual slide map so that each model overlapped an adjacent model by exactly the size of a single tile. Consequently, a model was made up of four individual adjacent tiles (images) captured during the scanning phase. This ensured that any query (FOV) image, which is the size of a single tile, acquired during object recognition would be entirely contained in at least one model provided the orientation of the slide is unchanged. In practice, this provides freedom to the operator to select any field on the slide to perform object recognition. Figure 2.3 illustrates how the models can be generated. The first model, M_1, consists of tiles 1,2,5 and 6 while the second model, M_2, is consists of tiles 2,3,6 and 7 and so on. It also applies in the y-direction, hence tiles 5 and 6 will be part of the 5, 6, 9 and 10 larger model.

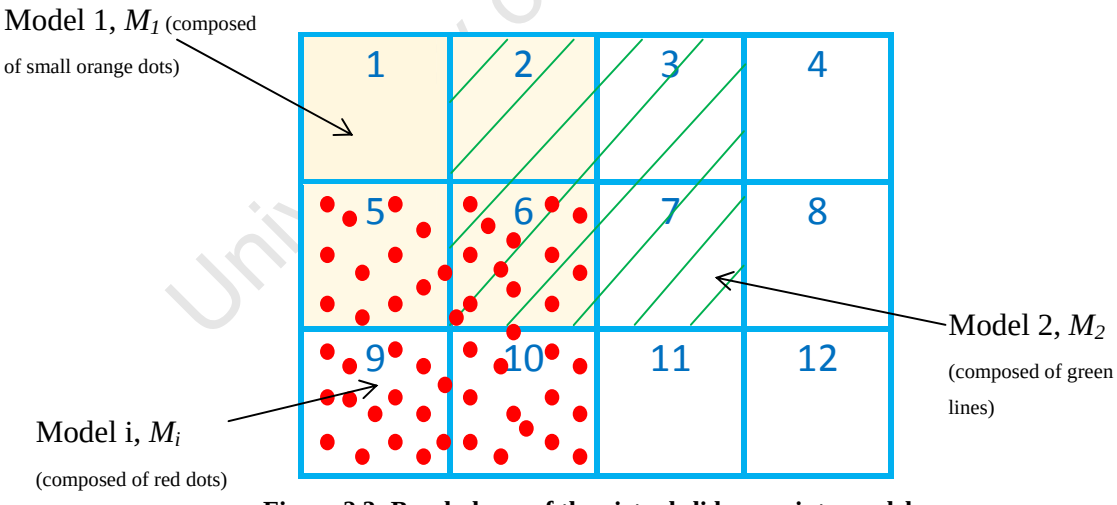


Figure 2.3: Break down of the virtual slide map into models.

In practice, due to imperfect slide placement (during re-loading), illumination changes and noise, the query image (FOV image) will not have an exact matching model but will have a best matching model (BMM), namely the model that most closely matches the FOV image. In order to successfully determine the BMM, the object-recognition scheme

needs to be robust to the above factors. Feature extraction plays a useful role in reaching this goal.

2.4 Feature extraction for auto-positioning

The boundary feature (Section 2.1.3.1) can be used for object recognition by representing the boundary in eigen-space or k-cosine-space (Sun et al. 2008, Sun et al. 2009). However, this approach is more suitable for bigger objects and shapes, such as machine tools, than bacilli in images.

For the recognition of small objects such as in microscopy auto-positioning, features are commonly represented as dots on the image followed by either assigning a descriptor to each dot or using the dots collectively to describe the whole image (Wolfson and Rigoutsos 1997). In the latter case, the image is represented as a point-pattern and common features extracted include linear segments, curvilinear linear segments, corners and points.

This section explores point-like detectors as they appear to be most suitable for recognition of small objects and microscopy object recognition (Begelman et al. 2006, Lifshits et al. 2004, Mehrotra et al. 2010). Hence from this point onwards, the word features refers only to feature points unless otherwise specified.

Mehrotra et al. (2010) use the scale-invariant feature transform (SIFT) to obtain feature points for iris recognition for biometric authentication. In microscopy imaging of integrated circuits, Lifshits et al. (2004) obtain feature points directly from the grey scale images of a wafer using the Harris corner detector. Begelman et al. (2006) adopt a medial axis transform on the segmented microscope images of lung and prostate tissue followed by junction point extraction to obtain the feature points.

2.4.1 Corner-based detectors

The Harris corner detector (Harris and Stephens 1988) is one of the most widely used point detectors. It responds to edge and corner-like features. It is based on the second moment matrix:

$$\mu = \begin{bmatrix} I_x^2(X) & I_x I_y(X) \\ I_x I_y(X) & I_y^2(X) \end{bmatrix} = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

where I_x and I_y represent the first derivatives of the image intensity I at position X in the x and y direction respectively. The cornerness measure c is efficiently computed by avoiding the eigenvalue decomposition of the second moment matrix using the equation:

$$c = Det(\mu) - k \times Tr(\mu)^2 = (AC - B^2) - k \times (A + C)^2$$

where k is a tunable sensitivity parameter. A non-maximum suppression step is then applied and a Harris-corner is detected by a high positive response of the cornerness function c. The advantage of this detector is that a large number of points are detected with high repeatability (Roth and Winter 2008).

Another corner detector is based on the Hessian matrix. This detector is based on a similar idea as the Harris-detector. It uses the second derivatives of the image intensity and hence responds to blobs and ridges in the image (Roth and Winter 2008).

Both these detectors extract points that are invariant to rotational changes but are non-invariant to scale (Roth and Winter 2008). However, these detectors can be made scale-invariant and these adapted versions are called Harris-Laplacian and Hessian-Laplacian detectors. They incorporate the scale normalized Laplacian, S, given by:

$$S = \sigma^2 \times \left| (I_{xx}(X) + I_{yy}(X)) \right|$$

Where σ is the standard deviation of Gaussian smoothing for scale space generation and where I_{xx} and I_{yy} are the second derivatives of the image intensity I at position X in the x and y direction (Mikolajczyk and Schmid 2001).

The computation of these detectors is relatively simple. However, their disadvantage is that they determine only the spatial locations of the feature points. They do not compute feature descriptors for each of the points. Consequently, Lifshits et al. (2004) employed the geometric hashing technique to create an invariant description of an image using the Harris corner points obtained from that image.

2.4.2 Medial axis transform

The medial axis transform (Blum 1967) is a shape representation technique in which foreground regions in a binary image are reduced to topology skeletons. Begelman et al. (2006) adopted this method as an intermediate step to allow the extraction of feature points, which were the junction points of the computed skeletons. They applied this transform to segmented images of lung and prostate tissue.

The topology skeleton is a thin version of the shape present in the image and hence acts as a shape descriptor. It largely preserves the extent and connectively of the original regions while throwing away most of the original foreground pixels (Loncaric 1998).

A topology skeleton, *H*, of a shape *S* with boundary δS, is a locus of points at equal distance from at least two boundary points.

$$H(\delta S) = \{ p \in S \mid q, r \in \delta S, q \neq r : dis(p,q) = dist(p,r) \}$$

where *dist (p,q)* is the Euclidean distance $||p\text{-}q||_2$. To facilitate in obtaining the skeleton, the distance transform (DT) is used:

$$DT(p) = \min_{q \in \delta S}(dis(p,q)), \quad \forall p \in S$$

This formula assigns to all the points *p* in the shape *S*, the distance to the closest boundary point *q* on δS. The topology skeleton is formed by all those pixels lying along the singularities such as curvature discontinuities of the distance transform (Bouix and Siddiqi 2000). The junction points of these topology skeletons can be used as the feature points as done by Begelman et al. (2006).

Begelman et al. (2006) employed the geometric hashing technique (Section 2.5.3) to create an invariant description of an image using the junction points obtained from that image.

2.4.3 SIFT features - Scale Invariant Feature Transform

The scale-invariant feature transform, SIFT (Lowe 2004), not only detects feature points, but it also associates a feature descriptor to each of the detected points. SIFT features are extracted directly from grayscale images. It uses the texture of the image to extract feature points, which are referred to as keypoints, and forms an invariant (to similarity transformation) descriptor vector around each detected keypoint.

It uses a cascade filtering approach to minimize the cost of feature extraction and obtains stable features from the image that are invariant to similarity transformation and partially invariant to illumination. It can be broadly divided into 2 stages, namely feature detection and feature description. The feature description is explained in Section 2.5.1.1. Feature detection involves detection of scale space extrema followed by keypoint localisation in the image.

Detection of scale space extrema – in this stage potential interest points invariant to scale are first detected. This is efficiently implemented using the difference-of-Gaussians (DoG) (Lowe 2004). The DOG of an image I is calculated for two nearby scales of the image by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

where k is a constant multiplicative factor used for changing the scale and x and y are the coordinates of a pixel in image I. $L(x,y,\sigma)$ is computed as :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $G(x,y,\sigma)$ is the Gaussian filter for smoothing the image and σ is the width of the filter.

Keypoint localisation – The keypoints are then detected with the help of local maxima and minima across different scales. This is done by comparing each pixel in the image to

eight neighbours in the same scale and nine neighbours in the neighbouring scales and if it is a local maximum or minimum, it is declared a candidate keypoint. Keypoints with low contrast, and hence high sensitivity to noise, are rejected using information that is derived by performing a detailed fit to the nearby data for location, scale, and ratio of principal curvatures (Lowe 2004).

Once the keypoints are detected in the image, they are assigned a feature descriptor vector as explained in Section 2.5.1.1.

2.4.4 SURF: Speeded Up Robust Features

SURF is a recent feature point detector and descriptor (Bay et al. 2006). Feature points are extracted directly from the grayscale images. The method employs the Hessian detector and uses the determinant of the Hessian matrix to select the keypoint location and scale. Given a point $X = (x,y)$ in an image I, the Hessain matrix, H, in X at scale σ is given by:

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix}$$

where $L_{xx}(X,\sigma)$, $L_{xy}(X,\sigma)$ and $L_{yy}(X,\sigma)$ are the convolution of the respective Gaussian second order derivatives with the image I in point X (Bay et al. 2006).

For each keypoint, a descriptor is then found. This descriptor is based on similar properties to the SIFT descriptor but its complexity is much lower as explained in Section 2.5.1.1.

2.4.5 Fiduciary markers

A fiduciary marker is an object/feature used in the field of view of a microscope which appears in the created virtual slide map and hence acts as a virtual mark. A fiduciary marker acts as a point reference, which can be used for measuring purposes or for auto-focusing and auto-positioning (Doerrer 2007). Microscope imaging and fiduciary

markers have been used for measuring and inspecting certain regions on a wafer (Lifshits et al. 2004). A wafer is a thin slice of semiconductor material used in the fabrication of integrated circuits. The fiduciary marker used can be simply a large and easy-to-track feature selected near the desired feature or region.

Fiduciary markers may also be created or set on a slide. Costantino et al. (2005) used two-photon optical microlithography to make fiduciary markers that can be used during fluorescence microscopy imaging. Fluorescent micro patterns are created on standard glass slides by the rapid fabrication of UV-cure resins to which fluorescent dyes have been added. This is done by using ultra-fast laser tools where a diffraction-limited laser spot is moved along the surface of a glass slide which is covered with a UV-cure resin. Due to two-photon nonlinear photo polymerisation, the material only solidifies in the direct vicinity of the focal spot. This technique has micrometer resolution. An automated x-y- stage allows precise movement of the sample with respect to the laser focus and thus two-dimensional fluorescent micro-patterns are created to act as reference points prior to scanning the slide and appear on the virtual slide map (Costantino et al. 2005).

2.5 Feature representation, storage and localisation

To fully represent an image, features not only need to be located but need to be descriptive of the image. Furthermore, to improve the robustness of the recognition scheme, the resulting image description should be invariant to geometric changes including translation, rotation and scaling. As mentioned earlier, a description can be attached to each feature point or the feature points can be collectively used to represent the whole image. In the former case, the description is called a feature descriptor. A feature descriptor describes the local neighbourhood around the already identified feature point using certain invariant properties. In the case of representing the image with all the points, the geometric hashing technique is most widely used (Begelman et al. 2006, Lifshits et al. 2004, Mehrotra et al. 2010).

The method of image representation determines the storage and object recognition methods and therefore all these stages are explained for each of the representation methods explored.

2.5.1 SIFT and SURF representation

After the keypoint localisation in the first stage (Section 2.4.3), the second stage of SIFT creates a descriptor for each keypoint (Lowe 2004).

2.5.1.1 Feature representation in SIFT and SURF

An orientation is assigned to each keypoint to achieve invariance to image rotation. This is done by computing a gradient orientation histogram in the neighbourhood of the keypoint. The scale of the keypoint is used to select the Gaussian smoothed image, L. For each image sample $L(x,y)$, the magnitude $m(x,y)$ and orientation $\theta(x,y)$ are computed as:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$$

An orientation histogram is formed from the gradient orientations of each keypoint. The orientation histogram has 36 bins covering the 360 degree range of orientations and each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with σ of 1.5 times that of the scale of the keypoint.

Peaks in the orientation histogram correspond to dominant orientation and then any local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation. This ensures stability during matching (Lowe 2004).

This is followed by the computation of a feature descriptor as a set of orientation histograms on 4x4 pixel neighborhoods. These orientation histograms are relative to the keypoint orientation. The histogram contains eight bins and each descriptor contains an array of four histograms around the keypoint, hence providing a 4x4x8 = 128

dimensional feature vector for each keypoint. Consequently, these descriptors are invariant to similarity transformations and are also partially invariant to illumination (Lowe 2004).

The SURF descriptor is based on similar properties to those of the SIFT descriptor. However, the SURF descriptor of each keypoint is a 64-dimensional feature vector as opposed to the 128-dimensional feature vector (Bay et al. 2006).

Both the keypoint location and the descriptor are stored in a database. Since in model-based object recognition, multiple numbers of images are participating, the location of the keypoint refers to both the image it arises from and the position of the keypoint in that image.

2.5.1.2 Object recognition in SIFT and SURF

During the SIFT object recognition stage, keypoints are extracted from the query image. This is followed by matching each keypoint independently to the database of keypoints. Due to image variations and noise, finding an exact match is unlikely and hence a best match is found for each keypoint. This is done by finding the nearest neighbour in the database which is the keypoint with minimum Euclidean distance for the invariant descriptor vector. Query keypoints that don't have 'good enough' matches are discarded. These include keypoints that were possibly not detected in the training images (model images). A distance-ratio threshold is used to measure how good the match is. This is the ratio of the distance of the closet neighbour to that of the second-closest neighbour (Lowe 2004).

Clusters of at least 3 features are first identified that are common in both the query image and a model image and its pose. This is because clusters of matched features have a much higher probability of being correct than individual feature matches. Each cluster is then checked by performing a detailed geometric fit between the query image and the candidate model, and the result is used to declare whether or not the candidate model is the best match to the query image.

Object recognition by SURF is done similarly (Bay et al. 2006). Bay et al. (2006) used photography images to compare SURF with previously proposed matching schemes including SIFT and showed that SURF not only computed and compared much faster, but it also approximated or even outperformed the others with respect to repeatability, distinctiveness, and robustness.

According to Lowe (2004), a typical image of 500x500 pixels would give rise to about 2000 stable SIFT keypoints (though this depends on the image content). If the number of images participating in the object recognition application is large, then an extremely large number of keypoints needs to be stored in the database. Each keypoint is described with a 128-dimensional vector and hence computer requirements are escalated. Furthermore, the simple approach of an exhaustive search by comparing each keypoint to all the database keypoints would be very time consuming (Mehrotra et al. 2010). As the number of keypoints in the database increases, the percentage of keypoints correctly matched decreases. A database of about 100,000 keypoints results in only 75-80% of keypoints correctly matched (Lowe 2004). Identical problems may arise with SURF keypoints.

Indexing schemes can be used to improve the efficiency and reduce the time taken to retrieve the matching model image. Mehrotra et al. (2010) integrate SIFT with a widely used indexing scheme, namely geometric hashing, for biometric authentication by iris recognition.

2.5.2 Fiduciary markers

The fiduciary marker is selected or created prior to scanning the slide (Section 2.4.5) and hence will appear on the virtual slide map. In the case where the fiduciary marker is a large, distinctive and easy-to-track feature, invariant (to similarity transformation) descriptors can be easily extracted for this particular feature. This may include shape, size, compactness, pixel intensity or any characteristic of that feature. In Costantino et al. (2005) the pattern of the fiduciary marker is known and characteristic and hence can act as the descriptor. The relative position of the desired region from this marker is

determined from the virtual slide map (Doerrer 2007). The marker descriptor and the relative position of the region-of-interest are stored. During the localisation process, the marker acts as an acquisition target which is first searched for by the system. By knowing the relative position from the marker, the region-of-interest can be detected (Lifshits et al. 2004).

The system may or may not satisfactorily locate the fiduciary marker. The first approximate location of the fiduciary marker is found and then the motorised stage of the microscope is automatically moved in a circular manner (with increasing radius, i.e. in a spiral) around this approximation comparing each field-of-view with the stored one (containing the fiduciary marker pattern) until the best match is found. The comparison of the FOV to the stored image is performed by comparing and matching the descriptors.

The operator may have to intervene and manually locate the marker around the approximation if it could not be found automatically. Once the fiduciary marker is located and in the right orientation, the regions-of-interest can easily be located and brought to the field-of-view of the microscope using the stored coordinates.

Using fiduciary markers is a time consuming process and suffers from a lack of precision that may arise from failure to replace the glass slide as it was when being scanned (Doerrer 2007). Another drawback to the technique proposed by Costantino et al. (2005) is that it may suffer from artefacts that may arise during the creation of fiduciary markers. Furthermore, it is less robust as the correlation-based matching techniques for localisation are highly sensitive to possible changes in the visual appearance of the feature. Such changes include nonlinear contrast variation, colour inversion and partial feature obliteration (Lifshits et al. 2004).

2.5.3 Geometric hashing scheme (GHS)

The geometric hashing scheme is a widely used model-based object recognition technique. It uses geometric invariants in images to find the model image in the database that best matches the query images. It employs dots obtained from the images which are

collectively used to generate an invariant description of that image and thus offers high tolerance to possible image variations including translations, rotations, scaling and illumination. The dots represent the locations of features extracted, which could be linear segments, curvilinear segments, corners and points (Wolfson and Rigoutsos 1997). If the features extracted are point features, then the points themselves act as the dots in the feature space in the geometric hashing technique. To explain this scheme it is assumed that the features extracted are points to simplify the explanation.

Lifshits et al. (2004) and Begelman et al. (2006) used this technique to create invariant representations of their images since the feature detectors they used, the Harris corner detector (Section 2.4.1) and the medial axis transform (Section 2.4.2), only give locations of the features and no invariant description. Although the SIFT keypoints have both location and descriptors and therefore is sufficient by itself for model-based object recognition, it can be integrated with the geometric hashing indexing scheme to improve efficiency and processing time as done by Mehrotra et al. (2010) for iris recognition.

2.5.3.1 Model representation and index generation in GHS

The query image, which is the current field-of-view, and all the model images, which are generated from the virtual slide map (Section 2.3.1), originate from the same slide. Therefore, if Q is the query image and the predefined objects (the model set) are $(M_1, M_2, M_3, \ldots M_n)$, then it is assumed that there is an M_i from that set that represents Q. M_i may be required to undergo a transformation to match Q. Under a microscope the transformations likely to occur include rotation, translation and/or uniform scaling which collectively are known as similarity transformation (Begelman et al. 2006, Lifshits et al. 2004), and intensity transformations. Each model is represented by invariant co-ordinates by computing the co-ordinates of the feature points with respect to a co-ordinate frame formed by the points themselves. This representation is invariant to similarity transformation.

To explain how this is done, the model representation of a single model is described. The same is done for all the other models.

Let $\{m_1, m_2, \ldots, m_k\}$ be the feature points of a given model, M. To form a co-ordinate frame, a pair of ordered points is selected. This pair of points is termed the basis of the 2-D co-ordinate frame. If the basis is formed by the ordered pair points (m_1, m_2), then a vector $(m_2 - m_1)$ and another vector at the midpoint of m_1 and m_2 and perpendicular to it form a co-ordinate frame in spatial domain as shown in Figure 2.4. The perpendicular vector is formed by rotating vector $(m_2 - m_1)$ by 90° about the midpoint of m_1 and m_2. The origin of the coordinate frame will be at the midpoint of m_1 and m_2.

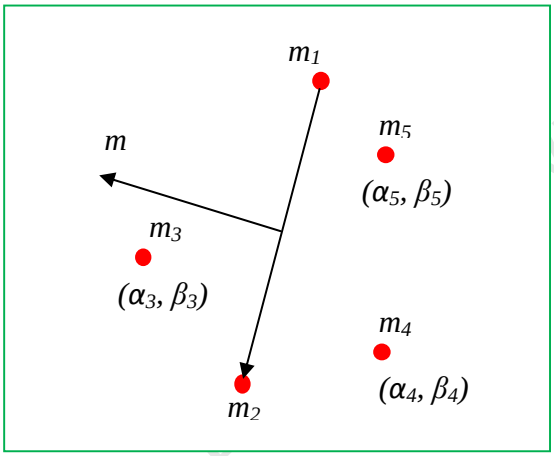


Figure 2.4: Image representation using feature points (dots) in the spatial domain.

In this frame, the coordinates (α_j, β_j) of each of the other feature points m_j in that model for $3 \leq j \leq k$, satisfy the equation below

$$m_j = \frac{m_1 + m_2}{2} + \alpha_j (m_2 - m_1) + \beta_j (m - m_1)$$

where m is the end point of the vector obtained by rotating vector $(m_2 - m_1)$ by 90°. This essentially re-scales the image such that the magnitude of the vector $(m_2 - m_1)$ is equal to 1 (Wolfson and Rigoutsos 1997). If a linear transformation, T, is applied on the model M, then the point m_j is transformed to:

$$Tm_j = \frac{Tm_1 + Tm_2}{2} + \alpha_j (Tm_2 - Tm_1) + \beta_j (Tm - Tm_1)$$

Consequently the new point Tm_j will have the same coordinates (α_j, β_j) in the frame formed by the ordered basis pair (Tm_1, Tm_2) and hence the co-ordinates (α_j, β_j) are referred to as invariant co-ordinates under similarity transformations (Begelman et al. 2006, Lifshits et al. 2004).

In a model image, there are different combinations of a basis pairs that can be selected to form a co-ordinate frame. If the number of feature points in a model M is denoted as N_i, then there are $\binom{N_i}{2}$ different bases in that model.

A transformation-invariant representation of the model is formed by computing the invariant coordinates (α_j, β_j) using each one of these bases $B_{\mu v} = \{m_\mu, m_v\}$ for every other model point m_j in that model. The corresponding entry $(M_i, B_{\mu v})$ is then stored in a hash table bin with index (α_j, β_j) , $3 \leq j \leq k$. This representation is done for each of the models in the database. Many hash table bins will receive more than one entry, and those bins will each contain a list of entries of the form $(M_i, B_{\mu v})$. The entries in the hash table are an invariant representation of the model image.

A hash table is a data structure or look-up table that offers very fast insertion and searching of elements. No matter how many data items are present in the hash table, insertion and searching can take close to constant time, referred to as *O(1)* in Big-Oh notation (Lafore 1999). The computed coordinates α_j and β_j act as indexes (keys) to the hash table and they may be transformed into a single key using an appropriate hash function. A hash function is a function that maps an item into a small index or into a more appropriate index. Certain types of keys do not require a hash function and can be directly used as array indices. The use of a hash function may introduce a complication called collision. A collision occurs when two or more different items hash out to the same index. Several methods such as linear probing, quadratic probing and separate chaining can be adopted to overcome this problem (Weiss 1999).

Hash tables are based on arrays and large arrays are difficult to expand after they have been created. Consequently, one has to predict in advance the size of the database in order to simplify the implementation of a hash table. If the size is predictable then hash tables offer high speed and convenience in insertion and searching items (Lafore 1999).
If a model has n feature points then the model will have about n^3 entries into the hash table . The database storage requirements may therefore become a major issue as the

number of feature points per image increases. Highly populated hash table bins also degrade the performance of the system in terms of both time taken and false matches (Lifshits et al. 2004).

2.5.3.2 Object-recognition

The object recognition stage comprises a voting stage followed by a verification process.

Indexing and voting stage

An invariant representation of the query image, *Q*, is obtained in a similar manner as the model representation. However, it is represented by only one arbitrary basis pair, *Bq*. This is so because all the possible basis combinations in the matching model have been used in model representation and hence, theoretically, any basis selected in the query image has got a matching basis in the database.

The invariant coordinates (α_j, β_j) of all the other feature points in *Q* are computed with respect to the coordinate frame formed by B_q. The coordinates are then used as an indexing key to access the hash table and vote for entries $(M_i, B_{\mu v})$ stored in the accessed hash table bins. If the selected B_q has a corresponding basis in one of the models in the database, then that entry $(M_i, B_{\mu v})$ is expected to receive votes from all the unconcluded points belonging to that model. Even if several feature points in the query image may not have a corresponding point in the matching model, recognition is still possible as long as sufficient number of points hash to the corrects bins (Wolfson and Rigoutsos 1997). In general, the voting scheme does not give only one candidate solution. Models that receive a significant number of votes (above a threshold, λ) are deemed possible matches to *Q*. These possible matches are termed as the candidate models (Lamdan and Wolfson 1988).

Due to the presence of noise, there are positional errors of feature points in the query image and the matching model image. These positional inaccuracies introduce errors in the extracted values of the computed coordinates and this may result in accessing and voting for incorrect bins of the hash table. However, the 'correct' bins that would have been accessed had the query image been noise-free lie in the neighbourhood of these

'wrong' bins (Wolfson and Rigoutsos 1997). One way to overcome this problem is instead of simply voting for entries in this 'wrong' bin, a rectangular region of the hash table around this bin can be accessed (Lamdan and Wolfson 1991). By voting for all the entries in these surrounding bins, the votes for the correct model will not be lost. Another way to overcome the problem is to use weighted voting as done in (Costa et al. 2002).

If no possible matches with a significant number of votes are detected, the voting stage is repeated using another arbitrary basis pair in Q. This is because in practice, due to image variations and noise, it is possible that Q may have additional feature points called outliers. Consequently, if one of these points is selected to form the basis pair, B_q, then it is possible that this basis pair will not correspond to any of the model basis pairs in the database. Possible inaccuracies in the basis point location (induced by noise), may also have similar effects. Hence the selection of the two points that form the basis, B_q, plays an important role and influences the outcome of the recognition process. Wolfson and Rigoutsos (1997) show that the quality of a given basis is dependent on the distance between the points making up that basis. The larger the separation between the basis points the smaller the spread in the space of invariants. Hence a large separation of the feature points making up B_q results in an image description that is less sensitive to noise, i.e., the computed invariant coordinates will have smaller noise-induced inaccuracies. They also show that noise sensitivity is further reduced if the remaining feature points are closer to the centre of the formed coordinate frame.

Due to the nature of the entries, i.e. $(M_i, B_{\mu\nu})$, stored in the hash table bins, the voting stage does not only reveal the candidate models, but it also reveals which basis pair in those candidate models is a potential correspondence to B_q of Q. Hence they can be referred to as candidate model-basis-pair combinations, *CMBs*. The voting scheme therefore acts as a sieve reducing significantly the number of candidate hypotheses for the verification stage.

Verification and image registration

Owing to the presence of noise and outliers in the query image, the best matching model-basis-pair combination (*BMMB*) is not necessarily the *CMB* that received the highest vote. A verification process is carried out to find out which *CMB* matches the query image by comparing each *CMB* to the query image, *Q*.

To match the query image *Q*, a candidate model may have to undergo a similarity transformation, *T*, which is a combination of translation, rotation and isotropic scaling.

Since the query image and a candidate model image are represented as dots, they are essentially two sets of point patterns. Consider two point sets of patterns:

$M = \{m_j \mid j = 1,2,\ldots k\}$ where m_j are feature points in the matching model and are described with coordinates x and y and denoted as $m_j = (x_j, y_j)$.

$Q = \{q_i \mid i = 1,2,\ldots r\}$, where q_i are the corresponding feature points in the query images and are described with co-ordinates x and y and denoted as $q_i = (x_i, y_i)$.

If *T* is the similarity transformation candidate model *M* undergoes to match the query image *Q*, then $Q = T(M)$ which is essentially equivalent to $q_i = Tm_j$. A similarity transformation is fully described by 4 parameters, (t_x, t_y, s, θ):

t_x = translation in the x-direction
t_y = translation in the y-direction
s = scale factor
θ = angle of rotation

Therefore the transformation mapping between the corresponding feature points in the query image and the model image can be formulated as:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = T\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x_j \\ y_j \end{pmatrix}$$

The verification process involves approximating the desired similarity transformation, *T*, and its parameters and checking for the number of inliers present between the candidate model and *Q*.

Observing the above equation, there are two equations and four unknowns, (t_x, t_y, s, θ) to be solved. Hence a unique solution of a similarity transformation *T* can be obtained by using two point-to-point correspondences between *Q* and *M* (Cheng 1996). The voting stage produces a basis in the candidate model that corresponds to basis, B_q in the query image. These corresponding bases (which essentially are two point-to-point correspondences) are used as the control points to find an approximation of T, $\hat{T}$. Upon applying $\hat{T}$ to the candidate model, every model point $\hat{T}m_j$ will correspond to the closest feature point $q_i^{'}$ in *Q*. That is:

$$q_i^{'} = \arg\min_k d(q_k, \hat{T}m_j)$$

where sub index *k* indicates any feature point in *Q*, and *d(x, y)* is the Euclidean distance between two points *x* and *y*. Thus to compute all the point-to-point correspondences between a candidate model *M* and the query image *Q*, only the distance between each point $q_i^{'}$ and the transformed model point $\hat{T}m_j$ needs to be checked. If the distance between correspondences is below a threshold value, then those corresponding points are considered as inliers. Otherwise they are classified as outliers. Outliers can also be defined as an inconsistent feature point in *Q* that does not have a corresponding point in the model. The threshold distance is empirically determined (Hartley and Zisserman 2003).

Lifshits et al. (2004) and Begelman et al. (2006) adopt the Voronoi tessellation of the point patterns to accelerate the computation of finding the correspondences. Voronoi tessellation partitions a point pattern plane containing *n* points into *n* convex polygons such that each polygon contains exactly one point and every point in a given polygon is closer to its central point than any other (De Berg et al. 2008). Figure 2.5 illustrates a Voronoi tessellation of an image containing 25 feature points. Constructing a Voronoi

tessellation on feature points, q_i in Q, allows the corresponding model point of m_j to be found by simply checking which polygon within the Voronoi tessellation contains the transformed point $\hat{T}m_j$ and choosing its centre point. To do so, a search data structure based on Delaunay triangulation (a dual of Voronoi tessellation) can be utilized to perform this task (De Berg et al. 2008).

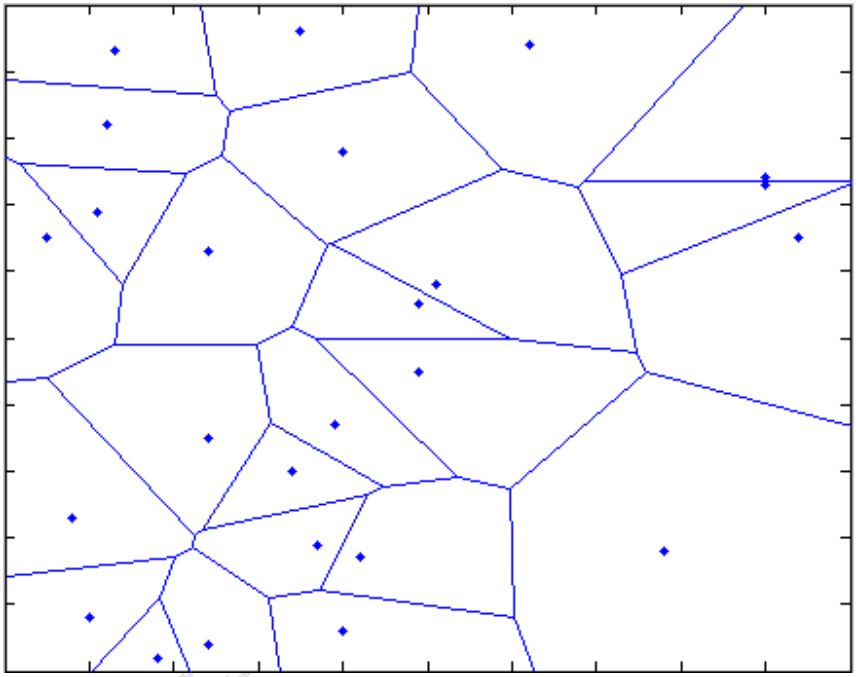


Figure 2.5: Voronoi tessellation of a given set of feature points

The candidate model that results in the highest number of inliers is declared the best match model to Q.

In practice, due to noise and errors introduced by the feature detector, some feature points in the query image, Q, might be mistakenly reported and will not match any model point – outliers. These outliers severely disturb the estimated transformation and need to be identified and discarded to allow optimal transformation estimation (Hartley and Zisserman 2003).

Liftshits et al. (2004) and Begelman et al. (2006) employ a robust estimator, the RANSAC algorithm (Fischler and Bolles 1981), to make it robust to outliers. The

primary basis correspondence between the candidate model and the query image is used to compute an approximate transformation $\hat{T}$ which along with Voronoi tessellation and Delaunay triangulation, is used to generate a putative set of correspondences which is essential for RANSAC.

2.6 Random sample consensus (RANSAC)

RANSAC is an algorithm for robust fitting of two point patterns. It is capable of interpreting and smoothing data containing a significant percentage of gross errors arising from noise and error-prone feature detectors (Fischler and Bolles 1981). Before RANSAC can be performed, a putative set of point-to-point correspondences between the two point patterns is required. A sample from this set is then randomly chosen and this sample is used to compute the transformation linking the two point patterns. The size of the sample is equal to the minimum number of point-to-point correspondences required to estimate the free parameters of the transformation. As explained above, a similarity transformation can be estimated using a minimum of two point-to-point correspondences and hence the sample size should be 2.

Given a putative set of correspondences $x_i \leftrightarrow x_j$, a sample of two point-to-point correspondences is randomly selected and the approximation of the similarity transformation $\hat{H}$ is computed. The number of inliers, which are the point-to-point correspondences consistent with $\hat{H}$, is computed in terms of correspondences that are within a distance threshold, *t*. That is, a point is deemed an inlier if:

$$d(x_i^{'}, x_j) < t$$

where $x_i^{'} = \hat{H}x_i$ and $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$. For practical applications, the distance threshold is chosen empirically (Hartley and Zisserman 2003).

This process is repeated several times with different samples and $\hat{H}$ which provides the largest number of inliers is selected as the best estimate transformation between the two point patterns.

Normally, the transformation parameters estimated by RANSAC are not very precise and therefore they are re-computed using only the inliers found by RANSAC. This can be done by a least squares fit (Hartley and Zisserman 2003, Fischler and Bolles 1981).

This process can be performed for each and every candidate model to determine the optimal number of inliers and optimal transformation between the query image and the candidate model. The candidate model that shares the most inliers with the query image is then declared the best matching model.

2.7 Combining images to construct the virtual slide map

As explained Section 2.3.1, to create a virtual slide map the slide is first scanned by the microscope by moving over the surface of the slide while capturing images of adjacent fields. These individual images, tiles, are combined to form the virtual slide map. There are two ways to combine the individual images - by end-attachment of individual images and by image stitching.

2.7.1 End-attachment of individual images

A quick and easy method to create a virtual slide map is to simply attach the adjacent images at their ends. However, this requires a well-calibrated and accurate motorised XY stage to allow precise movement to an adjacent field without any overlapping with the previous field. It also requires precise alignment of the camera with the stage to ensure that objects move parallel to the x- and y-axes when the stage is moved to eliminate the element of rotation between adjacent images.

2.7.2 Image stitching

Image stitching is the process of merging two or more images that share a common region (i.e. overlap each other). Image structures of the overlap region are used to combine adjoining images and therefore during the scanning process, movement of the microscope stage must be such that there is an overlap between the previous FOV and the new FOV prior to capturing the image of the new FOV. Consequently, with image stitching, requirements of an accurate mortised stage and precise alignment are relaxed. In fact even a non-motorised XY stage can be used since manually moving to adjacent

fields while having an overlap is not a difficult task. There should be sufficient information/structures in the overlap region to ensure good matching and hence good stitching of adjacent images.

Image stitching can be done manually or automatically. It comprises two stages; image registration followed by image merging. Image registration, which is the core component, involves identifying the common overlap between adjacent images and the appropriate transformation relating the overlapping regions. This information is then used to merge the two images at this overlap. The identification of the overlap region is therefore critical to the image stitching process (Brown and Lowe 2003).

Image stitching can be conducted manually using software such as Adobe Photoshop 7.0 since the human eye can easily identify overlapping regions between adjacent images. Once the overlap region is identified, the images can be manually moved, aligned and superimposed to create a composite image, referred to as a panorama. However, this process can be highly time consuming and laborious especially if a large number of images needs to be stitched.

Automatic image stitching requires automatic image registration i.e. automatic identification of the overlapping regions between two adjacent images and computation of the transformation. Therefore, the stitching algorithm is based on pattern recognition. Firstly, a structure is detected in the overlapping region in an image. An attempt is then made to find this structure in the overlapping region of the neighbouring image. The matching structures are used to compute the transformation linking the two images. This transformation is then used to stitch the images to form a composite image (Brown and Lowe 2007). Incorrect stitching position may result particularly if the specimen displays similar or recurring structure patterns.

The merging of the images can be done by simply replacing the common overlapping regions by that from one of the constituent images – the no-blend method. However neighbouring image edges may show undesirable intensity discrepancies which form a visible seam line. To eliminate these and hence to improve the visual quality of the

composite image, a blending algorithm can be applied. These include linear blending and multi-band blending algorithms which can be used to compute the final values of the pixels in the overlapping zone (Ma et al. 2007).

Several freeware such as Hugin (Hugin 2010) and Panorama maker (Panorama Maker 2010) are available to automatically stitch overlapping images. These are based on the same principle. Using pattern recognition or matching algorithm, common points between two overlapping images are detected to act as control points. These are used to compute the transformation linking the two images which is then used stitch the images into a panorama. These and other stitching software were developed for applications in photography. Ma et al. (2007) tested several such software for automatically stitching microscope images of mouse lymph node and intestine and found them unsuitable for this application. Therefore, an auto-stitching scheme suitable for one application may be unsuitable for other applications.

Brown and Lowe (2007) used the matching scheme, SIFT, to automatically stitch images and create panoramas, and as a basis for their proprietary software, Autostitch (Autostitch 2010), which was developed for the application in photography.

Digital microscopy dedicated software such as AxioVision 4.8 - Panorama and MosaiX modules - offers both manual and automatic image stitching. For this software, the recommended overlap between adjacent images is 20-30% (ZEISS 2010).

2.7.2.1 Stitching quality

Ma et al. (2007) tested the use of Autostitch for automatically stitching microscope images of mouse lymph node and intestine. They stitched a maximum of 8 images. They measured the stitching quality visually by the similarity of the stitched image to each of the input images and by the visibility of the seam between the stitched mages.

They also used a simple quantitative test to measure stitching quality. They divided an original image of a mouse lymph node, measuring 3840 x 3072 pixels, into four overlapping images and stitched them using the Autostitch software. Three points were

selected in the original image and the corresponding points in the stitched composite image found. The three points in the original image were joined to generate a triangle and the ratio between the lengths of the three sides were computed and compared to that of the corresponding triangle in the stitched composite image. They did so for several different triplets of points and concluded the SIFT scheme is very efficient for stitching microscopy images.

2.8 Methods for evaluating the performance of the object-recognition technique

The object recognition process has two components: localising the FOV, which is equivalent to finding the model from the database that is the best match to the query image, and secondly, image registration, which involves finding the optimal transformation relating the query image and that model. Performance of both the stages can be evaluated.

2.8.1 Hit rate to evaluate the performance of the localisation stage

The hit rate *(HR)*, which is equivalent to the true positive rate (TP rate), can be used to measure the performance of an object recognition scheme.

Lifshits et al. (2004) and Begelman et al. (2006) both experimented with microscope images obtained at 20x magnification and used the geometric hashing scheme for localisation. Lifshits et al. (2004) used images obtained by capturing the different fields in a wafer microscope slide and used these images to create the virtual slide map. For their experiments they used only 1 real wafer map. They divided the map into models. Their virtual slide map generated 72 models. To perform a localisation task, a random image equal to the size of a field-of-view image (which they refer to as eye-point) was selected directly from the virtual slide map, which acted as the query image. They then performed the object recognition task. In a second test they systematically altered the query images to simulate visual changes such as illumination changes and image rotations which are likely to occur in practice.

If their recognition scheme correctly located the query on the slide map (i.e. if the recognition scheme correctly found the best matching model to *Q*), the result was reported as a true positive *(TP)*. Since *Q* was selected from the virtual slide itself, the best matching model and its location were known. In addition, the simulated distortions on *Q* were known. Hence ground truth was available to determine a true positive result. They regarded the result as a false positive *(FP)* if an incorrect location was produced by their algorithm; if no location was found, simply because none of the database models got enough votes, it was regarded as a miss. They used the hit rate, *HR = number of TP / number of Tests*, to evaluate the performance of their algorithm. For undistorted query images they achieved a hit rate of 95%, a false positive rate of 4% and a miss rate of 1%. In the cases of illumination changes and image rotations (up to 8°) they achieved a hit rate of above 80%. They used a maximum of 5 bases per query image during the voting stage.

Begelman et al. (2006) also used hit rate to evaluate the performance of their method. However, if either an incorrect or no location was produced, they reported it as a miss. They do not report the number of models generated from the virtual slide map but using the figures they provide, it can be estimated they used about 90 models. They captured images at 20x magnification of the different fields in a slide and used them to create the virtual slide map. In their experiments they used only 2 slides, one for lung tissue and the other for prostate tissue. They selected a query image, which they refer to as region-of-interest (ROI), from the virtual slide itself and added Gaussian noise to the extracted feature points prior to localising it. They achieved a hit rate of > 90% for a moderate noise level of $\sigma < 1$. In the second phase of testing they distorted the query images to simulate real changes. For illumination changes and image rotations (up to 12°) a hit rate of above 80% was achieved. They used about 50 different bases per query image during the voting stage of a query image.

2.8.2 Discriminative Power

In the verification stage (Section 2.5.3.2) of the geometric hashing technique, all the candidate models-basis combinations (*CMBs*) are sequentially compared to the query

image to find best matching model-basis combination *(BMMB)*. Consequently, a large number of candidate model-basis combinations would degrade the performance of the recognition scheme in terms of both time taken and false matches.

If the *BMMB* has received significantly much higher votes than the other *CMBs* in the database, then by arranging the candidate list in descending order of received votes and, only verifying the top few *CMBs*, the *BMMB* would still be found (Lifshits et al. 2004).

The discriminative power can be defined in terms of the position of the *BMMB* in the candidate list sorted in decreasing number of votes. The higher the *BMMB* in this list, the greater the discriminative power of the object recognition algorithm.

Lifshits et al. (2004) showed that a quadtree enhancement improves the discriminative power and hence aids in reducing the number of models to verify and also reduces false matches. The quadtree decomposition of the virtual slide map can be done at various depths. The virtual slide map is divided into four quadrants and each quadrant is further divided into 4 quadrants and so on. Figure 2.6 illustrates the quadtree decomposition of the virtual slide map (Lifshits et al. 2004). However, this method requires the initial position of the query image on the virtual slide to be roughly known. Thus only models within this 'expected region' need to be used during the voting stage.

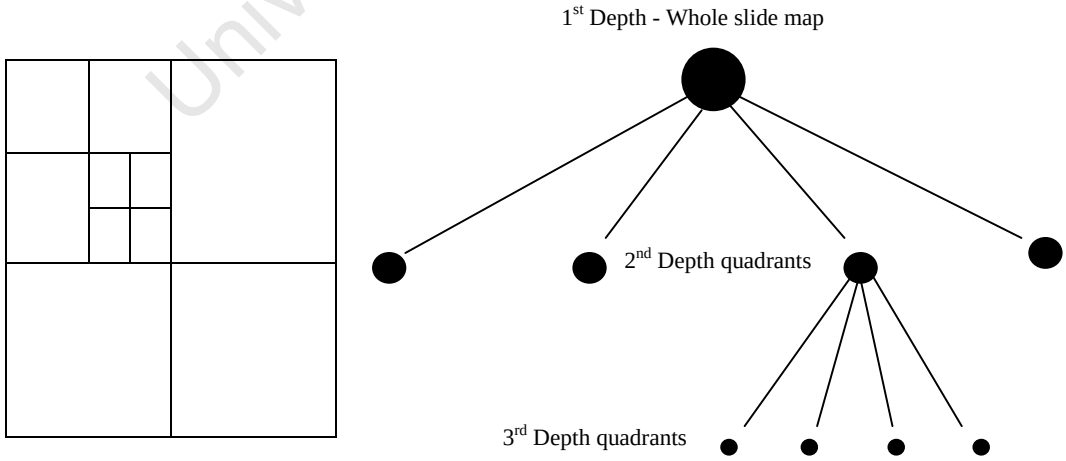


Figure 2.6: Quadtree decomposition of the virtual slide map.

Therefore, for example, if the query image is known to lie in one of the 2nd depth quadrants, then the voting stage is performed using only those models that are within that quadrant. The entries in the hash table are also stored using a quadtree data structure (Lifshits et al. 2004). This allows any partial slide map area to be searched for to localise the query image, hence reducing the number of irrelevant models. This in turn results in lower number of *CMBs* to verify in the verification stage and hence the position of the *BMMB* is much higher on the candidate list.

They experimented with the quadtree implementation with depth two. The position of the *BMMB* was improved by a factor of four (i.e. it moved four times nearer to the top of candidate list) which was expected as only a quarter of the slide map was actually participating in the voting stage.

2.8.3 Evaluation of image registration

Model-based object recognition performed using feature points is essentially a point-pattern recognition technique. A query image point pattern needs to be matched to one of the sets of point-patterns that are pre-defined, which are called models. Once the matching model point-pattern is identified the transformation matching the query point-pattern and model point-pattern can be estimated. The registration of the query image and model image is represented by this estimated transformation.

2.8.3.1 Root mean square estimation error and root mean square residual error

Lifshits et al. (2004) and Begelman et al. (2006) formulate the inaccuracy of their transformation using the root mean square estimation error, ε_{est}, and root mean square residual error, ε_{res}, (Hartley and Zisserman 2003). Assuming the feature points of the query image are measured with a Gaussian error of standard deviation σ,

$$\varepsilon_{est} = \sigma\left(\frac{2}{n}\right)^{\frac{1}{2}} \qquad \varepsilon_{res} = \sigma\left(1-\frac{2}{n}\right)^{\frac{1}{2}}$$

where n is the number of point-to-point correspondences between the query point-pattern and the best matching model point-pattern (Hartley and Zisserman 2003). These

formulas are applicable only when synthetic data is used and error is assumed to be in the query image only (Hartley and Zisserman 2003).

For example, with 50 matching pairs and with measurement error level assumed to be 1 ($\sigma = 1$), the estimation error is 0.2 pixels and residual error is ~0.98 pixels.

2.8.3.2 Mean square error

The mean square error is commonly used to evaluate registration error between two images (Zitova and Flusser 2003). Cheng (1996) refer to this as the average pairwise error and use it to determine the matching error between two point patterns.

Let $\{q_1, q_2, \ldots, q_k\}$ be points in the query image Q, and $\{m_1, m_2, \ldots, m_k\}$ be the corresponding points of the matching model, M, and let T be the best estimated transformation mapping the model M to the query image Q, then $Q' = T(M)$, or $q_j' = Tm_j$ for $1 \leq j \leq k$, where Q' is the transformed point pattern of M and hence q_j' is the transformed point of m_j for $1 \leq j \leq k$.

The pairwise error, ε, of a point q_j is given by:

$$\varepsilon = \left\| T((m_j)) - q_j \right\|^2 = \left\| q_j' - q_j \right\|^2$$

The average pairwise error (mean square error) between the query point-pattern and the matching model point-pattern is simply the average of the pairwise errors of all q_j, $1 \leq j \leq k$ (Cheng 1996). The square root of the average pairwise error is equal to the fiducial registration error (Fitzpatrick and West 2002).

2.8.3.3 Comparative method

Another approach to evaluating image registration performance is to compare image registration results obtained using the method under investigation with those obtained using a comparative method. Small differences between the two sets of results indicate good image registration accuracy. The comparative method should preferably be a 'gold standard method', which is a method commonly believed to be the best in the particular application area or for the given image type. This approach is often used in medical

imaging. In application areas where a gold standard does not exist, a method of different nature can be used as the comparative method (Zitova and Flusser 2003).

2.8.3.4 Visual assessment

Visual assessment is the oldest method of estimating the accuracy of image registration. It is still in use at least as a complement of the above evaluation methods (Zitova and Flusser 2003).

3. Materials

Experiments were conducted on Ziehl-Neelsen (ZN) stained sputum smear slides. These slides were prepared by the National Health Laboratory Service (NHLS) at Groote Schuur Hospital in Cape Town. The NHLS routinely prepares auramine stained sputum slides. However, the residual sputum was stained using the ZN method to accommodate this investigation. Only slides that were positive for tuberculosis, as confirmed by the NHLS, were used. Sputum had been liquefied (digestion) and decontaminated. Cover slips were mounted onto the slides using the rapid resinous mounting medium, Entellan. Prior to mounting, the slides were dipped into Xylol 4, which acts as a clearing agent and as a solvent of the mounting medium.

All image processing algorithms were developed on a 32-bit desktop computer with a 2.67 GHz Intel processor and 2.96 GB of RAM using MATLAB R2007b, its image processing toolbox and the PRtools toolbox (Duin et al. 2004).

3.1 Microscope

The bright field microscope, ZEISS Axioskop 2, was used for this investigation. High resolution colour images (1030 x 1300 pixels) of the field-of-view were captured using the attached Axiocam HR digital camera. The microscope uses a 12V, 100W halogen bulb to illuminate the slides. The images were captured in a room lit by a fluorescent light. The microscope was used at 40x magnification with numerical aperture 0.75. It had a 10x ocular lens and a built-in ancillary magnification lens - 1x Optivar. Each pixel measured 0.27 x 0.27 μm. White balance was done on an empty slide.

The microscope was equipped with an XY stage which was not motorised. Hence, all slide movements were manually performed. No auto-focusing algorithm was integrated to the system and therefore the microscope was manually focused to capture images.

Both the microscope and digital camera were connected to a desktop computer and along with the microscope dedicated imaging software, Axiovision 4.7, they provided a high quality real time display of the slide FOV on the monitor.

Figure 3.1 shows the ZEISS Axioskop 2 and a typical 1030 x 1300 image produced by the microscope.

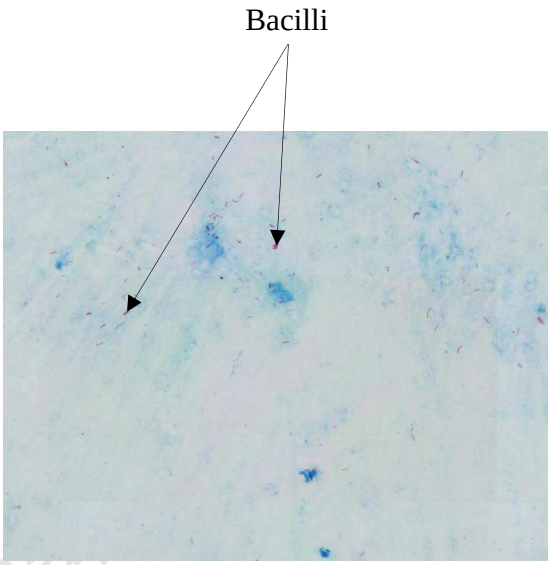


Figure 3.1: ZEISS Axioskop 2 microscope and a typical sputum smear image captured by the microscope.

4. Methods: Offline pre-processing stage

Based on the literature, the two most appealing object recognition techniques for auto-positioning were the geometric hashing scheme (GHS) and the SIFT scheme. For the TB microscopy application, the SIFT scheme was found to be unfavourable for auto-positioning mainly due to computer RAM constraints. Therefore, the object recognition algorithm that was developed was based on the GHS scheme which has also been used for similar applications in other fields of microscopy (Begelman et al. 2006, Lifshits et al. 2004). The GHS scheme is a model-based object recognition technique and therefore constitutes two stages, namely the *off-line pre-processing stage* and the *online localisation stage* (Section 2.3.1). This chapter details the methods that were used for the *off-line pre-processing stage*. Chapter 5 details the methods that were used for the *online localisation stage*.

The *offline pre-processing stage* involves the construction of the virtual slide map, decomposition of it into models, the pre-processing of the models and the storage of the models into a suitable database. This stage is highly time consuming since every single model image needs to be processed, but this time is of little significance as execution takes place offline, prior to the online localisation.

4.1 Construction of the virtual slide map

The larger the virtual slide map, the greater the computing requirements. Owing to this, a virtual slide map was not created for the entire smear on the slide but only for a reasonable rectangular section of the smear. The use of a cover slip allowed a rectangular region to be marked - with a permanent marker - without affecting the actual smear. Since a non-oil 40x objective lens was used, the rectangular mark was not affected during the experiments.

To create the virtual slide map of the marked rectangular region of a slide, the slide was first placed onto the slide holder of the microscope, followed by scanning and image acquisition, and finally combination of the acquired images.

4.1.1 Scanning and image acquisition

Ease of image acquisition is typically facilitated by a motorised stage and an auto-focusing algorithm integrated with the microscope. However, the microscope used lacked both of these features and therefore stage movement and focusing before image capture were done manually. The slide was placed in the slide holder so that it aligned to the x-axis of the XY stage i.e. at 0°. The camera was aligned to the stage so that objects would move parallel to the x and y axis of the field-of-view. Image acquisition was done at 40x magnification. The first field to be captured was the field at the top-left corner of the rectangular region.

Image acquisition was done in a simple continuous sequence as shown in Figure 4.1 where the black outline represents the rectangular region on the slide and the red arrows illustrate the direction of image acquisition.

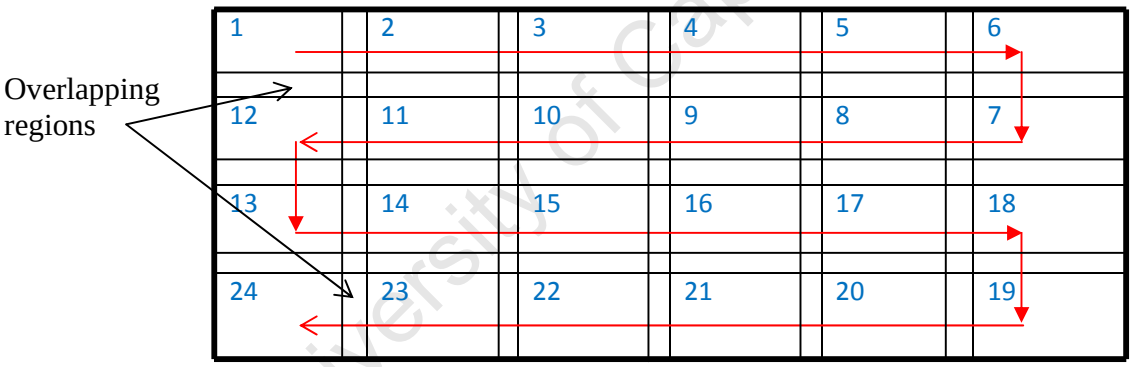


Figure 4.1: Image acquisition sequence assuming 6 images per row.

Due to the manual stage movement, moving precisely to adjacent fields is extremely difficult as explained in Section 2.7. To ensure that it would be possible to assemble an overall image correctly when constructing the virtual slide map, the single images (tiles) were acquired with overlapping regions as shown in Figure 4.1 and were subsequently combined by image stitching.

4.1.2 Assembling the acquired single images by image stitching

Image stitching is an intermediate stage in constructing the virtual slide map and therefore inaccuracies in this stage would carry forward to later stages and might affect

the overall performance of the object recognition scheme. Ideally, there should be no errors in the image stitching process. Both manual and automatic stitching were considered.

4.1.2.1 Manual image stitching

Manually assembling the acquired images to construct a virtual slide map was carried out using Photoshop 7.0, which allows multiple individual images to be copied as different layers onto a large image template. To perform stitching, image 1 and image 2 were copied onto the template as different layers. Layer 1 was moved and fixed near the top-left corner of the template. Upon roughly identifying the overlapping region, the transparency of layer 2 was changed for the visualization of the registration of the two images. Layer 2 was moved along *x*-, *y*-axes onto layer 1 until the identical parts of the two images were completely aligned. They were then reduced to one layer to obtain a merged image. In the same way, the third and subsequent images were added onto the image template to generate the composite image. Since, the camera was aligned to the stage so that objects would move parallel to the x and y axis of the field-of-view, rotation between adjacent images was ignored. The resulting composite image formed the virtual slide map.

4.1.2.2 Automatic image stitching

Automatic stitching of individual images requires automatic image registration i.e. automatic identification of the overlapping regions between two adjacent images and computation of the transformation. This can be achieved using a matching scheme (object recognition scheme) to detect the common feature points between images (Section 2.7.2). Two object recognition schemes, namely GHS (Section 2.5.3) and SIFT (Section 2.5.1), were considered. GHS was considered mainly because the developed object recognition algorithm for auto-positioning was based on the GHS scheme itself. The SIFT scheme was considered because it is a widely used matching scheme in photography (Roth and Winter 2008) and also because Ma et al. (2007) showed that the Autostitch software, which used the SIFT matching scheme, is suitable to stitch microscopy images although they tested only on mouse lymph node and intestine and for

a maximum of 8 images. An understanding of the object recognition algorithm developed – which is based on the GHS scheme - for auto-positioning will facilitate the understanding of auto-stitching using the GHS scheme. Therefore, the entire automatic image stitching procedure is discussed in Chapter 6 after the complete GHS-based object recognition algorithm is presented.

4.2 Decomposition of virtual slide map to generate models

Localisation of a field-of-view (FOV) on a virtual slide could be done using the SIFT scheme (Section 2.5.1). However, a single image of an FOV, which measured 1030 x 1300 pixels, was found to contain over 1000 SIFT keypoints. A virtual slide map which is expected to consist of hundreds of such images would therefore contain an extremely large quantity of keypoints and thus would have high computer storage and RAM requirements. A very large database of keypoints results in a high percentage of false matches of the query keypoints (Lowe 2004). Furthermore, using an exhaustive search method, in which a query keypoint is matched by comparing it to the entire database of keypoints and repeating the process for every single query keypoint, would be highly time-consuming (Mehrotra et al. 2010).

Since a 32-bit computer was used, which can use a maximum of only 4 GB of RAM, computer RAM size was a major constraint. To simplify and permit the execution of the matching process under computer RAM constraints, the virtual slide map was broken down into smaller portions referred to as models (Section 2.3.1). Models overlapped adjacent models by exactly one FOV image as shown in Figure 2.3. Since an FOV image, i.e. tile, measured 1030 x 1300, each model overlapped its adjacent left and right model by exactly 1300 pixels and its adjacent top and bottom by exactly 1030 pixels. Consequently, each model was made up of four tiles captured during the scanning phase and measured 2060 x 2600. This ensured that any query (FOV) image acquired for object recognition would be largely contained in at least one model regardless of the slide's orientation.

By decomposing the virtual slide map into smaller models, the SIFT keypoints (Section 2.5.1) can be extracted for every model followed by storing them using the geometric hashing indexing scheme (GHS) to improve efficiency and performance of the SIFT object-recognition scheme as done by Mehrotra et al. (2010). A single model, which is 4 times the size of a single FOV, would contain roughly 4000 SIFT feature points. Therefore, since a model with n feature points has roughly n^3 entries into the database (Section 2.5.3.1), the number of entries into the database of a virtual slide map consisting of 80 models, for example, would be approximately 5.12×10^{12} entries. This database would consequently have high computing requirements particularly RAM size. Therefore, under computer RAM constraints, the SIFT method integrated with the GHS is still unfavourable.

The number of feature points to be extracted per model should not be very high, so as to avoid high computational requirements, but also not very low, as too few feature points make matching unreliable (Lifshits et al. 2004).

This therefore prompted the use of the medial axis transform (Section 2.4.2), subsequent to image segmentation (Section 4.3), to extract feature points. In order for the object recognition scheme to achieve invariance to geometric changes such as translation, rotation and scaling, the models were represented using the geometric hashing scheme (Section 2.5.3) and hence the overall object recognition algorithm is based on the geometric hashing technique – which has also been used for object recognition in other fields of microscopy (Begelman et al. 2006, Lifshits et al. 2004).

4.3 Image segmentation

In ZN sputum smear images, acid-fast bacilli stain red against a blue background (Figure 1.1(b)). Using this a priori information, an image pixel can be classified as a bacillus pixel or a non-bacillus pixel. Each pixel was classified by considering its values in the three channels of the RGB colour space (Khutlang et al. 2010). Although Khutlang et al. (2010) used a combination of pixel classifiers, the improved accuracy over one classifier was small. Therefore, the segmentation of the images for the experiments was done

using only the quadratic pixel classifier, found by Khutlang et al. (2010) to perform best in segmenting candidate bacillus objects.

The classifier requires training using image pixels as objects. Pixels of bacilli in the focal plane are labelled as +1 while a subset of background pixels are labelled as -1. A quadratic mapping between the objects and their labels is then established. The discrimination between the two classes is drawn using the class mean and covariance matrices. A quadratic decision function (Equation 4.1) is used to assign labels to query image pixels based on the inequality, classifying it as either a bacillus or on-bacillus pixel.

$$\frac{1}{2}\left((X-M_1)^T\Sigma_1^{-1}(X-M_1)\right)-\frac{1}{2}\left((X-M_2)^T\Sigma_1^{-1}(X-M_2)\right)+\frac{1}{2}In\frac{|\Sigma_1|}{|\Sigma_2|} \quad > / < \quad In\frac{P_1}{P_2}$$

Equation 4.1

where X is an object feature vector, M is the mean vector, Σ is the covariance matrix, and P_1 and P_2 are prior probabilities of the classes (Khutlang 2009).

Filters were used to remove non-bacillus objects from the segmented images and thus to reduce the amount of irrelevant information. Area and eccentric filters were utilized as done by Forero et al. (2006) and Khutlang et al. (2010). Area and eccentricity boundary descriptors of every object were extracted from the segmented images and if the object area or eccentricity did not fall within the threshold values, they were discarded.

The objects remaining in the image after filtering were used for subsequent image processing. Despite the filtering process, some inappropriately segmented background areas may persist. These play an important role when images contain very few bacilli. These non-bacillus objects can be used to generate feature points and allow better voting results since an insufficient number of feature points in an image may result in false matches or no match at all (Lifshits et al. 2004).

4.4 Extraction of feature points

The medial axis transform (Section 2.4.2) was employed to extract the topology skeletons of the objects in the filtered segmented images. The resulting branch points acted as the feature points. However, owing to the simple, long and thin shape of TB bacilli, skeletons of many bacilli are a small single curve i.e. branchless. For these objects, the mid-point of the branchless skeletons was used as a feature point. The branch points in addition to the mid-points of the branchless skeletons ensured that every object present in the filtered segmented image is represented by at least one feature point and hence every object contributed to the invariant description of the image, which was done using the geometric hashing technique (Section 2.5.3.1). Figure 4.2 summarises the process for feature point extraction on a zoomed sub-image. The red dots represent the extracted feature points.

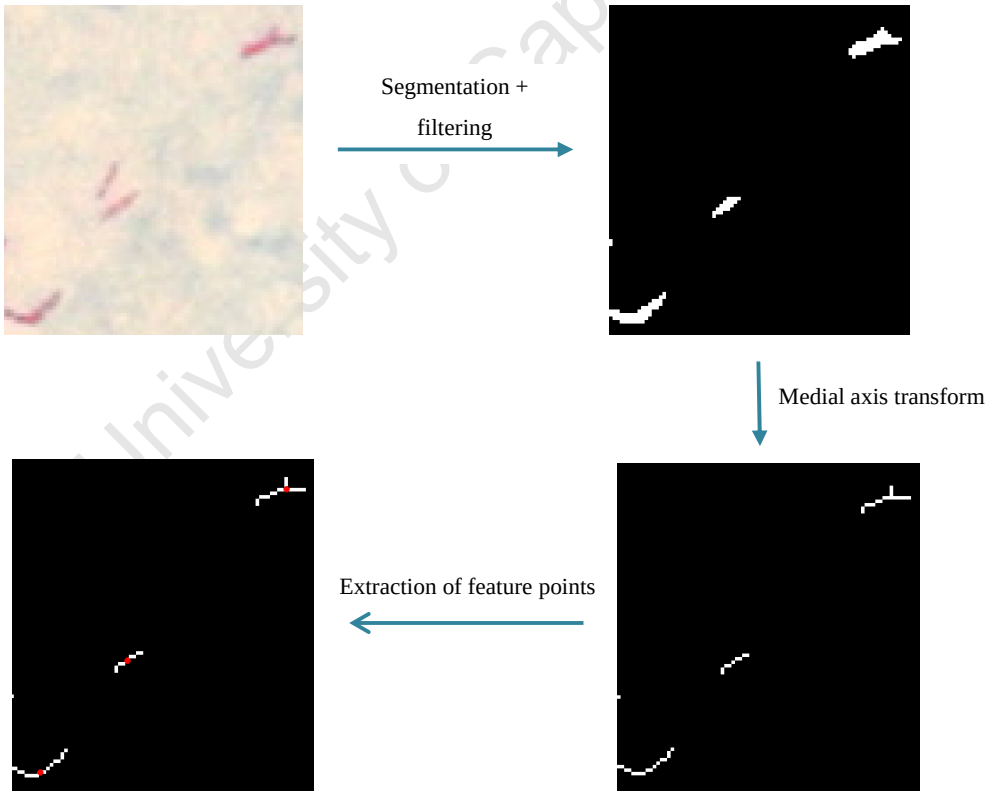


Figure 4.2: Feature point extraction.

4.5 Model representation

In microscopy, typical image variations include illumination changes and geometrical changes such as translation, rotation and scaling. Geometric changes are due improper placement of the slide in the slide holder. In order to make the recognition scheme robust, the model needs to be represented so that it is invariant to these changes.

An invariant description for every model was created as done in the geometric hashing technique (Section 2.5.3.1).

For a selected ordered pair of basis points, for example, m_1 and m_2, the invariant coordinates (α_j, β_j) of the remaining feature points, m_j, $3 \leq j \leq k$ satisfy the equation:

$$m_j = \frac{m_1 + m_2}{2} + \alpha_j (m_2 - m_1) + \beta_j (m - m_1)$$

Equation 4.2

Equation 4.2 essentially re-scales the image such that the magnitude of the vector m_1m_2 is equal to 1 (Wolfson and Rigoutsos 1997). The origin of the image was taken to be the top-left most point of the image. The position of every feature point m_j in the image was therefore, given by the row number and column number of it in the image:

$$m_j = \begin{pmatrix} r_j \\ c_j \end{pmatrix}; \text{ therefore } m_1 m_2 = m_2 - m_1 = \begin{pmatrix} r_2 - r_1 \\ c_2 - c_1 \end{pmatrix} = \begin{pmatrix} R \\ C \end{pmatrix}$$

$$\text{Then } m_1 m = m - m_1 = A \begin{pmatrix} R \\ C \end{pmatrix} = \begin{pmatrix} -C \\ R \end{pmatrix}$$

where A is the rotation matrix $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ that rotates the vector $\boldsymbol{m_1m_2}$ by 90 degrees.

Substituting in Equation 4.2 gives:

$$\begin{pmatrix} r_j \\ c_j \end{pmatrix} = \frac{1}{2} \begin{pmatrix} r_1 + r_2 \\ c_1 + c_2 \end{pmatrix} + \alpha_j \begin{pmatrix} R \\ C \end{pmatrix} + \beta_j \begin{pmatrix} -C \\ R \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} R & -C \\ C & R \end{pmatrix}\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} = \begin{pmatrix} r_j \\ c_j \end{pmatrix} - \frac{1}{2}\begin{pmatrix} r_1 + r_2 \\ c_1 + c_2 \end{pmatrix} = \begin{pmatrix} U_j \\ V_j \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} = \begin{pmatrix} R & -C \\ C & R \end{pmatrix}^{-1}\begin{pmatrix} U_j \\ V_j \end{pmatrix} \qquad \textbf{Equation 4.3}$$

where $U_j = r_j - 0.5(r_1+r_2)$ and $V_j = c_j - 0.5(c_1+c_2)$

The invariant coordinates (α_j, β_j) of the feature points with respect to the coordinate frame formed by $\boldsymbol{m_1m_2}$ were computed using Equation 4.3 - a derivation of Equation 4.2. The entry (M_i, m_1, m_2) was then stored in the hash table at indices (α_j, β_j); $3 \leq j \leq k$ where k is the number of feature points in the model image. This process was performed for every possible ordered pair of points (bases) in the model image and the entire process was performed for all the model images of the virtual slide map.

4.6 Database construction

The database was constructed using hash tables, which allow fast insertion and retrieval of items – (M_i, m_μ, m_v). A two dimensional cell array in MATLAB was employed to represent a hash table. The invariant co-ordinates (α_i, β_i) were used as indices to the 2D cell array: α_i acted as the row index while β_i acted as the column index.

The size of the 2D cell array i.e. the hash table was predicted to simplify its implementation. If a chosen basis is formed by say m_1 and m_2 which are feature points lying at adjacent pixels in the image, then the image is not rescaled when applying Equation 4.2 since the magnitude of the vector $\boldsymbol{m_1m_2}$ is already 1. Under this condition, the maximum possible value of a coordinate would be equal to the diagonal length of the image. The size of a single model image was 2060 rows of pixels by 2600 columns of pixels resulting in a maximum diagonal length of 3317. Consequently the largest dimension of the 2D cell array used measured 3500.

4.6.1 Division of each cell in the hash table into 4 bins

A given 2D co-ordinate frame formed by a basis pair m_μ and m_v, has four quadrants as shown in Figure 4.3(a). Accordingly, each cell of the cell array was divided into four sections - each section representing one quadrant. Since each entry is of the form (M_i, m_μ, m_v) i.e. a 1 x 3 array, each cell contained an array with 3x4 = 12 columns. Therefore, a single cell in the hash table is an N x 12 array equivalent of 4 bins in a hash table - N is the number of entries in that cell and each bin corresponds to 3 columns. This is summarised in Figure 4.3 and Table 4.1.

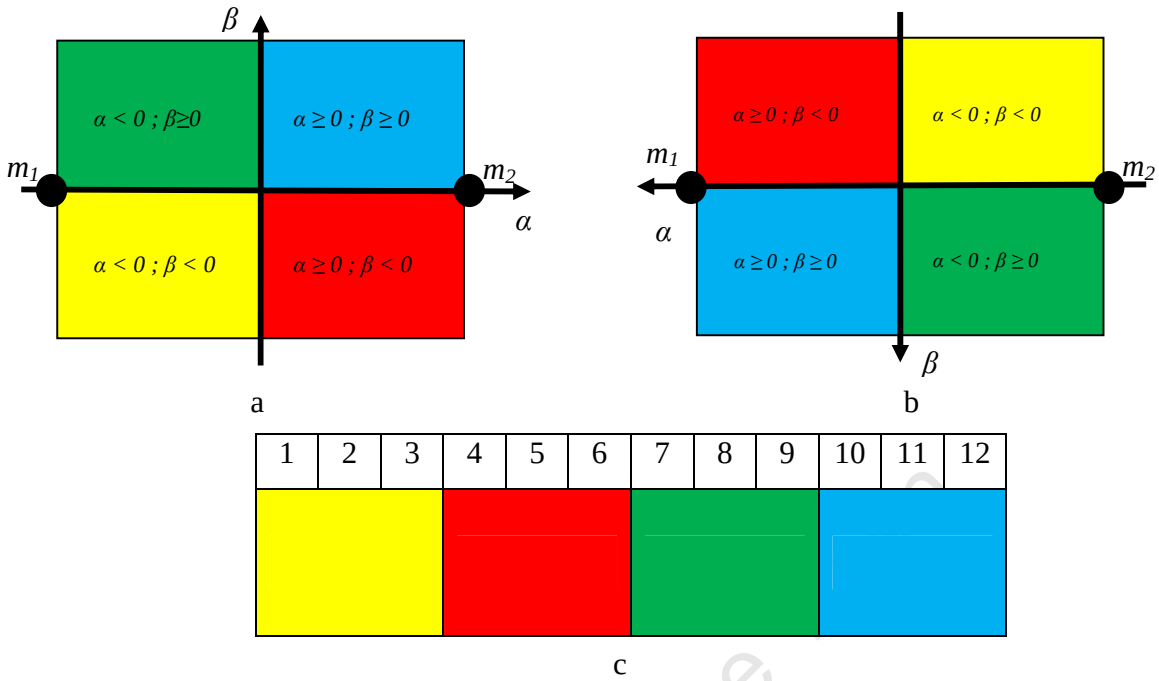


Figure 4.3: Hash table cell organization; (a) coordinate frame m_1m_2 (b) coordinate frame m_2m_1 (c) cell division into 4 bins.

Table 4.1: Indices to hash table to store the entry, (M_i, m_μ, m_v) based on which region the feature point m_j lies in in the coordinate frame $m_\mu m_v$.

Region in which feature point m_j lies	Resulting co-ordinates (α, β)	Index to hash table (2d cell array)	Bin columns
yellow	$\alpha < 0 ; \beta < 0$	(α, β)	1-3
red	$\alpha \geq 0 ; \beta < 0$	(α, β)	4-6
green	$\alpha < 0 ; \beta \geq 0$	(α, β)	7-9
blue	$\alpha \geq 0 ; \beta \geq 0$	(α, β)	10-12

4.6.2 Database formation with 4 hash tables

The computed co-ordinates (α, β) are not necessarily whole numbers and thus cannot be directly used as indices to the 2D cell array. The coordinates were therefore rounded to the nearest whole number. This brings in a complication if the co-ordinate after rounding

produces zero ('0') as '0' cannot be indexed to an array. To overcome this problem, each coordinate was rounded followed by incrementing it by 1 unit prior to using it as an index into the hash table.

If the selected basis points, m_μ and m_ν, are widely separated, then most of the feature points would lie near the origin of the formed coordinate frame. Under this condition, the feature points are less sensitive to noise (Wolfson and Rigoutsos 1997) and thus very small neighbourhood regions need to be considered during voting (Section 2.5.3.2). Therefore, more accurate coordinates of feature points lying near the origin of a given coordinate frame would be beneficial to the scheme's performance. Consequently, if the magnitude of a coordinate was less than 10, two decimal places were considered otherwise they were neglected.

Therefore, the hashing function, F, shown in Table 4.2 was used depending on the magnitude of a coordinate, c, where c is either α or β:

Table 4.2: Hashing Function.

Magnitude of coordinate, c	Hashing function
$\|c\| \leq 10$	$F = round(100 \times c) + 1$
$\|c\| > 10$	$F = round(c) + 1$

The maximum index value when $|c| \leq 10$ is *round (100* x *10.00) + 1 = 1001* while when $|c| \geq 10$ is *round (3317) + 1 = 3318*. In the latter case, the maximum was taken to be 3500.

Serval complications arise using this hash function; for example coordinates (9.85, 2.58) and (985,258) would both be returned as (986, 259) by the hash function. If a single hash table was used, then entries corresponding to coordinates (9.85, 2.58) or (985, 258) would be stored in the same bin with indices (986, 259). Therefore, using a single hash table would result in bins comprising entries which correspond to different coordinates and hence would ultimately result in false matches during object recognition.

To overcome these difficulties, the database was divided into 4 hash tables and the entry (M_i, m_μ, m_ν) was stored in the corresponding hash table dictated by the magnitudes of the coordinates, α and β, as shown in Table 4.3. The size of a hash table was derived based on the maximum index values for that category.

Table 4.3: Database division into 4 hash tables and insertion based on magnitude of α and β.

magnitude of the coordinates (α, β)	**Hash table 1 Size: 1001 x 1001**	**Hash table 2 Size: 3500 x 1001**	**Hash table 3 Size: 1001 x 3500**	**Hash table 4 Size: 3500 x 3500**
$\lvert\alpha\rvert < 10 ; \lvert\beta\rvert < 10$	●	○	○	○
$\lvert\alpha\rvert \geq 10 ; \lvert\beta\rvert < 10$	○	●	○	○
$\lvert\alpha\rvert < 10 ; \lvert\beta\rvert \geq 10$	○	○	●	○
$\lvert\alpha\rvert \geq 10 ; \lvert\beta\rvert \geq 10$	○	○	○	●

4.6.3 Speeding up the database filling process

For a given basis pair points, m_1 and m_2, in model M_i, there are two possible co-ordinate frames: one in which the positive x-axis is formed by the vector $\boldsymbol{m_1m_2}$ (Figure 4.3(a)) and the other in which the positive x-axis formed by the vector $\boldsymbol{m_2m_1}$ (Figure 4.3(b)). These coordinate frames can be named by the vector forming the positive x-axis i.e. coordinate frame $\boldsymbol{m_1m_2}$ and coordinate frame $\boldsymbol{m_2m_1}$ respectively. Comparing the two figures, it can be seen that there is a relationship between the two co-ordinate frames; for example, the blue region in coordinate frame $\boldsymbol{m_1m_2}$ will always become the yellow region in coordinate frame $\boldsymbol{m_2m_1}$, the green region in coordinate frame $\boldsymbol{m_1m_2}$ will always become the red region in coordinate frame $\boldsymbol{m_2m_1}$ and so on. Consequently, if the coordinates of a third feature point, m_j, are (α, β) in the coordinate frame $\boldsymbol{m_1m_2}$, then the coordinates of that point in the coordinate frame $\boldsymbol{m_2m_1}$ will be $-(\alpha, \beta)$. The coordinate '0' remains unchanged in both the coordinate frames.

In general, for a given model M_i and a basis pair selection m_u and m_v, the coordinates (α_j, β_j) of the remaining feature points in that model were computed only in the coordinate frame m_um_v. Their coordinates in the coordinate frame m_vm_u were indirectly determined based on the above relationship. Therefore, for every computed (α, β) ,two items - $(M_i,$

m_μ, m_ν) and (M_i, m_ν, m_μ,) - were entered into corresponding bins of the correct hash table in the database, hence speeding up the database filling process.

Table 4.4 shows in which bins (columns) the two items were stored depending on the signs of (α, β).

Table 4.4: Filling the hash table by only computing invariant coordinates in coordinate frame $m_u m_v$.

coordinates (α, β) **in coordinate frame $m_u m_v$**	**Entries at index (α, β) to correct hash table in database**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
$\alpha < 0 ; \beta < 0$	M_i	m_u	m_ν							M_i	m_ν	m_u
$\alpha > 0 ; \beta < 0$				M_i	m_u	m_ν	M_i	m_ν	m_u			
$\alpha < 0 ; \beta > 0$				M_i	m_ν	m_u	M_i	m_u	m_ν			
$\alpha > 0 ; \beta > 0$	M_i	m_ν	m_u							M_i	m_u	m_ν
$\alpha > 0 ; \beta = 0$							M_i	m_ν	m_u	M_i	m_u	m_ν
$\alpha = 0 ; \beta > 0$				M_i	m_ν	m_u				M_i	m_u	m_ν
$\alpha = 0 ; \beta = 0$										M_i	m_u	m_ν
										M_i	m_ν	m_u
$\alpha < 0 ; \beta = 0$							M_i	m_u	m_ν	M_i	m_ν	m_u
$\alpha = 0 ; \beta < 0$				M_i	m_u	m_ν				M_i	m_ν	m_u

4.6.4 Number of unique entries and total number of entries in database per model

For a model image, M_i, with n feature points, there are $\binom{n}{2}$ unique pairs of points (Section 2.5.3.1). For each pair, there are two possible coordinate frames since they are two possible directions of the positive x-axis (Figure 4.3) and hence a pair of points can be considered as two unique bases. For each pair of points m_μ and m_ν – which are two

unique bases, $m_\mu m_v$ and $m_v m_\mu$ - the coordinates (α, β), of each of the remanning $(n-2)$ feature points are computed only in the coordinate frame $m_\mu m_v$ and used to enter the unique entries (M_i, m_μ, m_v) and $(M_i, m_v, m_\mu,)$. A unique basis therefore corresponds to a unique entry in the database. Therefore, the number of unique entries and the total number of entries into to the database for a model image with n feature points are:

$$\text{No. of unique entries} = \text{No. of unique bases} = \binom{n}{2} \times 2 = \frac{n(n-1)}{2} \times 2 = n(n-1)$$

$$\text{Total no. of entries} = \text{No. of unique bases} \times (n-2) = n(n-1)(n-2)$$

For example a model image with 100 feature points would have 9900 unique entries and 970,200 entries in total in the database.

4.7 Summary of model representation and database construction

To store an invariant description of a model image, M_i, in the database, a unique pair of feature points m_μ and m_v, in the model was selected and the invariant coordinates (α, β), of each of the remaining feature points in the model were computed. The magnitudes of the coordinates (α, β) served two purposes; they were used to determine the hash table in the database in which to store the corresponding entries (M_i, m_μ, m_v) and $(M_i, m_v, m_\mu,)$; and secondly, they were used as indices to determine the cell in that hash table in which to store the entries. The signs of the coordinates (α, β) were then used to determine the columns (bins) in that cell in which to store the entries i.e. the entries (M_i, m_μ, m_v) and (M_i, m_v, m_μ) were both stored in the hash table at index (α, β) but in different columns based on the signs of (α, β) as shown in Table 4.4. The process was repeated for all possible unique pairs of points in the model.

Similarly, the invariant descriptions of all the other models of a virtual slide map were stored in the database. Many hash table bins will receive more than one entry. These bins will therefore each contain a list of entries of the form (M_i, m_μ, m_v).

Figure 4.4 shows a flow chart summarising the database construction process.

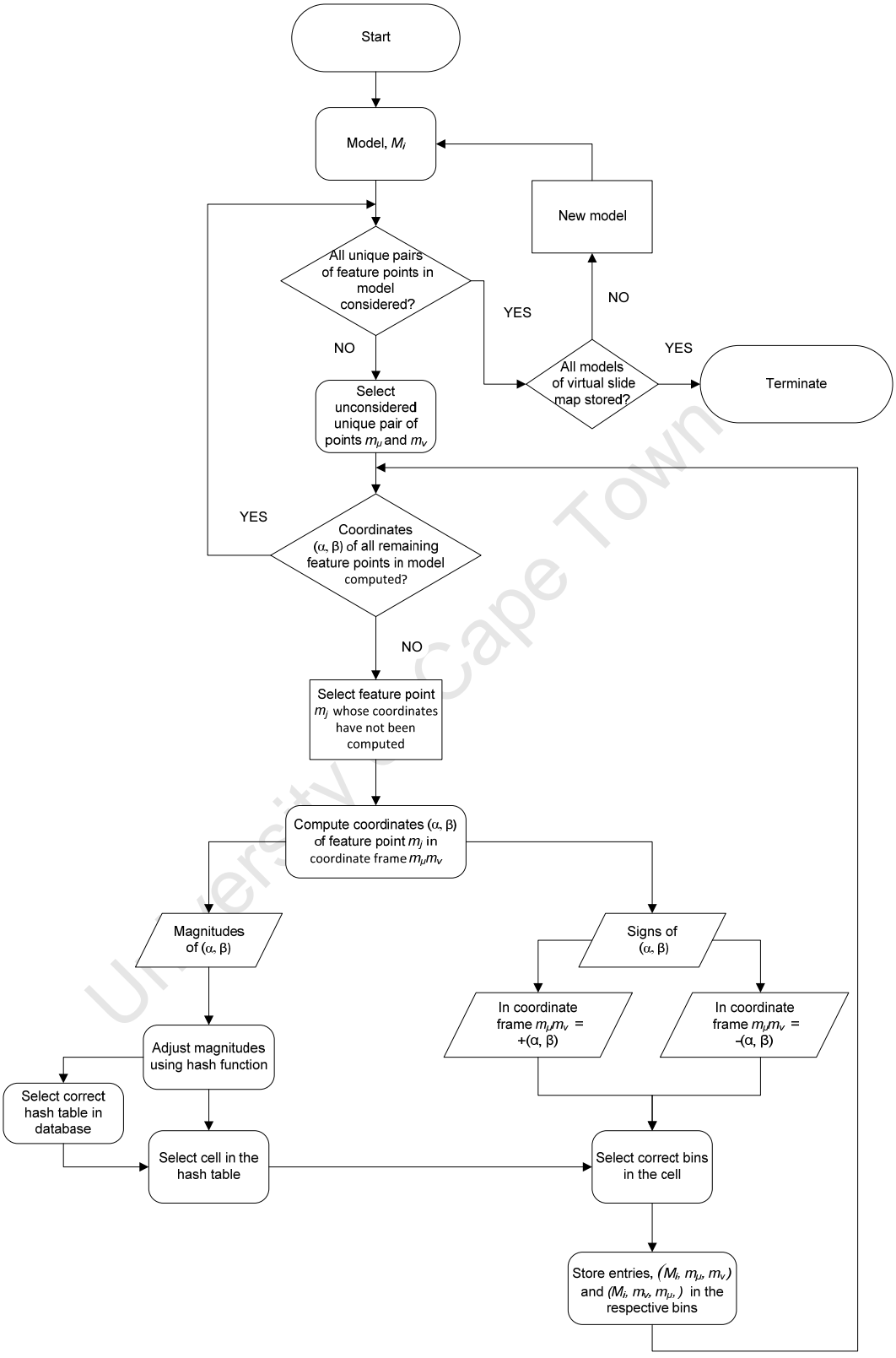


Figure 4.4: Database construction process.

5. Methods: Online localisation stage

The *online localisation stage* involves the recognition of the query image which in this case is the current field-of-view (FOV). Recognition of the query image includes localisation of the FOV image on the virtual slide map, which is equivalent to finding the matching model (in the database) of the FOV image and computing the similarity transformation relating the two images – image registration. The time taken by this stage is much less than for the *offline pre-processing stage* and it is this low online complexity that determines the actual time taken for object recognition. This stage can be sub-divided into three steps, namely processing of the FOV image, indexing and voting and finally verification. In addition to describing the *online localisation stage*, this chapter presents the techniques that were employed to assess the performance of localisation and image registration.

5.1 Processing the FOV image (query image)

To localise a FOV image on the virtual slide map, it was segmented and feature points were extracted exactly as done for the models (Sections 4.3 and 4.4). It was then represented similarly to the model representation (Section 4.5) but using only a single arbitrary basis pair of feature points, B_q. The invariant co-ordinates of the other feature points in the image were then computed in this coordinate frame. These invariant co-ordinates were used to execute the indexing and voting stage after which verification was performed.

5.2 Indexing and voting

The voting step was executed by using the computed invariant co-ordinates (α_j, β_j) as indexing keys to access the correct hash table bins and voting for the entries in them.

The coordinates (α_j, β_j) were adjusted by passing them through the hash function in Table 4.2. If the adjusted coordinates of a feature point in the coordinate frame of B_q were (α, β), the magnitudes of the co-ordinates, $|\alpha|$ and $|\beta|$, were used to extract an N x12 cell from the correct hash table in the database. The signs of the co-ordinates (α, β) were used to extract the correct bin out of the 4 bins present in the cell.

Due to the presence of noise, there is some error in the extracted values of the coordinates, which in turn may result in accessing incorrect cells and hence incorrect bins of the hash table. However, the 'correct' cells are in the neighbourhood of these 'wrong' cells (Wolfson and Rigoutsos 1997). In order to ensure the 'correct' cell (and therefore correct bin) was included in the voting, instead of extracting only the cell at index $(|\alpha|, |\beta|)$, all the cells in the rectangular region centered at index $(|\alpha|, |\beta|)$ were extracted followed by voting of entries in the appropriate bins in these cells.

The above may be explained as follows: assume the basis B_q made up of q_1q_2 corresponds to the basis m_1m_2 in model M. Assume there is no positional inaccuracies between these corresponding basis points but there is between the feature points q_3 , q_4, q_5 and its corresponding points m_3 , m_4, m_5 in the matching model. As seen in Figure 5.1, the corresponding points in the model however lie in the small neighborhood of the query feature points.

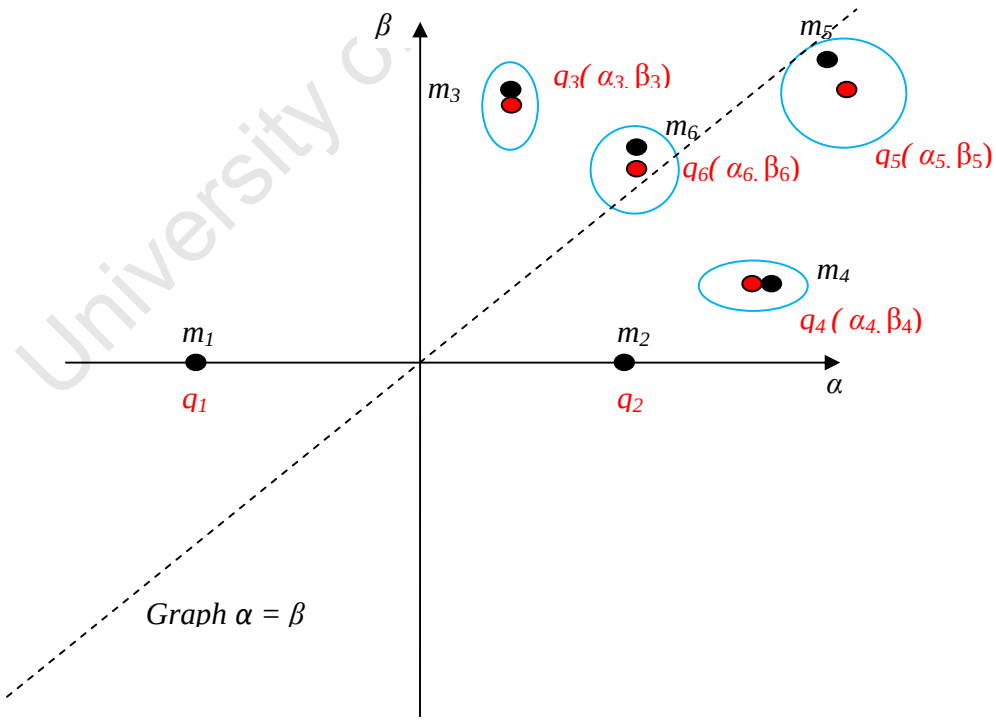


Figure 5.1: Neighborhood regions considered owing to positional inaccuracies induced by noise.

The size of the neighborhood region considered was not fixed. Wolfson and Rigoustsos (1997) showed that the closer the feature points in an image are to the origin of the formed coordinate frame, the less noise-sensitive is the image description. Therefore, the closer a feature point, with coordinates *(α, β)*, to the centre of the origin (i.e. smaller $|\alpha|$ and $|\beta|$), the less noise sensitive it is and the further it is the more noise sensitive it is. Based on this finding, the size of neighborhood region considered was varied proportionally to the magnitudes of α and β. This relationship also alters the shape of the neighborhood region. The closer the feature point lies to the graph $\alpha = \beta$ the more circular the neighborhood region and the further away from it the more elliptic the region as shown in Figure 5.1.

These elliptic or circular regions in the coordinate frame translate into a rectangular region in the hash table centered at the indices *(|α|, |β|)*. This is because the co-ordinates need to be rounded to be used as indices to arrays and secondly arrays themselves are rectangular. Consequently, for easy understanding, neighborhood regions are considered to be rectangular regions.

For a given *(α, β)*, the α component acts as the row index and β acts the column index. The range of variations, δ_α and δ_β, were used to generate the boundaries of the rectangular region to be accessed from the hash table:

$$\delta_\alpha = p \times |\alpha| \qquad \text{and} \qquad \delta_\beta = p \times |\beta|$$

where p is permissible error which was an empirically determined constant. For all the experiments the value of $p = 0.05$ was used.

The rows occupied by the desired rectangular region in the hash table therefore ranged from $\alpha - \delta_\alpha$ to $\alpha + \delta_\alpha$. This range is denoted as h_α. And the columns occupied by the desired rectangular region in the hash table ranged from $\beta - \delta_\beta$ to $\beta + \delta_\beta$. This range is denoted as h_β.

Both the lower and upper limits of h_α *and* h_β were rounded since indices need to be whole numbers. For example if *(α, β) = (40,60)* and taking p to be 5 %, then $\delta_\alpha = 2$, and

$\delta_\beta =$ 3 and h_α would range from *40 –2 = 38* to *40 + 2 = 42* and h_β would range from 60 – 3 = 57 to *60 + 3 = 63.* Therefore, the rectangular region of the hash table to access for a query feature point with coordinates *(40,60)* is shown in Figure 5.2.

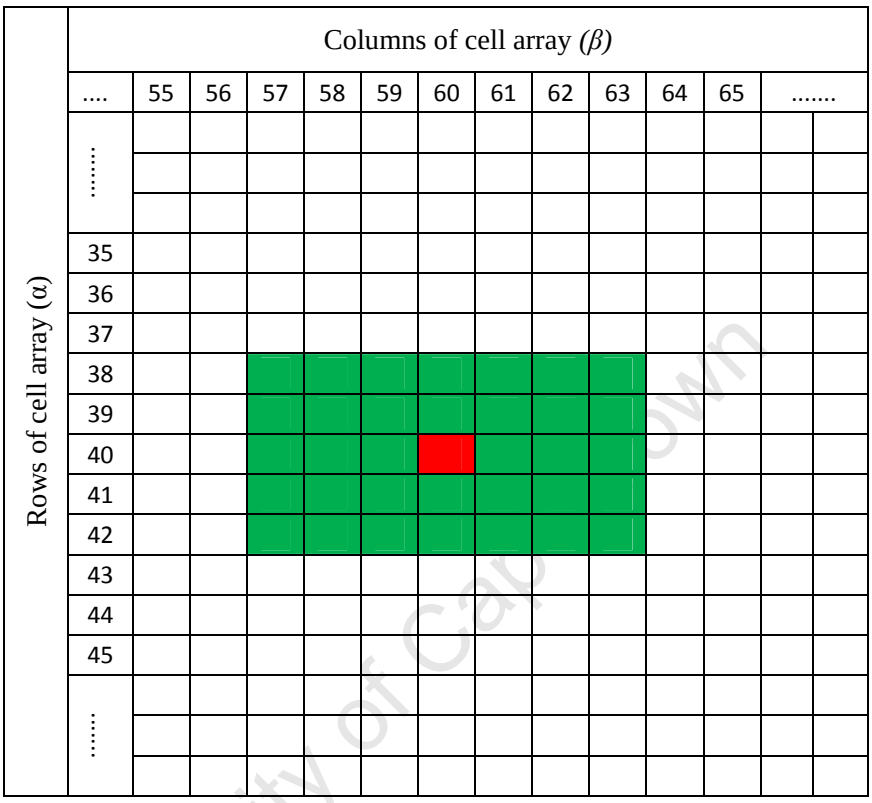


Figure 5.2: Rectangular region of hash table accessed for a particular *(α, β)*.

The magnitudes of α and β were used to extract the cells-of-interest (coloured in Figure 5.2) from the correct hash tables (Table 4.3) and their signs were used to extract the correct bins (Table 4.1) from each of these cells. All the entries (M_i, m_μ, m_v) in these bins were given a vote. The more entries in the bins the longer it takes to execute the voting process.

The process was repeated for all the coordinates (α_j, β_j) of every feature point, q_j , $3 \leq j \leq k$ where k is the number of feature points in the query image, Q. The total time taken by the voting stage is dependent on the total number of entries in the accessed bins.

The entries, which were of the form (M_i, m_μ, m_v), that accumulated a significant number of votes were then taken to the verification stage. These formed the candidate model -

basis combinations (*CMB*) that are possible matches to the query image *Q* (Section 2.5.3.2). This candidate list can be referred to as *CL1*. A single model can have several basis pairs in it that may be candidate matches but only one or none will be a best match to B_q and therefore all the *CMBs* in *CL1* need to be verified.

5.2.1 Reduction of the verification load

The verification process on the entire *CL1* may be highly time-consuming. Several techniques were applied to reduce the verification burden:

5.2.1.1 Sorting of the candidate list

CL1 was sorted in descending order of votes received. The best matching *CMB* is likely to accumulate a significant number of votes and hence would lie near the top of this sorted *CL1*. If V_{max} was the maximum vote accumulated, then only *CMBs* that received at least $V = f \times V_{max}$ votes, where f was empirically determined, were considered for subsequent verification. For all the experiments the value of $f = 0.4$ (i.e. 40%) was used. Consequently, only a top fraction of the sorted *CL1* is carried forward to the verification stage. This new candidate list can be referred to as *CL2*. *CL2* was further shortened by a filtering process.

5.2.1.2 Filtering out *CMBs* highly unlikely to match basis B_q in query image *Q*

Due to the nature of the entries in the database - (M_i, m_μ, m_v) - the voting stage provides both the candidate model and the candidate basis pair in that model corresponding to the basis B_q of query image *Q*. These corresponding basis pairs were used to compute the parameters (t_x, t_y, s, θ) of the approximate similarity transformation, $\hat{T}$ mapping the query image, *Q* to the candidate model in *CL2* since a similarity transformation can be solved using two point-to-point correspondences (Section 2.5.3.2).

CL2 was further trimmed by filtering out *CMBs* that are highly unlikely to be matches to *Q*. *CL2* filtering was carried out by employing an allowable range of rotational angles and scale factors between the candidate model and the query image.

The thresholds of the range were determined as follows:

Scale factor, s – Since images were always taken at 40x magnification, the scale factor between the query image and best matching model should to be 1 theoretically. However, practically, due to noise, there are positional inaccuracies in the corresponding basis points and thus the scale factor will not be exactly 1 but will be very close to 1. Thus a scale factor range of 0.85 – 1.5 was used. If the computed scale factor, *s*, between a given *CMB* and B_q of *Q* did not fall in this range, then that *CMB* was discarded.

Rotational angle, θ - According to Begelman et al. (2006), manual slide placement can result in rotational angles between query image and model image of up to 12 degrees. This however depends on the mechanical design of the slide holder of the microscope used. Therefore, to account for large image rotations, a wide range of up to 30 degrees was selected. If the magnitude of the computed rotational angle, θ, between a given *CMB* and B_q of *Q* was greater than 30° then that *CMB* was discarded.

The filtering process considerably shortened *CL2* and hence only a few *CMBs* needed verification.

5.3 Verification

For a given *CMB* in *CL2*, the already computed transformation approximation $\hat{T}$ was applied to all the feature points, q_j, in *Q* to find their corresponding points in that candidate model with the help of Voronoi tessellation and Delaunay triangulation (Section 2.5.3.2). The set of corresponding points formed the putative set required to execute RANSAC (Section 2.6) to estimate the transformation which is free of outliers. The distance threshold used in RANSAC was $t = 10$, which was empirically determined.

The estimated transformation, $\hat{H}$, that gave the largest number of inliers between the candidate model and the query image was declared the best transformation obtained by RANSAC for those two images. To improve the estimation, the inliers corresponding to that best $\hat{H}$ obtained from RANSAC were used to re-compute least-squares-fit

transformation, *T*, mapping query image *Q* to that candidate model as suggested in (Hartley and Zisserman 2003).

This process was repeated for every *CMB* in *CL2*. The scale and orientation filters were once again applied to the computed transformations' parameters. A *CMB* was eliminated if the corresponding transformation parameters did not fall within the thresholds (Section 5.2.1.2) of these filters.

Out of the remaining *CMBs*, the candidate model that resulted with the highest number of inliers was declared the best match to the query image, *Q*, and the corresponding *T* the best estimate of the transformation relating the two. Therefore, the algorithm produces the matching model and the registration parameters simultaneously.

Due to noise in the query image, the algorithm may fail to match it to a model for the chosen basis, B_q, as explained n Section 2.5.3.2. When this occurred, another attempt was made by re-executing the algorithm with another arbitrary B_q from the query image. Several attempts were allowed to accommodate considerably noisy images. If the query image was not matched within a maximum allowable number of attempts, it was declared a miss.

5.4 Summary of query image object recognition

To localise the current FOV (query image) on the virtual slide map, it is segmented, filtered and feature points are extracted. An arbitrary basis pair, B_q, is selected and the coordinates of the remaining features points computed. These coordinates are used to extract rectangular regions from the appropriate hash tables in the database and every entry in the accessed bins is voted for. The candidate list is formed and trimmed using the angle and scale filters. The resulting candidate model-basis combinations (*CMBs*) are then carried forward for verification. The candidate model sharing the highest number of inliers with the query image is declared the best matching model with the corresponding least-squares fit transformation. If no matching model is found by the algorithm within a

maximum allowable number of attempts, it is declared the miss. Figure 5.3 shows a flow chart summarising the query image object recognition process.

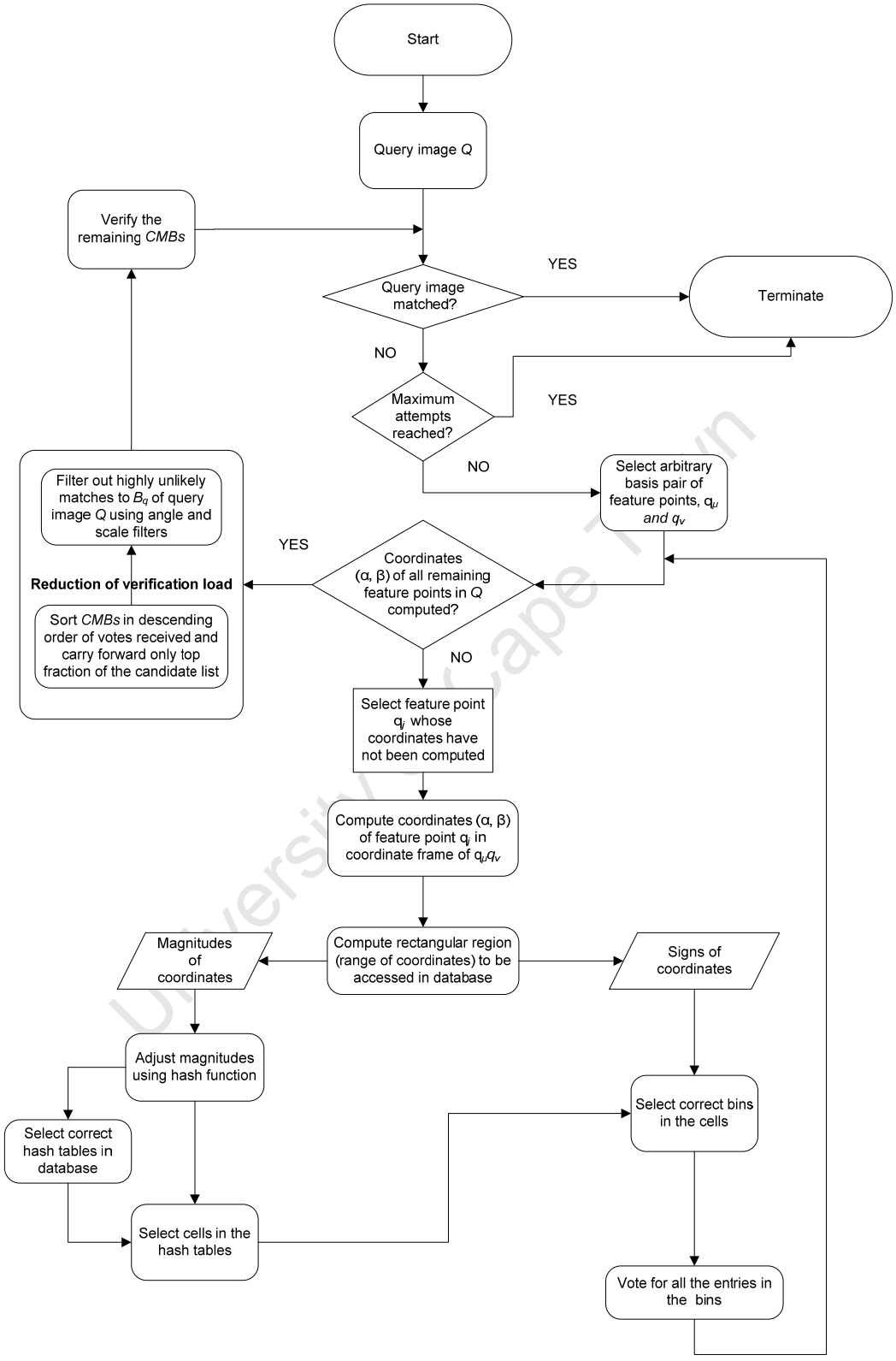


Figure 5.3: Query image object recognition process.

5.5 Performance assessment

Object recognition performance was assessed for the two components, namely localisation of the query image on the virtual slide map and image registration. Recall that localising the query image on the virtual slide map is equivalent to finding the model in the database that best matches the query image (Section 2.3.1). Hence the performance of the localisation task is directly related to the performance of the object recognition task.

5.5.1 Evaluation of object recognition (localisation)

The performance of object recognition was measured using the hit rate and the discriminative power.

5.5.1.1 Hit rate

Two sets of query images were obtained as follows:

Query images Set 1

This set comprised images extracted directly from the virtual slide maps themselves. To form a query image, *Q*, a region measuring 1030 x 1300 pixels (i.e. the same size as a FOV image) was randomly selected from the virtual slide map. Since the query image emerges from the virtual slide itself, it is completely noise-free relative to the matching model and therefore there will be a model in the database that will exactly match *Q*. The object recognition task was then performed to determine which model in the database best matches it. Since the slide was aligned to the x-axis of the XY stage during virtual slide map creation, the rotational angle reported by the algorithm for these query images could be expected to be very close to 0°.

Query images Set 2

For the second set of query images, real images were obtained at a different time to the scanning process. This set of query images comprised several sub-sets of images where each sub-set was obtained at a different time and different orientation of the slide. Different angles were considered for different slides to cover the range from 0° to 26°.

To form a sub-set of images, the slide was placed on the slide holder and rotated. This orientation of the slide was measured relative to the x-axis of the XY stage using a protractor. However, the mechanical blockages and the limited resolution of the protractor prevented an accurate measure of the angle. Therefore, the measured angle was only a rough estimate. Random fields-of-view (1030 x 1300 pixels) lying within the rectangular mark on the slide were then captured at 40x magnification. This is the same magnification as used in the scanning phase (Section 4.1.1). The FOVs were manually focused prior to image capture. About 150 images were captured for each sub-set and all these formed the *Query images Set 2*

Due to the deliberate improper placement of the slide, geometric changes including rotational and translation changes were introduced in the query images relative to its matching model in the database. Furthermore, illumination changes were also present. Additionally, since manual focusing is not repeatable (due to factors including different slide orientation, illumination changes and human judgment), further errors were introduced in the query images. Owing to all these factors the query images in *Query images Set 2* were considerably noisy relative to the model images.

For every query image, the status of the output of the object recognition algorithm was reported based on the model declared as the best matching model as summarised in Table 5.1.

Table 5.1: Object recognition output status.

Object recognition algorithm output	**Object recognition algorithm output status**
Correct model	True Positive
Incorrect model	False Positive
No model	Miss

To determine whether or not the output model was the correct model or not, the colour images of the query image and the reported matching model image were visually compared. Large distinctive features in the images such as big blobs and structures

simplify the visual comparison as the human eye can quickly pick these up. In the absence of these, smaller distinctive features and patterns including smaller blobs and/or groups of bacilli and their relative positions to one another can also be identified without difficulty.

The hit rate (HR) was computed as the true positive rate i.e. HR = TP rate = *number of TP / number of Tests*. The miss rate was computed as MR = *number of misses / number of Tests* and false positive rate was computed as FPR = *number of false positives / number of Tests*. These rates were a measure of the performance of the object recognition scheme.

5.5.1.2 Discriminative power

The discriminative power was also used to assess the performance of the scheme. This was measured by simply finding the rank of the best matching *CMB* (Section 2.8.2) in the sorted candidate list *CL1* (Section 5.2.1.1).

Quadtree enhancement

The algorithm's discriminative power can be improved if prior knowledge of the expected location of the query image on the virtual slide map is available. The experiments with Q*uery images Set 1* were re-performed but this time with quadtree implementation (Section 2.8.2). The virtual slide map was divided into 4 quadrants and only models in the quadrant expected to contain the query image were allowed to participate in the voting stage. Since these query images were extracted from the virtual slide map itself, the exact location of a given query image was known. Therefore, during object-recognition of this query image, only the corresponding quadrant of the virtual slide map was allowed to participate in voting.

5.5.2 Evaluation of image registration

For the evaluation of the image registration, it only made sense to consider the true positives. The accuracy of image registration of a query image to its matching model was evaluated in three ways:

5.5.2.1 Visual assessment

The query image was transformed using the corresponding computed registration parameters and overlaid onto the matching model. The quality of the image registration was then checked by visual assessment and hence was only used as a rough measure of the registration parameters.

5.5.2.2 Mean square error (ms) and root mean square (rms) error

The mean square error (average pairwise error), ε_{avg}, computed as explained in Section 2.8.3.2 was used to measure the image registration error, which incorporates the errors in all the registration parameters *(t_x, t_y, s, θ)*. The root mean square error, $r\varepsilon_{avg} = \sqrt{\varepsilon_{avg}}$, was also computed.

5.5.2.3 Comparative method

Ma et al. (2007) show that the Autostitch software, which uses the SIFT matching scheme, is very efficient in stitching microscope images. This is equivalent to saying the SIFT scheme is a very efficient image registration technique for microscope images since image registration is the core component in image stitching (Section 2.7.2). Therefore, the registration parameters obtained by the method under investigation were compared to those obtained using the SIFT scheme.

The registration parameters using the SIFT scheme can only be obtained after the object recognition algorithm reveals the matching model to the query image. For every query image the registration parameters using SIFT were computed as follows:

1. Once the matching model was found, SIFT keypoints were extracted from the query image and the matching model and compared to find the matching keypoints.

2. The matching keypoints formed a putative set and RANSAC (Section 5.3) was used to eliminate outliers.

3. The resulting inliers were used to compute the least-square-fit transformation.

Using the computed transformation, the mean square error and root mean square error (Section 5.5.2.2) were also computed for comparison.

6. Methods: Automatic image stitching

Since the XY stage is manually driven, the scanning process cannot be automated for the microscope used. However, by automatically stitching the images manually acquired from the scanning process, the construction of a virtual slide map can be semi-automated. Auto-stitching requires the identification of the overlap region i.e. finding the region in one image that matches a region in the other image. This therefore is also an object recognition task.

The scanning process was performed by capturing images of adjacent fields of a slide while ensuring 30-50% overlap in area between them. A smaller percentage of overlap may not provide sufficient common structures between adjacent images, which may cause the auto-stitching process to fail.

Since the images were sequentially captured (Figure 4.1) prior knowledge was available as to which image needs to be stitched to which image. Let the captured images be labelled I_i, $1 \leq i \leq n$, where n is the total number of images captured for that slide. A reference frame was created which was large enough to fully contain the complete virtual slide map. The complete virtual slide, V_n, was constructed by sequentially stitching the next image, I_{j+1}, to the image last stitched to the partial virtual slide, V_j.

$$V_{j+1} = V_j + I_{j+1} \quad ; 0 \leq j \leq n$$ where n is the total number of images captured

The symbol + in this context refers to stitching of images and should not be confused with mathematical addition. The first image, I_1, i.e. when $j = 0$, was placed and fixed near the top-left corner of the reference frame and this forms the first partial virtual slide V_1. This first image represents the field on the slide at the top-left corner of the rectangular region marked on the actual slide (Figure 4.1).

Stitching the last image, I_n, to the partial virtual slide, V_{n-1} , results in the complete virtual slide map i.e. $V_n = V_{n-1} + I_n$.

Errors resulting in stitching I_{j+1} to V_j are carried forward and therefore if stitching is performed in the continuous sequence as showed in Figure 4.1, the gross error accumulated after completion of V_n may be very large especially if n is large. The stitching process may even fail before completion of the virtual slide map. Therefore, the images were relabelled and stitching was performed in a non-continuous sequence (raster-style) as shown in Figure 6.1. The gross error in the virtual slide map constructed in this manner is expected to be less. For example, using the sequence shown in Figure 4.1, the total error present before ,say, I_{r2+1} is stitched, includes all the stitching errors that resulted after stitching all the images in the previous two rows i.e. image I_1 to I_{r2}. However, using the sequence shown in Figure 6.1, the total error present before I_{r2+1} is stitched includes only the stitching error that resulted by stitching I_{r1+1} to I_1. This raster-style configuration was preferred over a spiral configuration, in which images are stitched in a combined vertical and horizontal direction, due to its simplicity. Additionally, if a spiral configuration was used, image re-labelling would be very difficult. This is mainly because the overlap between images varied from 30-50% and therefore, it would be difficult to determine which images overlap one another when moving in a spiral manner.

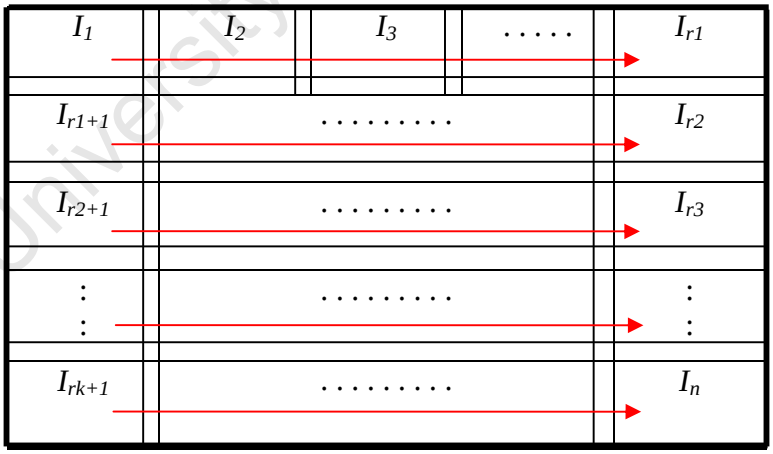


Figure 6.1: Re-labeling of images in a non-continuous sequence.

where I_{rk} is the last image of row k and the total number of rows of images = $R = rk+1$.

Automatic construction of the virtual slide requires automatic identification of the overlap region between V_j and I_{j+1}. An automatic matching scheme that detects common

points between the two images was used to perform this task. However, for larger values of j, V_j is will be very large relative to the image I_{i+1} and therefore the matching scheme would be prone to errors. The freeware available (Section 2.7.2), which were developed for photography, could not be used as they had limitations in the size of images and the number of images that can be stitched. Some of these only allowed stitching images in the horizontal direction and hence could not be used to generate a composite image. Also, they did not seem to perform good stitching. This may be due to the nature of TB sputum smear images. Lastly, the source code could not be found and hence neither could it be modified to perform the sequential stitching process nor could the computed transformation parameters be extracted to allow sufficient performance checks of the stitching process.

6.1 Image stitching using a small portion of the partial virtual slide map

The matching process would be more accurate and quicker if a smaller portion of V_j is considered in the matching process. Since the images were sequentially captured prior knowledge was available as to which image needs to be stitched to which image. As a result, the portion-of-interest, POI_j, of V_j that shares an overlap region with the image, I_{j+1}, to be stitched is known. Therefore, this POI_j can be extracted from the V_j. Using the sequence shown in Figure 6.1 the position of the POI_j varies depending on the current configuration of V_j.

At a given time, V_j can be at one of three possible configurations shown in Figure 6.2. Shown also is the POI_j that needs to be extracted as it shares a common overlap region with the next image to be stitched.

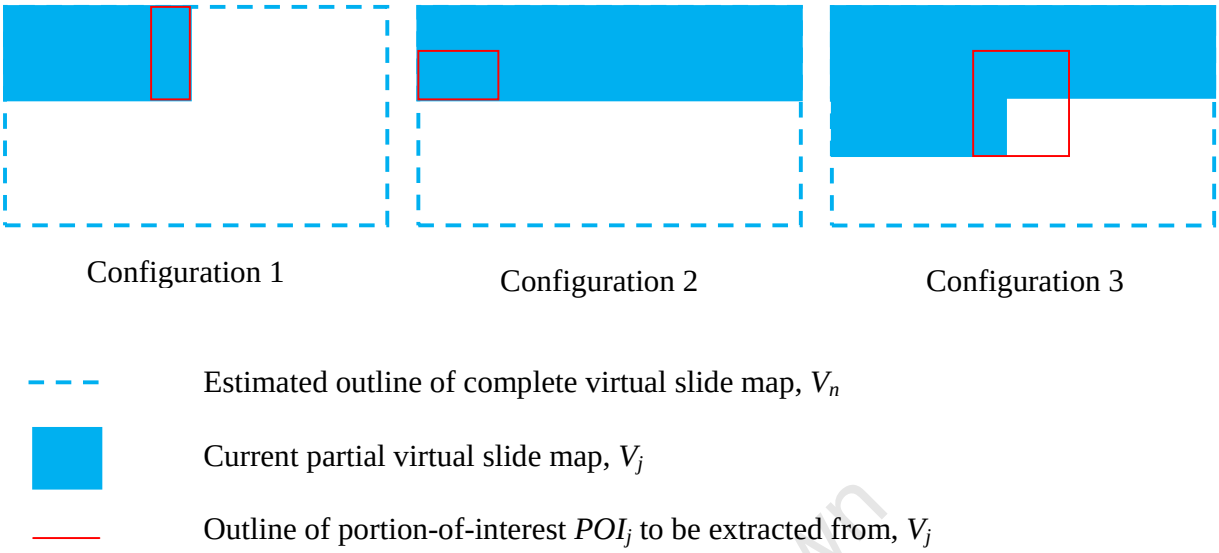


Figure 6.2: The 3 possible configurations of the partial virtual slide map, V_j.

Image stitching at configuration 1

The partial virtual slide map is at this configuration only during the stitching of the first row of images (I_1 to I_{r1}). In this configuration the POI_j is a sub-image of the previous image, I_j, that was last stitched. The POI_j extracted includes the right-half of I_j since a maximum of 50% overlap was allowed between adjacent images. After the first row of images is stitched, this configuration never occurs again.

Image stitching at configuration 2

This configuration occurs after the last image, I_{rk}, of a given row is stitched. The next image $I_{j+1} = I_{rk+1}$ is stitched near the beginning of the subsequent row. The overlap region between V_j and I_{j+1} therefore includes a portion of the first image of the former row i.e. $I_{r(k-1)+1}$. Consequently, the POI_j extracted includes the lower half of $I_{r(k-1)+1}$ since a maximum of 50% overlap was allowed between adjacent images. This configuration occurs whenever a complete row of images is stitched.

Image stitching at configuration 3

This configuration occurs when neither of the above two occurs. Under this configuration, the POI_j extracted includes the right-half of the previously stitched image,

I_j, and the lower half of the image just above the expected occupancy of image I_{j+1}. This configuration is encountered most often.

The extracted POI_j is much smaller then the entire partial virtual slide map and therefore matching can be more simply and accurately performed. Common feature points between the POI_j and I_{j+1} were detected and these were used to compute a transformation linking the two. Prior to the scanning process, the camera was well aligned to the XY stage i.e. objects moved parallel to the *x* and *y* axis of the field-of-view. This guaranteed the correct movement of the stage; and secondly also prevented the individual adjacent images from being rotated in relation to each other during image acquisition. Furthermore all images were taken at 40x magnification and therefore the scale factor between adjacent images is 1. Therefore, the orientation and the scaling can be assumed to stay the same throughout the scanning and processing of the images, and hence the computed transformation relating POI_j and I_{j+1} consists only of translation (parameters t_x and t_y).

Image I_{j+1} was transformed with respect to the reference image, V_j, using the computed transformation. They were then combined to form V_{j+1} by replacing the common overlapping region between them by only that of Image I_{j+1}. This process was repeated until image I_n was stitched to V_j to form V_n, which is the complete virtual slide map of the actual slide.

Automatic identification of the overlap region requires a matching scheme (object recognition scheme) to detect the common feature points between images. Two object recognition schemes, namely the GHS scheme (Section 2.5.3) and SIFT scheme (Section 2.5.1), were considered. The reasons why these two schemes were considered is explained in Section 4.1.2.2.

The suitability of the two methods to stich numerous TB microscopy images was compared by comparing the resulting composite images. The resulting composite images were also tested in object recognition of numerous real images since the ultimate goal of constructing the virtual slide maps was to use them for auto-positioning.

6.1.1 SIFT

At any given time, only two images participate in the stitching process, POI_j and I_{j+1}. Neither of the two is very large and therefore SIFT keypoints (Section 2.5.1) could be extracted from the two images and compared without high computing requirements. The matching keypoints were used to compute the translation parameters of the transformation linking the two images. Image I_{j+1}, was then stitched by replacing the common overlapping region between them by only that of Image I_{j+1}. The process was repeated until the entire virtual slide, V_n, was constructed.

6.1.2 Geometric hashing scheme (GHS)

The entire geometric hashing process explained in Sections 4.3 to 5.3 was carried out to match POI_j and I_{j+1}. POI_j acted as the one and only model, M, while image I_{j+1} acted as the query image, Q. Image POI_j was segmented and filtered (Sections 4.3). Feature points were extracted from the resulting image and used to create an invariant image representation of POI_j as explained in Section 4.5. To execute the matching process, I_{j+1} was segmented, filtered, and feature points extracted. An arbitrary basis, B_q, in I_{j+1} was selected based on the overlap region shared between it and the extracted POI_j which is dependent on the configuration of V_j. This ensured that the overlapping regions were correctly matched. Table 6.1 shows the selection of basis, B_q, depending on the configuration of V_j.

Table 6.1: Basis, B_q, selection based on the configuration of V_j.

Configuration	The pair of feature points forming basis, B_q		Reason
	Location of feature point 1	Location of feature point 2	
1	Left-half	Left-half	only the left half of I_{j+1} overlaps with POI_j
2	Upper-half	Upper-half	only the upper half of I_{j+1} overlaps with POI_j
3	Left-half	Upper-half	Both the left and upper halves of I_{j+1} overlap with POI_j. Therefore, selecting a B_q made up of one feature point lying in the left half and one lying in the upper half of I_{j+1} ensured that that I_{j+1} was stitched accurately to both the image above and left of it

The invariant coordinates (α_j, β_j) of the other feature points in I_{j+1} in the coordinate frame formed by B_q were computed. These coordinates were used to assess the respective hash table bins and the voting process was carried out. The voting stage produces the matching model (which is of course image POI_j since it is the only model) and candidate basis pairs corresponding to B_q .These candidate basis pairs list was filtered to remove highly unlikely matches. The remaining candidate basis pairs were verified as explained in Section 5.3. The inliers obtained from RANSAC were used to re-compute a better estimation of the translation parameters of the transformation in the least-squares sense. The *CMB* with the highest number of inliers was the best match to B_q with the corresponding least-squares-fit transformation. Image I_{j+1} was then stitched by replacing the common overlapping region between them by only that of Image I_{j+1}. The process was repeated until the entire virtual slide, V_n, was constructed.

6.2 Methods for comparison of the GHS auto-stitching scheme to the SIFT auto-stitching scheme

For the microscope system used, an image can be captured with a maximum size of only 1030 x 1300 pixels. A sufficiently large microscopy image could not be obtained to allow extraction of several images from one original image to test stitching quality as

done by Ma et al. (2007). Therefore, the stitching quality of the two methods was compared to one another by testing the two methods on 62 images acquired from a rectangular region covering 1.6 x 1.1 mm^2 area on a slide, named slide O. Each scheme independently produced one complete virtual slide map of slide O (VSO). Several methods were used for this comparison:

6.2.1 Visual inspection

The regions at and near visible seam lines (Section 2.7.2) were visually inspected in the two VSOs.

6.2.2 Triangle method

If the stitching quality of both the methods is same then the corresponding triangles - defined by 3 pairs of corresponding points - in the two VSOs are expected to be congruent. Therefore, the triangle method explained in Section 2.7.2.1 was employed for a quantitative test. A triplet of points was selected in the SIFT VSO and the corresponding points in the GHS VSO were found. The three points were joined to generate a triangle and the ratio between the lengths of the three sides were computed and compared. Congruent triangles also have the same size and shape and hence they also have the same perimeter. Therefore, perimeters of the corresponding triangles were also compared.

6.2.3 Performance in object recognition for auto-positioning

Since the purpose of image stitching was to construct a virtual slide map for the auto-positioning application, object recognition tasks with real images were carried out using both the SIFT VSO and GHS VSO. This test served two purposes; firstly to check if the stitching methods were suitable for constructing virtual slide maps for object recognition for auto-positioning and secondly to indirectly compare the stitching quality of the two VSOs by comparing the performance of the individual VSOs in object recognition.

To obtain real query images, at a different time to scanning, slide O was placed on the slide holder and aligned to the x-axis of the XY stage i.e. slide orientation 0°. The

numerous FOVs captured formed the set of real query images used to perform the object recognition tasks on both the VSOs.

Since the object recognition scheme developed for auto-positioning was based on GHS scheme, the GHS auto-stitching scheme was selected over the SIFT auto-stitching scheme to construct a larger virtual slide map in order to maintain consistency.

7. Methods Summary and New Contributions

The core component in auto-positioning is to form a point of reference on a virtual slide map, a process that may be formulated as an object recognition task. To the authors' knowledge, this is the first work done related to auto-positioning in TB microscopy.

To construct a virtual slide map, the actual slide was first scanned. The microscope lacked a motorised XY stage and an auto-focusing algorithm and therefore the scanning process was manually performed to acquire images of the different fields on a slide. The area covered on the slide was less than the entire smear as larger virtual slide maps demand high computing requirements. The acquired images were then combined, by image stitching, to form the virtual slide map of that slide. The construction of the virtual slide map was semi-automated by auto-stitching. The GHS auto-stitching scheme and SIFT auto-stitching scheme (Brown and Lowe 2007) were considered and compared.

Several adaptations were required to allow the construction of a large virtual slide map from numerous individual images. Instead of matching an individual image to the entire partial virtual slide map, it was only matched to a small portion of the virtual slide map. This not only reduces the processing time but also reduces the number of false matches and hence reduces the chances of failure of the stitching process. Errors introduced in stitching an image to the partial virtual slide map are carried forward and hence instead of stitching images in a single continuous sequence as in Figure 4.1 (in which case the stitching process may even fail before completion of the virtual slide map), the stitching process was conducted in a non-continuous sequential manner (Figure 6.1). This was possible since prior knowledge as to which image overlapped with which image was available. The GHS matching scheme was compared to the SIFT scheme and then it was applied to a large number of images to form a large virtual slide map. In the GHS auto-stitching scheme, to ensure correct stitching of the image to be stitched and the portion of the partial virtual slide map, it is important to select an arbitrary basis, B_q, appropriately depending on the configuration of the partial virtual slide map (Table 6.1).

To the author's knowledge, this is the first application of the GHS matching scheme to stitching microscopy images to re-create a digital image (virtual slide map) of a large section of the slide while retaining the original microscope resolution. Lifshits et al. (2004) and Begelman et al. (2006) used the geometric hashing scheme for localisation after construction of a virtual slide map.

To execute the object recognition process under memory constraints, the virtual slide map was decomposed into smaller portions referred to as models as done in Lifshits et al. (2004) and in Begelman et al. (2006). The model images were first segmented using the quadratic classifier as done in Khutlang et al. (2010). To reduce irrelevant information, most non-bacillus objects in the segmented images were eliminated using area and eccentricity filters as done in Forero (2006). The medial axis transform (Blum 1967) was applied to the remaining objects in the image and the branch points of the resulting skeletons were extracted to act as feature points as done in Begelman et al. (2006). However, due to the simple long and thin shape of TB bacilli, many objects lacked branch points. Therefore, the process was adapted so that for these objects the midpoint of its skeleton was extracted as a feature point. This ensured every object in the filtered segmented image participated in the subsequent invariant description of the image.

The geometric hashing scheme was used to generate invariant - to similarity transformation - descriptions of these model images and to perform query image localisation tasks as done in Liftshits et al. (2004) and Begelman et al. (2006).

The descriptions of the model images of a virtual slide map were stored in a database comprising four hash tables (Table 4.3), where each hash table was a 2D cell array. Each cell in a hash table had 4 partitions equivalent to 4 hash bins to represent each quadrant of a given coordinate frame (Figure 4.3). To the author's knowledge, this is a novel type of database suitable for the geometric hashing technique.

The hash table filling was conducted via appropriate hash functions (Table 4.2) applied to the computed invariant coordinates (α, β).

In the voting stage during the localisation of a query image, in order to account for noise, rectangular regions (Lamdan and Wolfson 1991) centered at the computed invariant coordinates (α, β), were accessed to ensure the 'correct' bin is included and the votes for the correct model are not lost. The voting scheme acts as a sieve reducing significantly the number of candidate hypotheses for the verification stage.

In addition to considering only the top fraction of the sorted candidate list as done in Lifshits et al. (2004), the scheme was extended to further reduce the verification load by employing scale and orientation filters.

Voronoi tessellation (De Berg et al. 2008) and the RANSAC estimator (Fischler and Bolles 1981) were used to accelerate the verification process and to make it robust to outliers as done in Lifshits et al. (2004) and Begelman et al. (2006). For each candidate model, the least-squares-fit transformation relating the query image and candidate model were computed using the result of the RANSAC (Hartley and Zisserman 2003). The model that shared the highest number of inliers with the query image was declared the best matching model and the corresponding least-squares-fit transformation the best estimate of the transformation relating the two.

Unlike in Liftshits et al. (2004) and Begelman et al. (2006), in which localisation tasks were performed using only simulated microscopy query images, the methods presented were tested also on real query ZN-stained sputum smear images. The performance of the localisation stage was measured using the hit rate and discriminative power as done in Lifshits et al. (2004) and the performance of image registration was measured using the mean square error (Cheng 1996, Zitova and Flusser 2003). A comparative method, the SIFT scheme, was also used to compare the image registration performance.

The flow chart in Figure 7.1 summarises the algorithm.

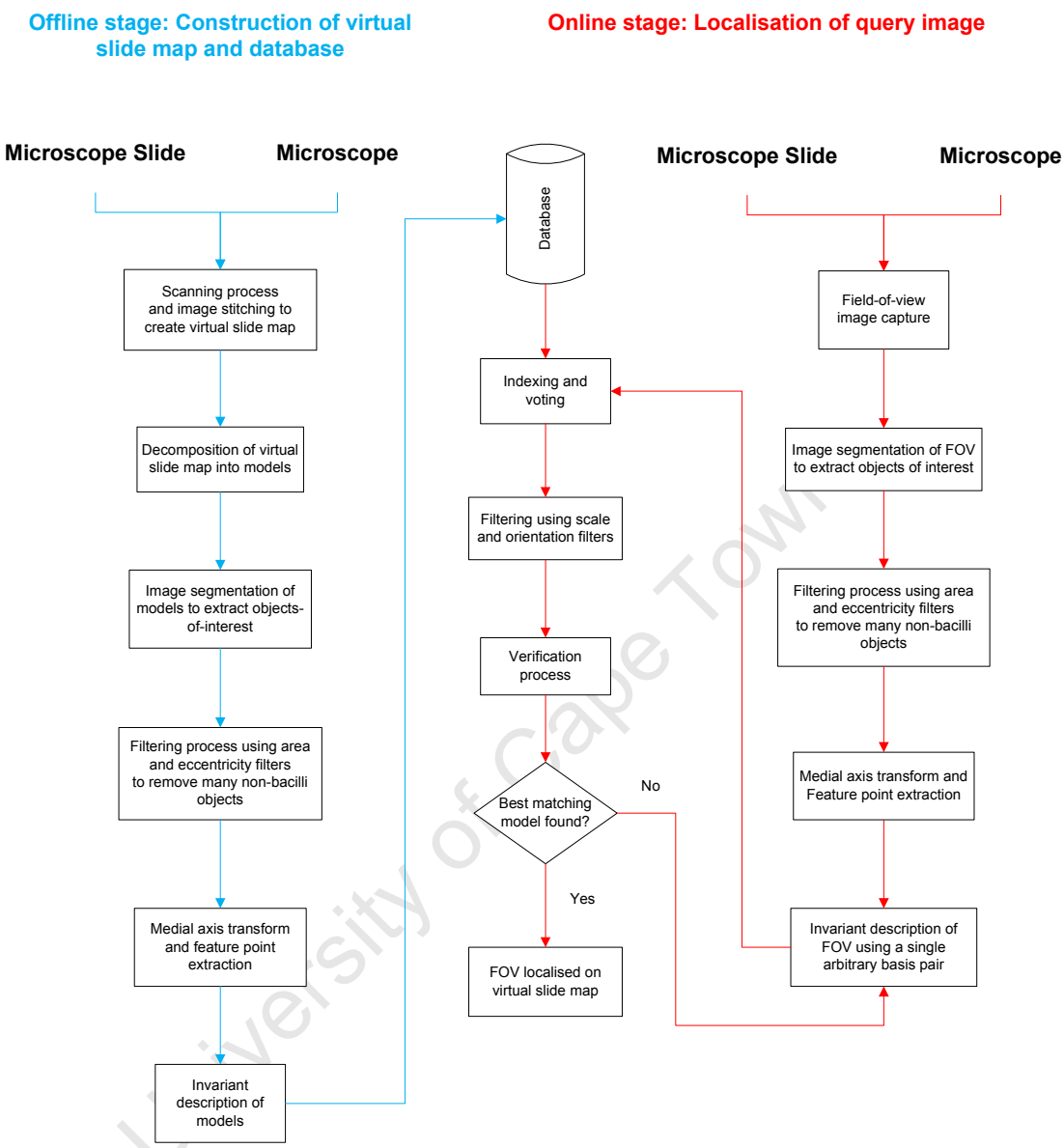


Figure 7.1: Flow chart summarising the algorithm.

8. Results

This chapter presents experimental results obtained using images from the bright field microscope, ZEISS Axioskop 2. An overview of the various tests performed and the performance assessment techniques used is first presented followed by the results obtained for the *offline pre-processing stage* and the *online localisation stage*.

8.1 Testing and performance assessment

The first step in the pre-processing stage involved constructing the virtual slide maps by image stitching. Three virtual slide maps were created of three different slides. The slides were named slide C, Slide O and slide D1. Slide C had relatively fewer bacilli per field. Images acquired from slide C were manually stitched to form its virtual slide map while those of slide O and slide D1 were automatically stitched. Slide O was used to compare the stitching qualities of the two auto-stitching schemes. Therefore, two virtual slide maps of slide O were constructed; one using the geometric hashing scheme (GHS), named GHS VSO, and the other using scale invariant feature transform scheme (SIFT), named SIFT VSO. The comparison was done using various tests as shown in Figure 8.1 and explained in detail in Section 6.2.

For a given slide, once the virtual slide map was constructed, further offline pre-processing was performed, which included decomposition of the virtual slide maps into models, segmentation of the model images, filtering the segmented images, extraction of feature points, model representation and storage in the database.

The overall object recognition algorithm for auto-positioning was tested using slide C and slide D1. Two different sets of images were tested; *Query images Set 1* comprised image extracted directly from the virtual slide maps while *Query images Set 2* comprised sub-sets of real images captured with different slide orientations and at different times. Since *Query images Set 1* was noise-free relative to the models, the set was also used to study the variation of the algorithm's discriminative power with the number of feature points in query images and with quadtree enhancement. *Query images Set 2* was used to

observe the robustness of the algorithm to geometric changes such as rotation and displacement of the slide, illumination changes and noise.

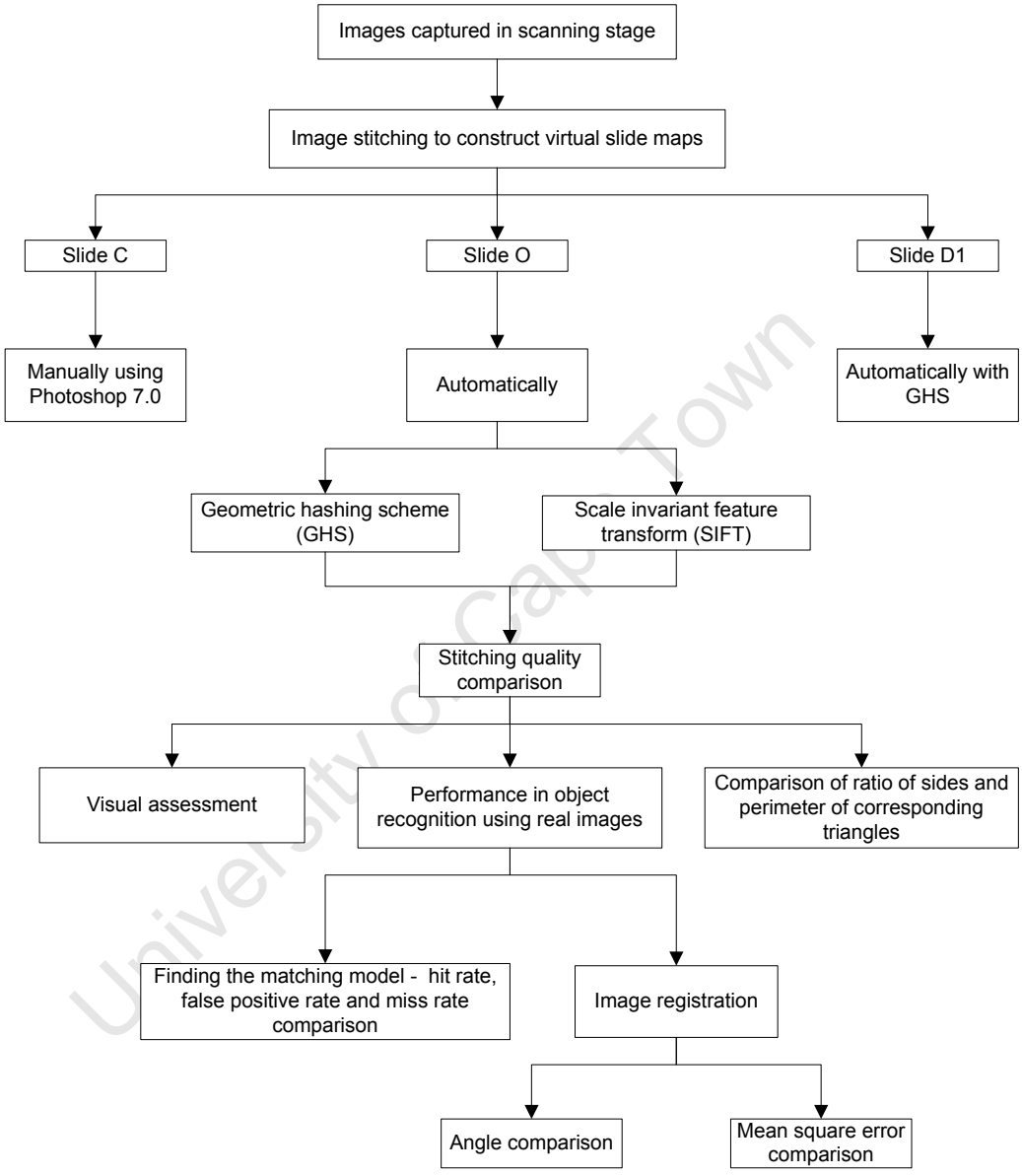


Figure 8.1: Image stitching of images from different slides and testing of stitching quality.

The image registration parameters computed by the algorithm were also compared to those obtained using the SIFT registration method subsequent to finding the matching model. Figure 8.2 summarises the object recognition tests performed and performance assessment techniques used.

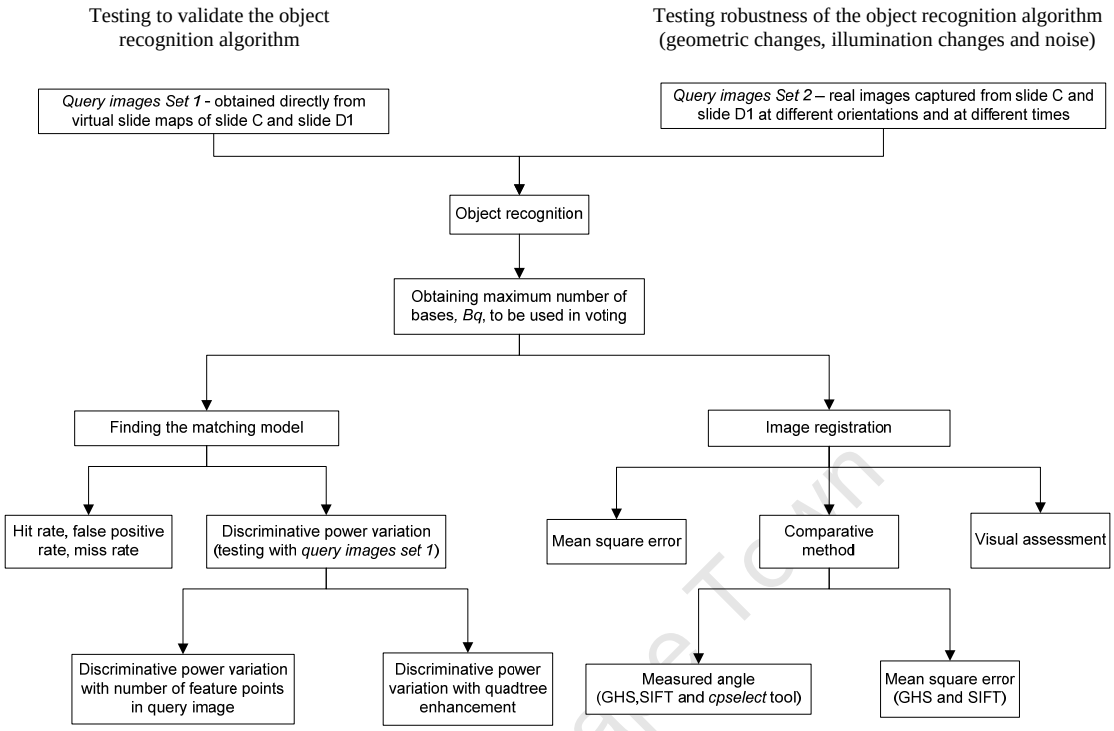


Figure 8.2: Objects recognition tests and performance assessment.

8.2 Results: Offline pre-processing stage

8.2.1 Construction of the virtual slide map

8.2.1.1 Scanning and image acquisition

The scanning and image acquisition was done manually and took 3-5 hours to perform. On average it took Table 8.1 summarises the images acquired.

Table 8.1: Images acquired from different slides.

Slide	Size of rectangular region on slide (mm^2)	Overlap region	Total number of images	Stitching method
C	4 x 2	5 – 15 %	127	Manual
O	1.6 x 1.1	30 – 50 %	62	GHS and SIFT auto-stitch
D1	4.5 x 2	30 – 50 %	307	GHS auto-stitch

Slide O was used to compare the stitching quality of the GHS auto-stitching scheme and the SIFT auto-stitching scheme for stitching microscopy images of ZN-stained sputum

smears. This was done by comparing the resulting composite images, GHS VSO and SIFT VSO. The suitability of the composite images in object recognition for auto-positioning was also tested with numerous real images since the ultimate goal to construct the virtual slide maps was to use them for auto-positioning. This test also provided another basis on which to compare the two auto-stitching methods as shown in Figure 8.1. Since the sole purpose of slide O was to compare stitching quality of the two methods, only a small rectangular region on slide O was covered, which generated 62 images.

Slide C and Slide D1 were used to test the performance of the overall object recognition algorithm for auto-positioning. For both the slides a roughly equal rectangular region was considered. RAM limitations of the computer dictated the size of the virtual slide map that could be constructed and hence also the size of the rectangular region marked on the slides prior to the scanning process. Although the rectangular regions for slide C and slide D1 were roughly equal in area, scanning Slide D1 resulted in a higher number of images due to the relatively larger overlap region used (Table 8.1). This was because images acquired from Slide C were manually stitched while those of slide D1 were automatically stitched and required sufficient overlap (Section 6).

8.2.1.2 Assembling the acquired single images by image stitching

Image Stitching of Slide C

Slide C's virtual slide map (VSC) was constructed by manually assembling the acquired images in Photoshop 7.0 as explained in Section 4.1.2.1. Limitations of the maximum image size of the composite image in Photoshop 7.0 in addition to the RAM limitations of the computer dictated the size of the virtual slide map that could be created and hence also the size of the rectangular region marked on the slide prior to the scanning process.

Image Stitching of Slide O

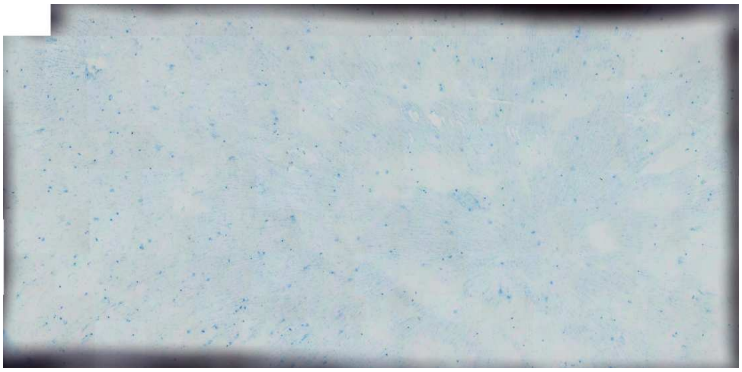
Two virtual slide maps of slide O (VSO) were constructed by automatically stitching the acquired images; one using the SIFT auto-stitching scheme and the other using the GHS auto-stitching scheme. One of the methods of comparing the two auto-stitching schemes

involved performing object recognition tasks for auto-positioning and hence required pre-processing of the VSOs. Therefore, comparison of the two auto-stitching schemes is presented in Section 8.3 after all the pre-processing results are presented.

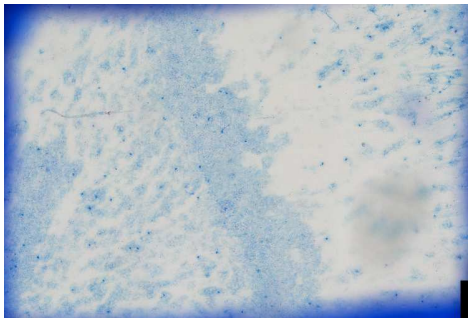
Image Stitching of Slide D1

The virtual slide map of slide D1 was constructed using the GHS auto-stitching scheme which was found to be comparable to the SIFT auto-stitching scheme and suitable for object recognition in the auto-positioning application.

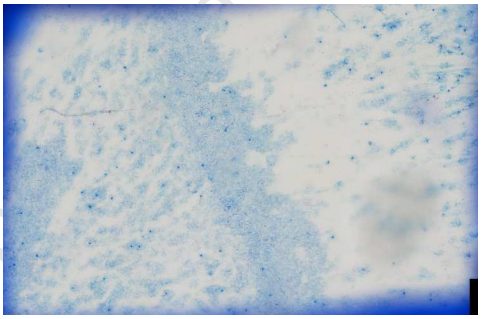
Figure 8.3 shows the virtual slide maps of the various slides. The dark portions at the border of the virtual slide maps are the boundaries of the rectangular regions on the slides which were drawn using a permanent marker. Figure 8.4 shows the difference error image between SIFT VSO and GHS VSO.



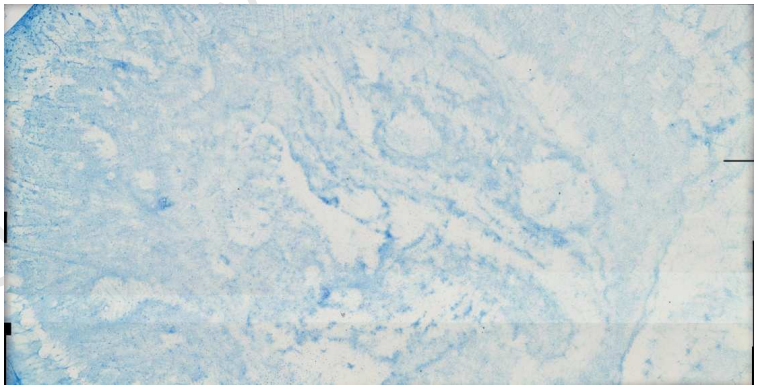
(a) Virtual slide map C - manual image stitching



(b) Virtual slide map O using SIFT auto stitch



(c) Virtual slide map O using GHS auto-stitch



(d) Virtual slide map D1 - GHS auto-stitching scheme

Figure 8.3: Virtual slide maps of the different slides.

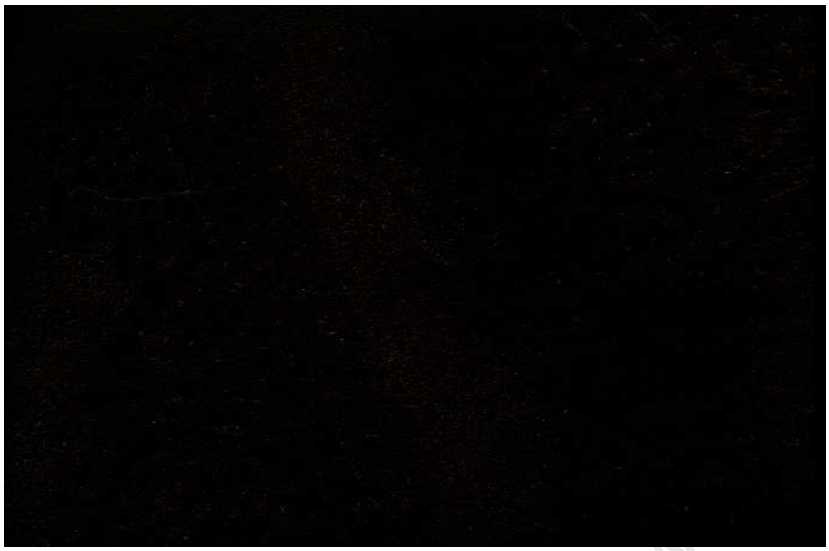


Figure 8.4: Difference error image between SIFT VSO and GHS VSO

8.2.2 Decomposition of virtual slide map to generate models

To simplify and permit the object recognition process under RAM constraints, a virtual slide map was broken into smaller portions referred to as models as explained in Section 4.2. Each model was invariantly represented and stored in the database. For a given slide, during object recognition, the database corresponding to that slide (Section 8.2.5) is needed to be in the computer RAM. Therefore, the Computer RAM size dictated the maximum size of the database and hence the number of models that can be stored in it. By breaking down the virtual slide map in the manner as explained in Section 4.2, the number of models produced by each slide is directly related to the size of the virtual slide map and hence to the rectangular region covered on the actual slide. Hence the size of the rectangular region was dictated by the computer RAM size. The number of models produced for each of the slides is shown in Table 8.2.

Table 8.2: Number of models produced for each virtual slide map.

Slide	Size of rectangular region	Number of models produced
C	4 x 2	77
O SIFT	1.6 x 1.1	12
O GHS	1.6 x 1.1	12
D1	4.5 x 2	84

8.2.3 Image segmentation

All the model images were segmented using the quadratic pixel classifier. Most segmented images contained numerous non-bacillus objects as seen in Figure 8.5.

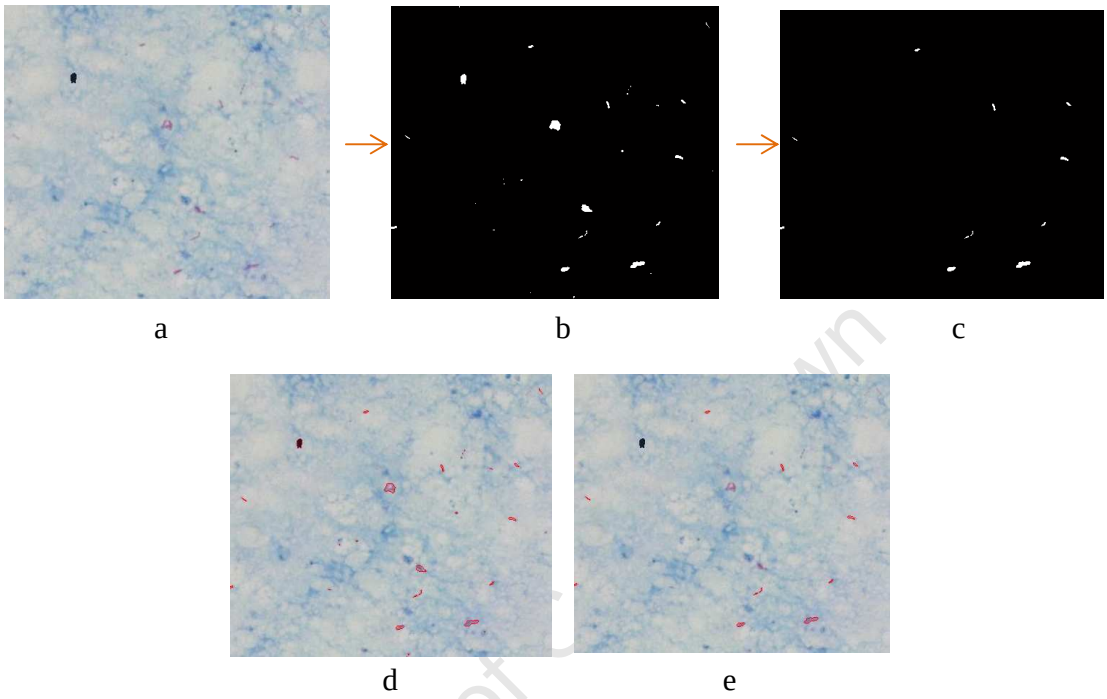


Figure 8.5: Image segmentation followed by filtering; (a) an example original image (b) segmented image of (a). (c) filtered segmented image of (a). (d) outlines of the segmented objects before filtering superimposed on (a). (e) outlines of the segmented objects after filtering superimposed on (a).

Many of the non-bacillus objects were removed using area and eccentricity filters. The threshold values were empirically determined and are shown in the Table 8.3.

Table 8.3: Area and eccentricity thresholds.

Segmented Model images	Area thresholds	Eccentricity
Slide C models	$5 < A < 500$	$e > 0.85$
Slide O SIFT models	$10 < A < 500$	$e > 0.85$
Slide O GHS models	$10 < A < 500$	$e > 0.85$
Slide D1 models	$10 < A < 500$	$e > 0.85$

A and *e* are the area and eccentricity respectively of an object in the segmented image. A lower area threshold was used for slide C model images because slide C contained very few bacilli per image and therefore required several non-bacillus objects to generate sufficient feature points to produce good voting results (refer to Sections 8.4.1.2 and 8.4.1.3 to understand why).

Table 8.4 shows the average number of objects in the segmented image and in the filtered segmented image. As seen, there is a huge drop in number of objects after filtering. This suggests that the image segmentation produced several non-bacillus objects.

Table 8.4: Average number of objects before and after filtering the segmented model image.

Models	# of objects in segmented image	# of objects in filtered segmented image
Slide C	217	85
Slide O SIFT	592	116
Slide O GHS	588	116
Slide D1	338	127

Since at least one feature point is extracted from each object (Section 8.2.4), the estimated number of feature points extracted from an image should at least equal the number of objects in the image. The number of entries into the database for a model containing *n* feature points is *n(n-1)(n-2)* as shown in Section 4.6.4. Table 8.5 shows how many entries would be saved in the database for images before and after the filtering process.

Table 8.5: Approximate number of entries into the hash table per model before and after filtering.

Models	Segmented image		Filtered segmented image		Difference in # of entries
	Min. # of feature points	# of entries into database	Min. # of feature points	# of entries into database	
Slide C	217	10077480	85	592620	9484860
Slide O SIFT	592	206424480	116	1520760	204903720
Slide O GHS	588	202261416	116	1520760	200740656
Slide D1	338	38272416	127	2000250	36272166

As seen in the last column of Table 8.5, there is a large difference in the number of entries into the database per segmented model image before and after filtering. Each entry is of the form (M_i, m_μ, m_v) as explained in Section 4.5. Furthermore each virtual slide map consists of numerous models. Consequently, the filtering process is a critical step which substantially reduces the number of entries into the database bins and helps in speeding up the voting stage (Section 5.2) and hence the object recognition process. Additionally, fewer entries means smaller size of the database and therefore the filtering process also allows more models to be stored in the database and hence allows a larger rectangular region of the slide to be covered even under tight computer RAM constraints.

8.2.4 Extraction of feature points

The feature points were extracted subsequent to applying the medial axis transform (Section 2.4.2) to the filtered segmented images. The feature points included two kinds of points:

- For objects with branched skeletons, the branch points were extracted as feature points. Some object skeletons were found to have multiple branch points.

- For objects with branchless skeletons, the mid point of the skeleton was extracted as a feature point. These object skeletons hence only generated a single feature point.

Figure 8.6 shows feature point extraction on a zoomed sub-image. The red dots represent the extracted feature points. As seen, one of the objects has a branched skeleton with 3 branch points while the other two are branchless and hence only generate one feature point each. Another example was shown in Figure 4.2.

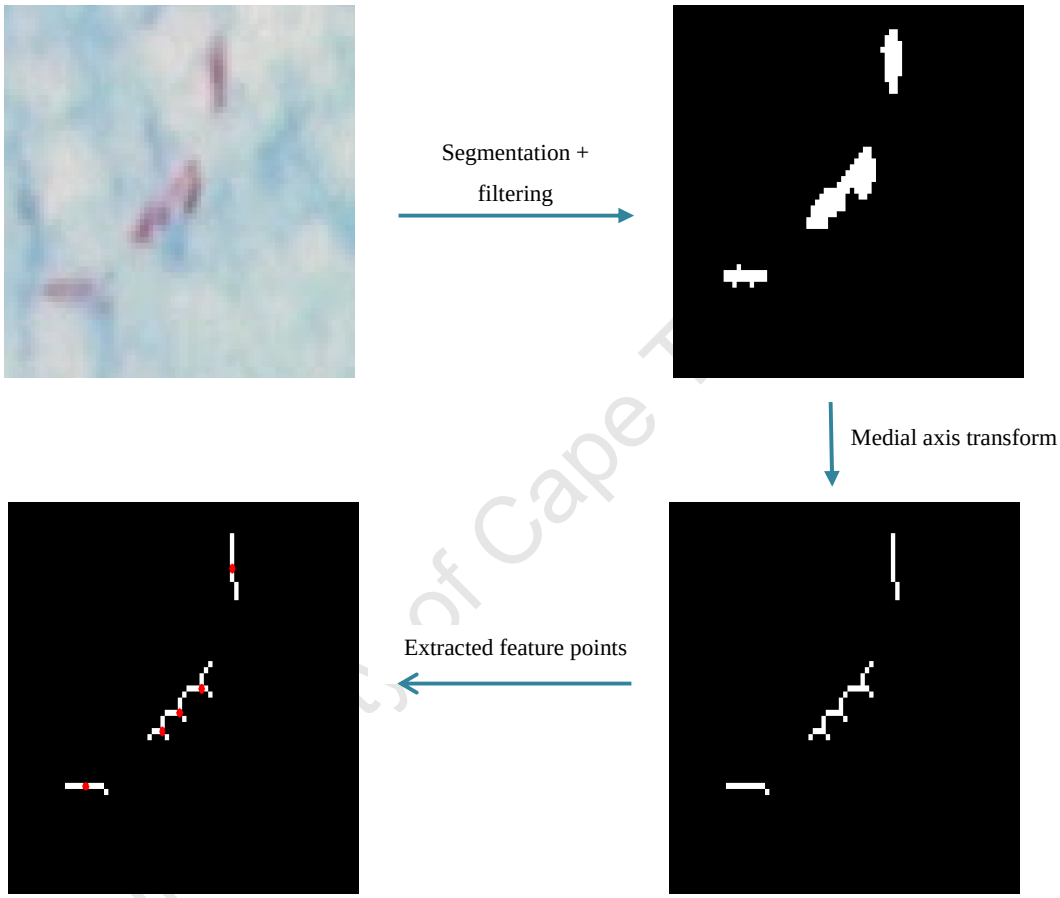


Figure 8.6: Feature point extraction.

Therefore, each object generated at least one feature point, and the number of feature points in an image was at least equal to the number of objects in that image. Table 8.6 shows the average number of feature points that were extracted per model image.

Table 8.6: Average number of feature points per model image.

Model image (2060 x 2600 pixels)	Average number of features points per model image
Slide C	90
Slide O model - SIFT	141
Slide O model - GHS	138
Slide D1	140

The difference in the number of feature points between model images of slide C and those of the other two slides was mainly because Slide C had relatively fewer bacilli per field.

8.2.5 Model representation and database construction

Representation of a model and hash table filling took on average 2 minutes. The time taken is of little significance since it is the offline stage. Table 8.7 summarises the properties of the database.

Table 8.7: Properties of the database of the various slides.

Database	Properties of database	
	Total # of entries	Size in RAM (MB)
Slide C database	104,326,080	457
Slide O GHS database	42,354,828	255
Slide O GHS database	42,510,480	256
Slide D1 database	242,387,172	890

As expected, the database of slide O was smallest since it comprised only 12 models. The size of slide D1's database is almost double that of slide C's database although the difference between the number of models in slide C and slide D1 is small (Table 8.2). This is because a slide D1 model comprised on average 50 feature points more than a slide C model (Table 8.6) which implies that per model there were 50(50-1)(50-2) = 117, 600 entries more in slide D1's database than in slide C's database. Therefore, the

database size is not only dependent on the number of models stored in it but also on the number of feature points present in those models.

8.3 Comparison of GHS auto-stitching scheme with SIFT auto-stitching scheme

The virtual slide map of slide O (VSO) shown in Figure 8.3 was constructed from 62 images - each measuring 1030 x 1300 pixels - using the GHS and SIFT auto-stitching methods. Several methods were adopted to compare the stitching quality of the two methods.

8.3.1 Visual comparison

The resulting SIFT VSO measured 4049 x 5976 pixels while the GHS VSO measured 4044 x 5979 pixels. The almost identical sizes indicate very little discrepancies exist between the two VSOs. In addition, the difference error image in Figure 8.4 shows there are little differences between the two VSOs. There were no detectable differences at and near the visible seam lines upon visual comparison of the two VSOs.

8.3.2 Quantitative test using the triangle method

A set of 1140 corresponding triangles in the SIFT VSO and the GHS VSO were used for this analysis. The triangles varied in size and covered various regions in the two VSOs. Table 8.8 shows the results obtained for 5 randomly chosen pairs of corresponding triangles.

Table 8.8: Quantitative test results of comparing the SIFT auto-stitch and GHS auto-stitch schemes.

Triangle	Lengths of sides of triangle		Ratio of sides in triangle	
	In SIFT VSO	In GHS VSO	In SIFT VSO	In GHS VSO
1	1858.06 : 5200.01 : 3343.50	1857.31 : 5201.11 : 3345.47	1 : 2.799 : 1.799	1 : 2.800 : 1.801
2	2200.48 : 4661.81 : 2624.95	2200.48 : 4662.54 : 2625.82	1 : 2.119 : 1.193	1 : 2.119 : 1.193
3	1247.68 : 5488.41 : 4730.46	1246.92 : 5491.53 : 4736.49	1 : 4.399 : 3.791	1 : 4.404 : 3.798
4	5488.41 : 5749.75 : 823.19	5491.53 : 5754.16 : 823.92	6.667 : 6.985 : 1	6.665 : 6.984 : 1
5	5641.28 : 3838.30 : 2751.95	5645.66 : 3841.02 : 2755.11	2.050 : 1.395 : 1	2.049 : 1.394 : 1

Figure 8.7 is a bar chart illustration of the ratios of the sides of the 5 triangles. For each triangle, the purple bars represent the ratio of the sides of the triangle in the SIFT VSO while the green bars represent the ratio of the sides of the triangle in the GHS VSO.

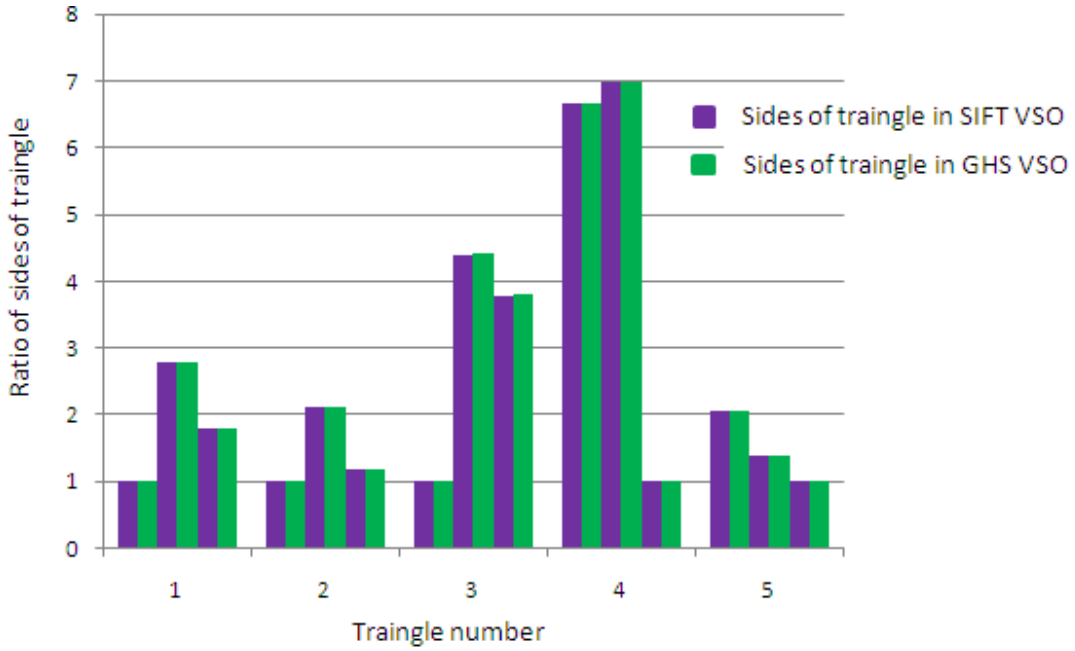


Figure 8.7: Bar chart relating the ratios of the sides of the triangles in the SIFT VSO and GHS VSO

If the stitching of images by the two methods is same then the corresponding triangles in the two VSOs are expected to be congruent – have the same size and shape and hence also have the same perimeter. As seen in Table 8.8 the discrepancies between the lengths and ratios of the triangles from the two VSOs are extremely small. The same was observed for all the other triangles. The average difference between the perimeters of corresponding triangles was 4.08 pixels with a standard deviation of 3.52 pixels, which is visually undetectable. Since these differences are extremely small, the two VSOs are similar and hence the stitching quality of the two auto-stitching schemes is comparable.

8.3.3 Suitability of auto-stitching in object recognition for auto-positioning

To validate the capability of the two auto-stitching schemes in constructing virtual slide maps that are suitable in object recognition for auto-positioning, object recognition tasks - online localisation stage - were performed using real query images of slide O and the two VSOs derived using GHS and SIFT. This test also served a secondary purpose; to

compare the stitching quality of the two auto-stitching methods by comparing the performance of the two VSOs in object recognition.

This task required pre-processing of the VSOs. All the pre-processing results for the SIFT VSO and GHS VSO were almost identical as observed in Table 8.2 to Table 8.7.

Table 8.9 and Table 8.10 show the performance of the two VSOs in object recognition for auto-positioning.

Table 8.9: Object recognition performance using VSO constructed with SIFT and GHS.

Auto-stitching scheme	Number of Images tested	Misses		False Positives		Hits	
		#	Miss rate	#	FP rate	#	Hit rate
SIFT VSO	40	2	0.05	0	0.00	38	0.95
GHS VSO	40	3	0.08	0	0.00	37	0.93

Table 8.10: Image registration performance using VSO constructed with SIFT and GHS.

Auto-stitching scheme	Slide O orientation (°)	angle reported (°)		MS error (pixels^2)		RMS error (pixels)	
		Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.
SIFT VSO	0	0.02	0.08	10.63	3.45	3.21	0.57
GHS VSO	0	0.01	0.06	10.79	2.87	3.25	0.45

For both VSOs, a hit rate of above 90% was achieved with an average mean square error (image registration error) of less than 11 pixel^2 corresponding to a root mean square registration error of 3.3 pixels. Considering that a given FOV image measures 1030 x 1300 = 1.34 x 10^6 pixels^2, a mean square error of 11 pixels^2 translates to 11 / 1.34 x 10^6 = 0.001% error. A pixel measures 0.27µm x 0.27µm (Section 3.1) and therefore, 11 pixels^2 is equivalent to 0.80 µm^2.

Therefore, the GHS and SIFT auto-stitching schemes are both suitable for constructing virtual slide maps for the auto-positioning application.

Both auto-stitching schemes missed the same two images. The GHS scheme missed an additional one which comprised 16 feature points. For both the VSOs, the average mean square error reported by the object recognition algorithm is almost identical. These results further show that the GHS auto-stitching scheme and the SIFT auto-stitching scheme produce similar stitching quality.

Conclusion - In all the comparison tests, the SIFT auto-stitching scheme and the GHS auto-stitching scheme are comparable and both the schemes are suitable for constructing virtual slide maps for the auto-positioning application.

The GHS auto-stitching scheme was selected to auto-stitch a larger set of images (307 images) to form the virtual slide map of slide D1 as shown in Section 8.2.1.2. The GHS auto-stitching scheme was selected over the SIFT auto-stitching scheme to maintain consistency since the object recognition algorithm for auto-positioning was based on the GHS scheme.

8.4 Results: Online localisation stage performance using slide C and slide D1

The major component of auto-positioning is object recognition which involves finding the matching model to the query image – localisation - and finding the transformation relating the two images - image registration. This section presents the results of the object recognition localisation and image registration tests. The object recognition algorithm developed is based on the GHS scheme. The algorithm produces the matching model and the registration parameters simultaneously (Section 5.3).

A set of query images was formed by obtaining images at a particular orientation, say θ_c, of the slide. Therefore, the registration angle parameter for any image in that set is expected to be θ_c. Since all the images were captured at 40x magnification, the registration scale parameter is expected to be 1. Only the translation parameters will differ from image to image in a given set of query images. This is because the models

are four times larger in size than any query image and hence the position of the query images in a model can vary.

For a given query image and its matching model, the registration transformation error can be computed as the mean square error. Therefore, registration parameters were evaluated by visual assessment (Section 8.4.3) and the average of the mean square error.

Additionally, the registration parameters obtained using the algorithm were compared to those obtained by the SIFT matching scheme, which was used as the comparative method (Section 5.5.2.3). For the evaluation of the registration parameters, only the true positives were considered.

To determine whether or not the output model was the correct model or not, the colour images of the query image and the reported matching model image were visually compared (Section 5.5.1.1). Figure 8.8 shows a few example 1030 x 1300 images and the features/patterns (enclosed in a purple outline) that helped simplify visual comparison of the query image and the matching model.

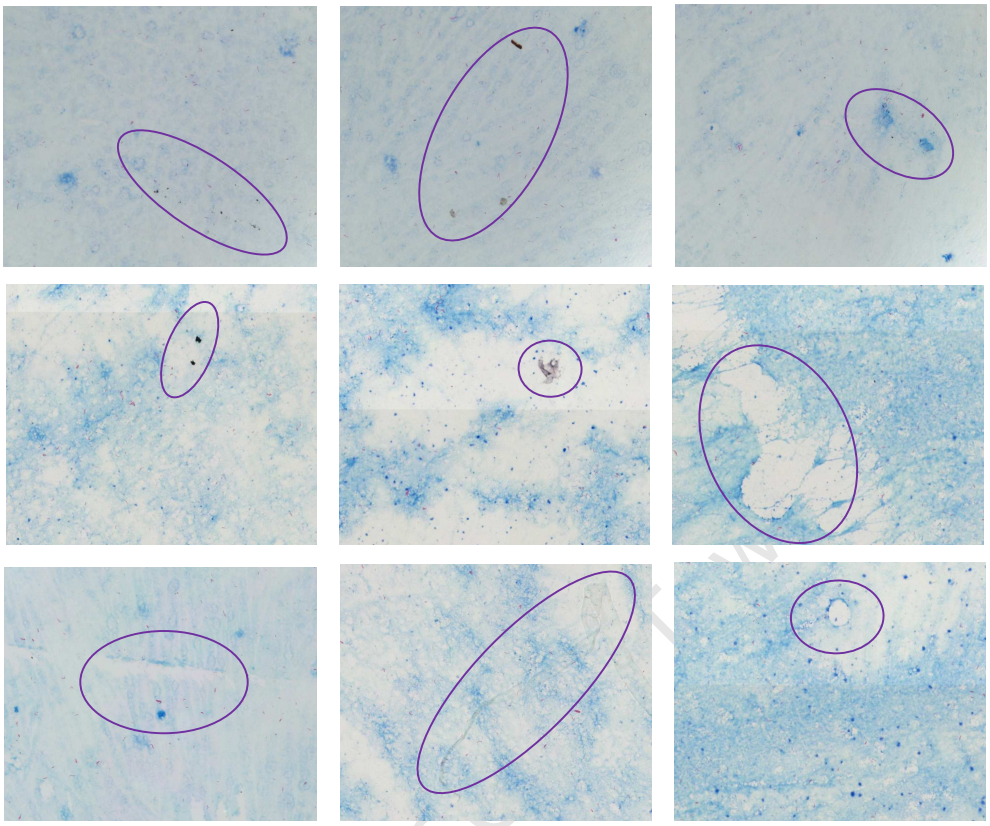


Figure 8.8: Features enhancing visual comparison of query image and best matching model image.

8.4.1 Object recognition of *Query images Set 1*

These query images were obtained directly from the virtual slide map. For each slide, 250 images were extracted. Since the query images were obtained directly from the virtual slide map, the orientation is theoretically exactly 0° and the scale factor is 1. The algorithm performance on this set of images is presented in Table 8.11 to Table 8.13. The maximum number of bases used in voting was set to 10 (Section 8.4.1.1).

Table 8.11: Object recognition performance of geometric hashing with *Query images Set 1*.

Slide	Number of Images tested	Misses		False matches		Hits	
		Number	Miss rate	Number	FP rate	Number	Hit rate
C	250	5	0.02	16	0.06	229	0.92
D1	250	3	0.01	2	0.01	245	0.98

Table 8.12: Comparison of the average angle reported by algorithm.

Slide	Slide Orientation (°)	Angle reported (°) - GHS		Angle reported (°) SIFT	
		Avg.	Std Dev.	Avg.	Std Dev.
C	0	0.001	0.060	0.000	0.003
D1	0	0.003	0.053	0.000	0.003

Table 8.13: Comparison of registration parameters using the average mean square error.

Slide	Registration errors - GHS				Registration errors - SIFT			
	MS error (pixels^2)		RMS error (pixels)		MS error (pixels^2)		RMS error (pixels)	
	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.
C	5.99	2.98	2.35	0.68	0.07	0.32	0.19	0.17
D1	7.84	3.03	2.75	0.55	0.09	0.45	0.22	0.21

In all cases, the scale parameter of the least-squares-fit transformation relating the query image to the matching model (Section 5.3), on average was 1.000 with standard deviation < 0.001.

8.4.1.1 Maximum number of bases used in voting

During the voting stage, an arbitrary basis pair, B_q, is first selected from the query image. The algorithm may or may not find the matching model to the query image using this basis pair, B_q. This is mainly because of noise as explained in Section 5.3. Consequently, a second arbitrary basis pair needs to be selected and the object recognition process needs to be repeated. To prevent the algorithm from endlessly running (in the case when the algorithm keeps failing to find a matching model to the query image), the maximum allowable number of bases, B_q, to be used in voting was set. An attempt to match a query image with an arbitrary basis can be referred to as a basis attempt. Therefore, for example if the maximum basis attempts was set to 5, and the algorithm failed to find the matching model using 5 arbitrary bases (i.e. 5 basis attempts), then it was declared a miss.

Figure 8.9 shows how the object recognition performance varies depending on the number of maximum allowable basis attempts per query image. All 500 images of *Query image Set 1* were considered for this analysis.

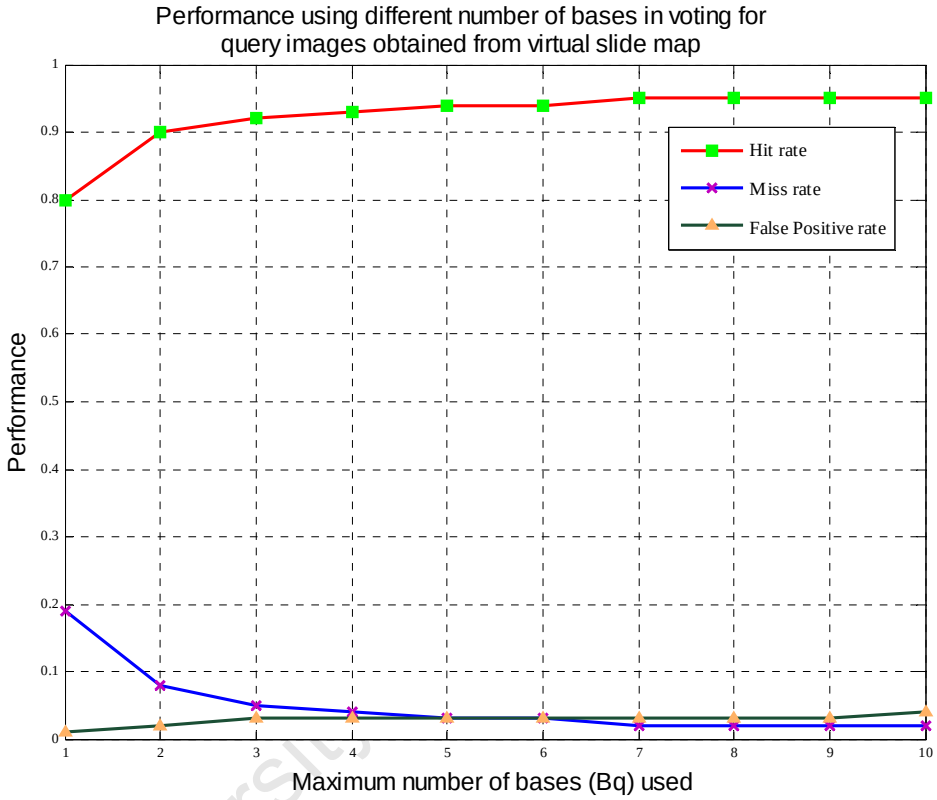


Figure 8.9: Object recognition performance with varying number of maximum attempts.

Since the images in *Query images Set 1* are completely noise-free relative to the matching model, a query image from this set is theoretically expected to be matched with the first arbitrary basis i.e. first basis attempt. However, as seen the hit rate using 1 arbitrary basis only was 80 percent and it improved as more basis attempts were allowed per query image. This observation can be explained by poor discriminative power of the algorithm for the missed query images. For a given query image, the selected basis may produce the correct model which does not lie near the top of the sorted candidate list *(CL1)* and hence is eliminated during the verification load reduction process (Section 5.2.1.1).

For the same query image, a different arbitrary basis, B_q, may produce the correct model which lies close enough to the top of the sorted candidate list *(CL1)* to not be eliminated and hence would be successfully matched. Therefore, multiple attempts improve the hit rate as shown in Figure 8.9. The miss rate did not fall to 0 even with 10 basis attempts. This was because several query images had very few features points, lowering the discriminative power as shown in Figure 8.10. For these images the correct model is highly likely to be eliminated before the verification stage, even with many different bases considered (i.e. even with many basis attempts).

8.4.1.2 Reduction of the verification load

The effectiveness of the methods used to reduce the verification load varied greatly and no pattern could be derived. However, to signify their effectiveness, localisation of some random query images of *Query images Set 1* was closely analyzed. Table 8.14 highlights the effectiveness of the methods employed to reduce the verification load.

Table 8.14: Effectiveness of the methods in verification load reduction.

Query image number	**# of feature points present**	**Hash table entries that received at least 2 votes**	**# of *CMBs* that received at least 40% of max. vote**	**# of *CMBs* after scale and angle filters (i.e. *CMBs* carried forward to the verification step)**
35	34	715064	32	1
67	42	1012239	74	2
74	42	1151440	100	4
76	31	748043	57	2
91	22	694188	483	5
132	14	196270	288	2
143	38	991533	134	6
219	15	488689	806	2
225	27	498816	109	2
247	31	861050	256	5

Only the *CMBs* remaining after the filtering process were carried forward to the verification stage. As seen in Table 8.14, the possible matches to the query image, *Q*, and the selected basis pair, B_q, in *Q* was reduced from hundreds of thousands to ≤ 6. For all the query images in all the tests (including *Query images Set 1* and *Query images Set 2*), less than 10 *CMBs* were carried forward to the verification stage.

The algorithm would be able to find the correct matching *CMB* only if that *CMB* received a significant number of votes and lied near the top of the sorted candidate list, *CL1* and therefore was not eliminated in the verification load reduction process. This motivated the analysis of the discriminative power of the algorithm.

8.4.1.3 Discriminative power variation

The discriminative power of the algorithm (Section 5.5.1.2) was found to be affected by the number of feature points present in the query image as shown in Figure 8.10. It can be seen the algorithm is highly discriminative (the best matching *CMB* is near the top of the sorted candidate list (*CL1*)) when the query image contains about 23 feature points. An additional feature point further improves the discriminative power since each additional feature point provides supplemental evidence for the presence of the correct model. The algorithm's discriminative power can be improved by quadtree enhancement (Section 5.5.1.2) as seen in Figure 8.10. This was expected since only an appropriate section of the virtual slide map is participating in the voting stage (i.e. many irrelevant models do not participate in the voting stage). The improved discriminative power would in turn result in improved hit rate.

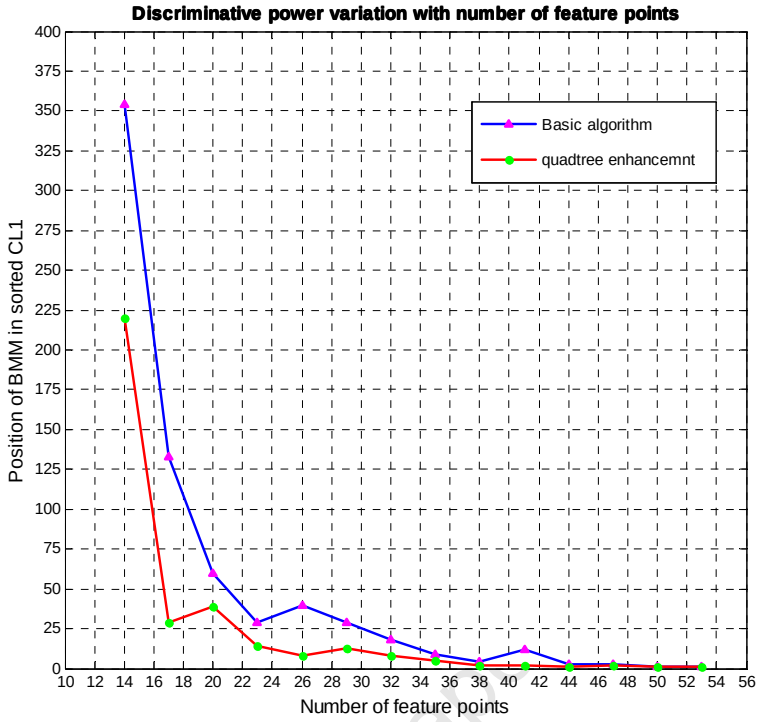


Figure 8.10: Discriminative power variation with number of feature points and quadtree enhancement.

8.4.2 Object recognition of *Query images Set 2*

Query images Set 2 comprised sub-sets of real images where each of the sub-sets was captured at a different time and orientation as explained in Section 5.5.1.1. Since all images were captured at 40x magnification, the scale factor between a query image and the matching model is expected to be 1. Different orientation angles were considered for the two slides to cover a range from 0 – 26 °.

8.4.2.1 Obtaining a better estimate of the orientation of the slide

For a given sub-set of images in *Query images Set 2*, the orientation of all the images is expected to be the same. This orientation could only be roughly measured using a protractor during the scanning phase due to mechanical blockages and the limited resolution of the protractor (Section 5.5.1.1).

For a given sub-set of images, a more accurate measure of the orientation of the slide was obtained with the help of the *cpselect* tool in MATLAB. This tool allows two images of different sizes to be navigated simultaneously at desired zoom levels and it facilitates the manual selection of point-to-point correspondences between the two images, which can then be used to compute the similarity transformation relating the two images. From the sub-set, 5 images and their respective matching models were selected. The 5 images were chosen so that they had distinctive features which simplified the process of finding their matching models manually (by visual comparison). Using the c*pselect* tool, 10-15 corresponding points between a query image and its matching model were manually selected. Two corresponding pairs of points are sufficient to compute a similarity transformation (Section 2.5.3.2). However, 10-15 corresponding pairs were selected to accommodate errors in matching points as at high magnification it is difficult for the human eye to differentiate between neighbouring pixels. The least-squares-fit similarity transformation parameters were then computed using these points.

The computed translation parameters of the 5 images would differ greatly. This is because a model is 4 times as large as a query image and hence the query image can lie anywhere within its matching model. However, for a given orientation, the computed scale and angle could be expected to be the same for all five images since all images were taken at 40x and all 5 images were taken at the same slide orientation. Table 8.15 shows the roughly measured angles along with those computed with the *cpselect* tool.

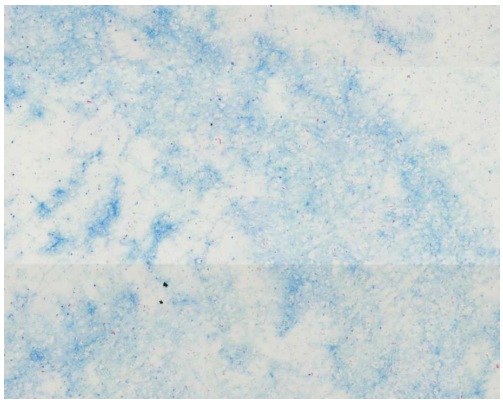
Table 8.15: Orientation estimation using *cpselect* tool in MATLAB.

Orientation Measured roughly with protractor (°)	Slide	Angle measured using *cpselect* tool (°)
0	D1	0.01
2	C	2.44
6	D1	6.43
8	C	9.59
10	D1	10.58
15	C	14.58
16	D1	15.91
20	C	20.00
24	C	23.72
25	D1	24.99
26	C	26.35

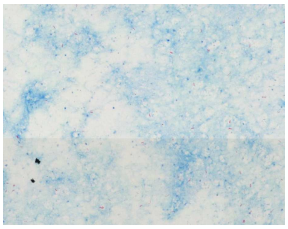
The sub-set of images of a given orientation was named based on the slide they were obtained from and the orientation of the slide computed with the *cpselect* tool. For example, Slide D1 - 10.58 represented the sub-set of images obtained using slide D1 and at orientation 10.58°.

8.4.2.2 Query image variations relative to model images

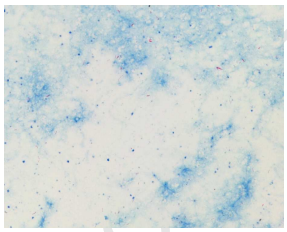
For the different sets of images, in addition to the different orientations, there were considerable image variations among the sets and also between the images and the matching model as can be seen visually in Figure 8.11 and Figure 8.12. Figure 8.11 shows one image from every set of images considered for slide D1 and the matching model in slide D1. Similarly, Figure 8.12 shows one image from every set of images considered for slide C and the matching model of slide C. An image from *Query image Set 1* is also included for each of the slides for comparison.



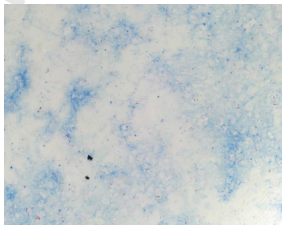
Matching model in virtual slide map of Slide D1



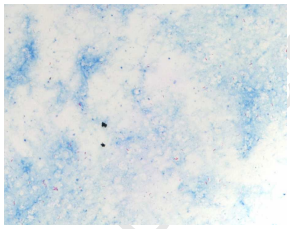
Slide D1 - direct from virtual slide map (*Query images Set 1*)



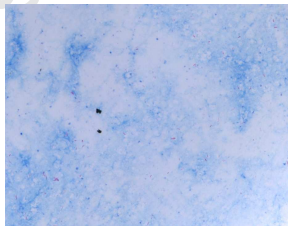
Slide D1 - 0.01



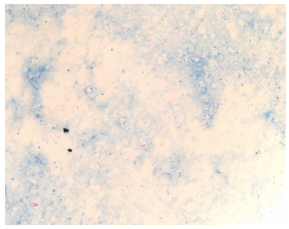
Slide D1 - 6.43



Slide D1 - 10.58

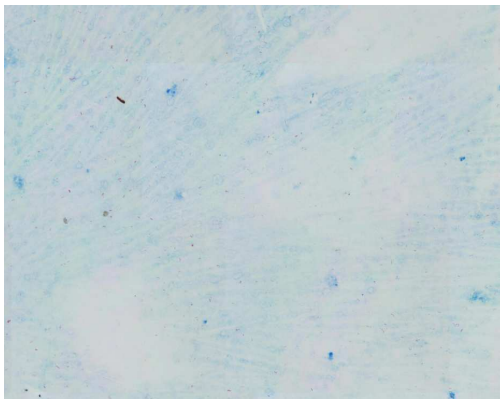


Slide D1 - 15.91

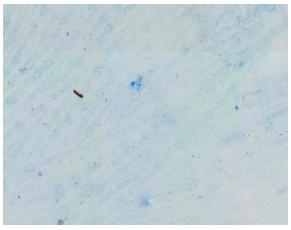


Slide D1 - 24.99

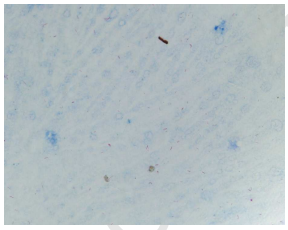
Figure 8.11: Image variations among the different sets of images of Slide D1.



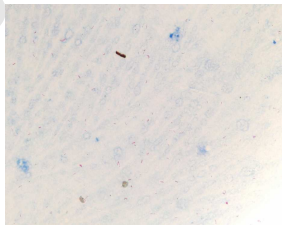
Matching model in virtual slide map of Slide C



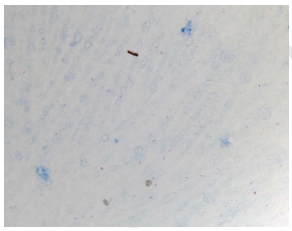
Slide C - direct from virtual slide map (*Query images Set 1*)



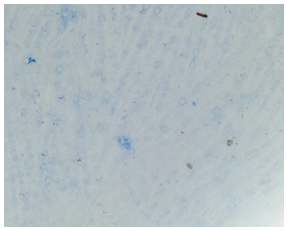
Slide C - 2.44



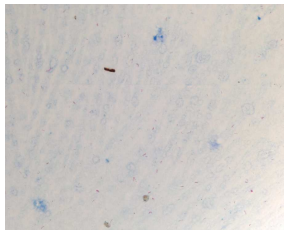
Slide C - 9.59



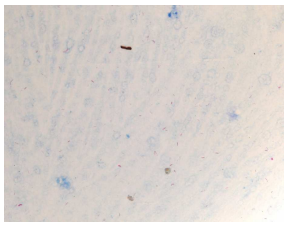
Slide C - 14.57



Slide C - 20.00



Slide C - 23.72



Slide C - 26.35

Figure 8.12: Image variations among the different sets of images of Slide C.

8.4.2.3 Setting the maximum number of bases to be used in voting

In order to determine how many basis attempts in voting should be allowed, analysis as done in Section 8.4.1.1 was performed using *Query images Set 2*. Over 700 real images of slide D1 were considered for this analysis. Since noise is the main cause of failure to match a query image, and since the real query images are noisy relative to the models, the range from 0 - 20 was considered for the maximum allowable basis attempts per query image. Figure 8.13 shows how the object recognition algorithm performance varied with different numbers of maximum basis attempts used in voting per query image.

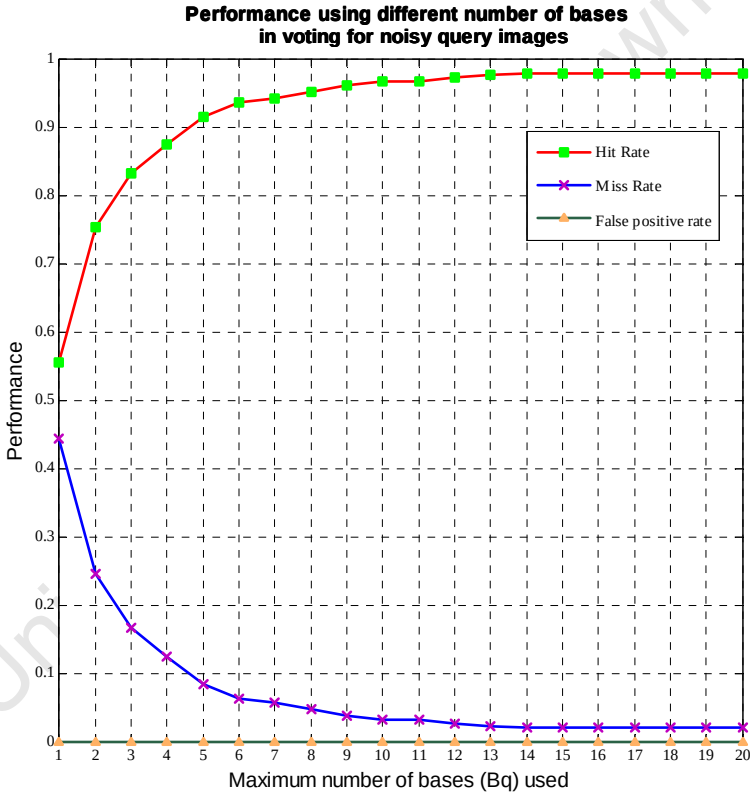


Figure 8.13: Object recognition performance with varying number of maximum attempts using real noisy images.

With few basis attempts on a query image, the algorithm is likely to produce miss because in the presence of noise, the chances of selecting a basis B_q in Q that have a good match in the database are low. Therefore several attempts need to be made to obtain a B_q with a good match and hence the performance improves as the number of

basis attempts is increased as shown in Figure 8.13. Another reason for the improved performance with increase in basis attempts is explained in Section 8.4.1.1.

The hit rate with 10 basis attempts was found to be 97% while that with 20 basis attempts was found to be 98%. Since there is no significant increase in the hit rate after 10 basis attempts, the maximum number of basis used in voting was set to 10. Therefore, all the results relating to *Query images Set 2* were obtained by conducting experiments using a maximum of 10 basis attempts. This meant that if a query image was not matched in 10 basis attempts, it was declared a miss.

8.4.2.4 Object recognition performance with *Query images Set 2*

The object recognition performance on *Query images Set 2* is presented in Table 8.16 to Table 8.19. In all cases, the scale parameter of the least-squares-fit transformation relating the query image to the matching model (Section 5.3), on average was 1.000 with standard deviation < 0.002.

Table 8.16: Object recognition performance with *Query images Set 2*.

Image sub-set	Number of images tested	Misses		False Matches		Hits	
		Number	Miss rate	Number	FP rate	Number	Hit rate
Slide D1 - 0.01	101	2	0.02	0	0.00	99	0.98
Slide C - 2.44	154	7	0.05	1	0.01	146	0.95
Slide D1 - 6.43	152	8	0.05	0	0.00	144	0.95
Slide C - 9.59	152	8	0.05	1	0.01	143	0.94
Slide D1 - 10.58	152	3	0.02	0	0.00	149	0.98
Slide C - 14.58	153	7	0.05	1	0.01	145	0.95
Slide D1 - 15.91	151	2	0.01	0	0.00	149	0.99
Slide C - 20.00	153	19	0.12	0	0.00	134	0.88
Slide C - 23.72	153	13	0.08	0	0.00	140	0.92
Slide D1 - 24.99	154	4	0.03	0	0.00	150	0.97
Slide C - 26.35	152	18	0.12	0	0.00	134	0.88

Table 8.17: Comparison of the average angle reported by algorithm on *Query images Set 2*.

Image sub-set	Reported average angle and standard deviation from average (considering ALL true positives)			
	GHS		SIFT	
	Avg.	Std Dev.	Avg.	Std Dev.
Slide D1 - 0.01	0.02	0.060	0.01	0.021
Slide C - 2.44	2.49	0.075	2.48	0.048
Slide D1 - 6.43	6.44	0.068	6.44	0.032
Slide C - 9.59	9.54	0.073	9.55	0.057
Slide D1 - 10.58	10.58	0.064	10.58	0.027
Slide C - 14.58	14.57	0.078	14.57	0.052
Slide D1 - 15.91	15.91	0.072	15.91	0.035
Slide C - 20.00	19.96	0.071	19.97	0.062
Slide C - 23.72	23.74	0.083	23.74	0.052
Slide D1 - 24.99	25.02	0.070	25.02	0.031
Slide C - 26.35	26.38	0.081	26.38	0.059

Table 8.18: Comparison of the percentage error

Image sub-set	Percentage error relative to the angle measure using Cpselect tool	
	GHS (%)	SIFT (%)
Slide D1 - 0.01	100.00	0.00
Slide C - 2.44	2.05	1.61
Slide D1 - 6.43	0.16	0.16
Slide C - 9.59	0.52	0.42
Slide D1 - 10.58	0.00	0.00
Slide C - 14.58	0.07	0.07
Slide D1 - 15.91	0.00	0.00
Slide C - 20.00	0.20	0.15
Slide C - 23.72	0.08	0.08
Slide D1 - 24.99	0.12	0.12
Slide C - 26.35	0.11	0.11

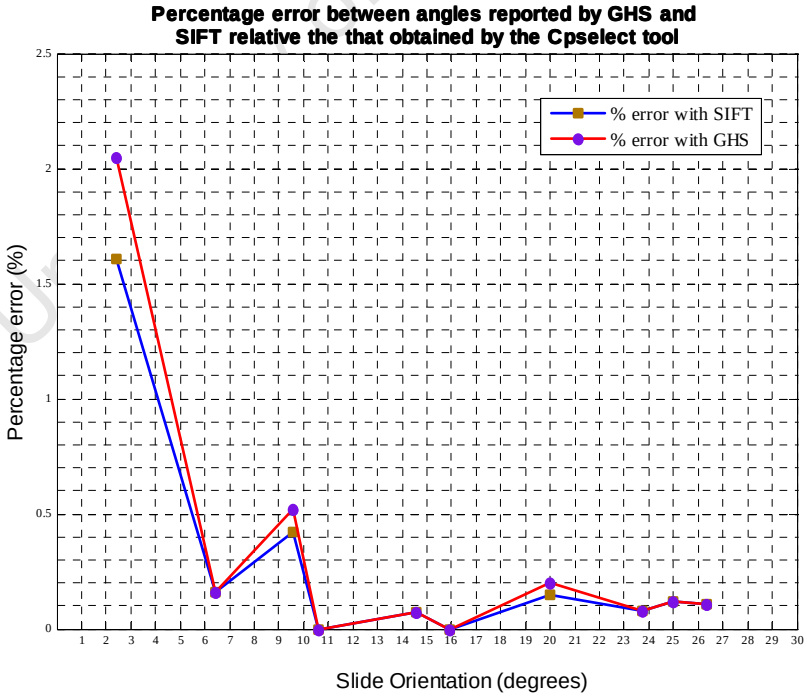


Figure 8.14 : Comparison of percentage error between angles reported by GHS and SIFT relative to that obtained by the Cpselect tool

The relative percentage error for orientation 0.01° for GHS is 100 % and it has not been included in the plot in Figure 8.14 since it would re-scale the y axis and make the figure less legible.

Table 8.19: Comparison of registration parameters by the average mean square error.

Image sub-set	Transformation estimation errors - GHS				Transformation estimation errors - SIFT			
	MS error (pixels^2)		RMS error (pixels)		MS error (pixels^2)		RMS error (pixels)	
	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.
Slide D1 - 0.01	9.15	3.09	2.98	0.53	0.38	0.55	0.56	0.26
Slide C - 2.44	10.35	3.26	3.17	0.53	0.59	0.79	0.69	0.33
Slide D1 - 6.43	11.47	3.25	3.35	0.50	0.35	0.47	0.54	0.25
Slide C - 9.59	9.16	3.69	2.97	0.60	0.89	1.64	0.79	0.51
Slide D1 - 10.58	9.18	3.06	2.99	0.51	0.31	0.24	0.53	0.16
Slide C - 14.58	11.81	3.34	3.40	0.49	0.54	0.79	0.65	0.34
Slide D1 - 15.91	9.17	2.79	2.99	0.47	0.31	0.34	0.52	0.19
Slide C - 20.00	10.84	3.39	3.25	0.51	0.64	0.67	0.73	0.34
Slide C - 23.72	13.33	4.02	3.61	0.56	0.77	1.32	0.75	0.46
Slide D1 - 24.99	11.51	3.36	3.35	0.51	0.31	0.24	0.53	0.17
Slide C - 26.35	12.96	4.14	3.55	0.59	0.87	1.56	0.79	0.50

There were no visible differences between the two image registration methods. Figure 8.15 shows a bar chart of the average mean square error variation with image orientation.

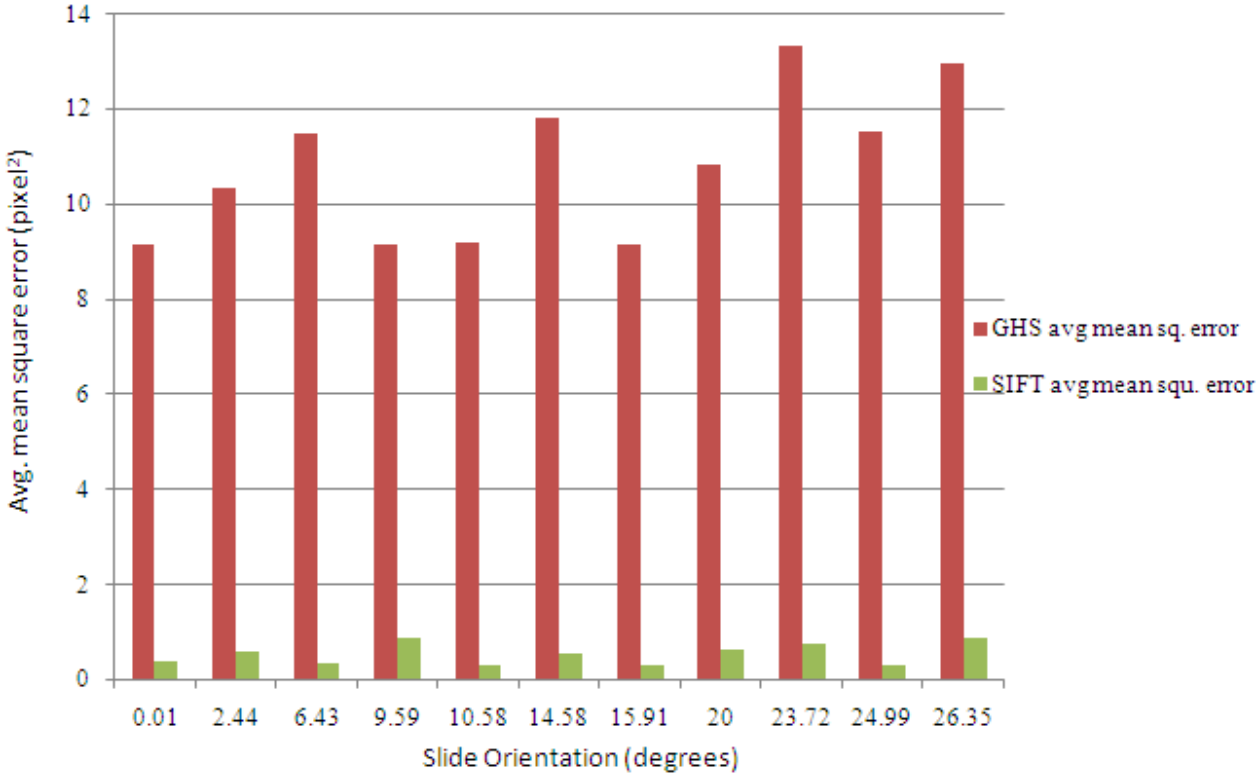


Figure 8.15: Variation of average mean square error obtained using the GHS and SFIT methods with image orientation

The object recognition performance on the individual slides, C and D1, is presented in Figure 8.16 and Figure 8.17 respectively.

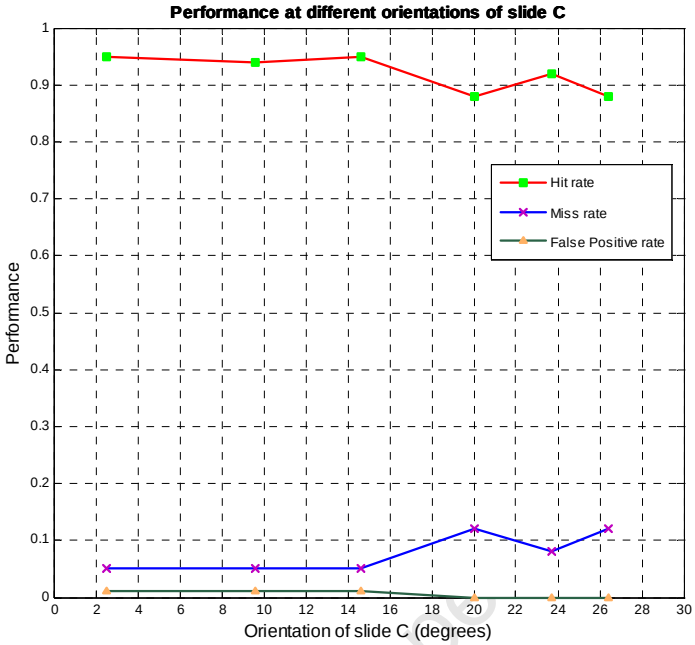


Figure 8.16: Object recognition performance at different orientations of slide C.

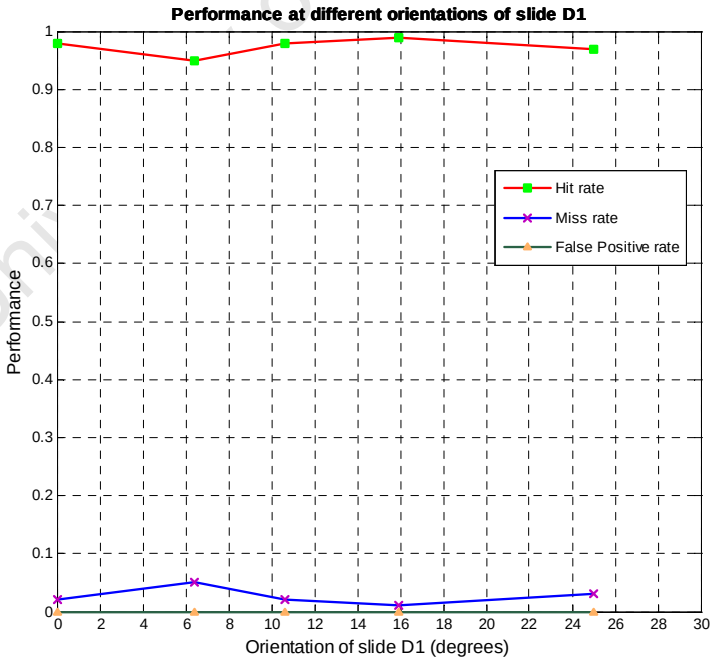


Figure 8.17: Object recognition performance at different orientations of slide D1.

8.4.3 Visual assessment of the registration parameters

Visual assessment was used as a rough indication of the correctness of the computed least-squares-fit transformation between a query image and the matching model image – image registration. This was done by overlaying the registered query image onto the matching model. An example superimposition is shown in Figure 8.18. Visual assessment of the registration parameters computed by the algorithm showed few or no visually detectable errors across all the image sets. Visual assessment was only used as a complement of the mean square error, which is a quantitative evaluation of image registration.

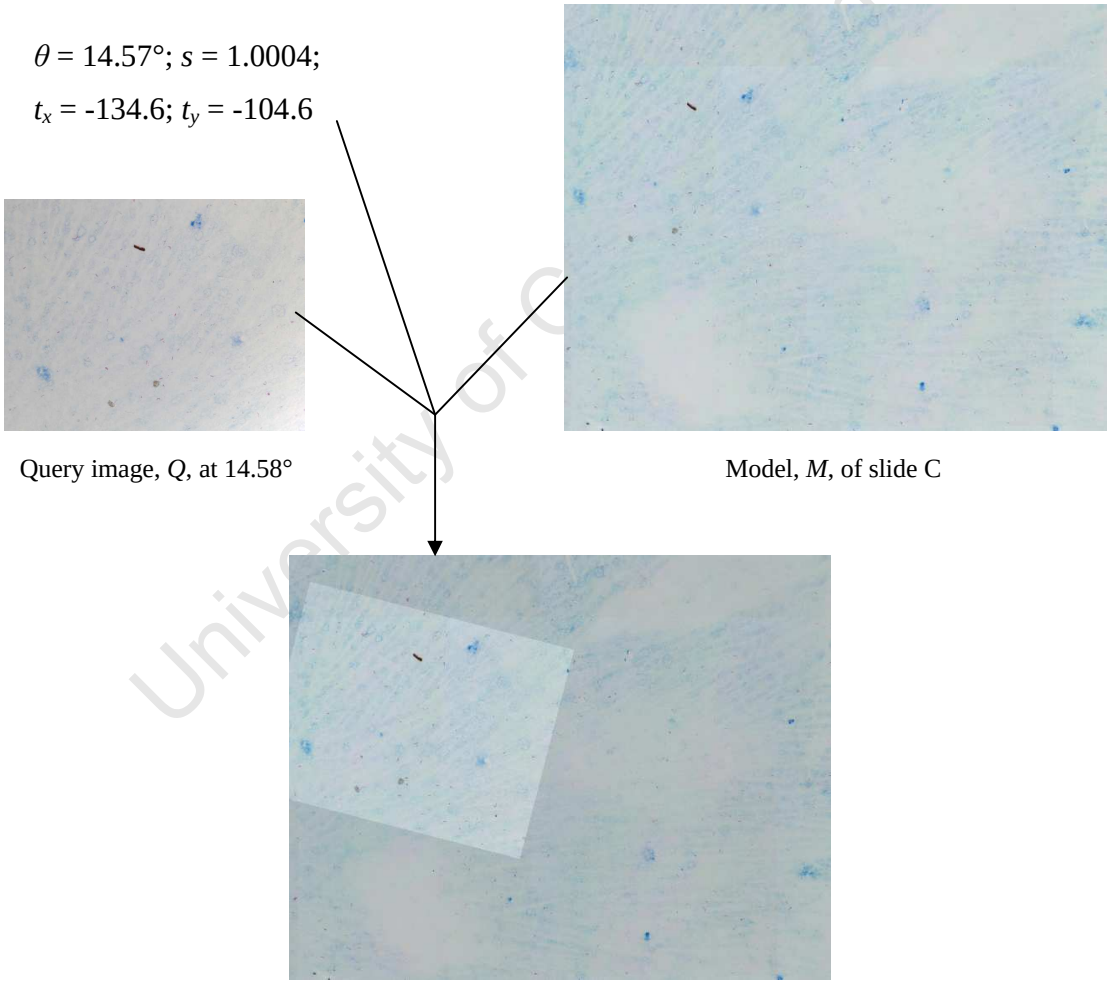


Figure 8.18: Visual assessment of image registration.

8.4.4 Processing time

The time taken by the algorithm to localise a query image ranged from 27s to 200s. The bulk of time was spent on the voting process. The time taken in the voting process is dependent on the number of bins accessed and the number of entries in those bins. The number of bins accessed is in turn mainly dependent on the number of feature points in the query image (Section 5.2). The more feature points present, the more bins accessed in the database and hence more voting which in turn translates to more time taken.

9. Summary and Discussion

This chapter summarises and provides a discussion of the auto-stitching results and the object recognition results and describes how the methods would be used in a fully automatic implementation.

9.1 Offline pre-processing stage

9.1.1 Auto-stitching to form the virtual slide map

To construct the virtual slide map in the pre-processing stage requires image stitching. Manual image stitching, although highly reliable, is very time consuming and laborious and therefore auto-stitching is preferred. Two auto-stitching methods were compared; namely the geometric hashing scheme and the scale invariant feature transform (SIFT).

Comparing the two virtual slide maps of slide O, SIFT VSO and GHS VSO, constructed by each of the methods, showed good agreement between the two. They were almost identical in size, i.e. and 4049 x 5976 and 4044 x 5979 respectively and there were no detectable differences between the two VSO images. A quantitative test of ratios between the sides of corresponding triangles in the VSOs further provided evidence of strong similarity between the two VSOs. The average difference in the perimeter of the corresponding triangles was found to be only 4 pixels which is too small to be visually detectable. Based on these results, the stitching quality of the methods may be considered similar.

In the object recognition tests using both the VSOs, a hit rate of over 90% was achieved with an average mean square error of less than 11 pixel^2 corresponding to a root mean square registration error of 3.3 pixels - this translates to less than 0.001% error since a given FOV image is 1030 x 1300 1.34 x 10^6. Since the conventional method for bacillus detection is visual (Section 8.4.3), and there is no visible difference at this level of error, it can be concluded that the error is acceptable and that both the GHS and SIFT auto-stitching methods are suitable for constructing quality virtual slide maps for the auto-positioning application. Closely comparing the performance of the two VSOs, it was found that the same two query images were missed using both the VSOs. One additional

query image was also missed using the GHS VSO. This query image comprised 16 feature points and 8 basis attempts were required to match it correctly using the SIFT VSO. This indicates that the algorithm's discriminative power for this query image is low (Section 8.4.1.3) and the correct model kept being eliminated during verification load reduction, even with 10 basis attempts using the GHS VSO, and therefore it was a miss with the GHS VSO.

The registration parameters of both the VSOs were almost identical as seen in Table 8.10.

Based on these findings, the GHS auto-stitching scheme can be considered to be comparable to the SIFT auto-stitching scheme and both are suitable in constructing virtual slide maps for the auto-positioning application in TB microscopy.

9.1.2 Pre-processing of the models of the virtual slide map

As shown in Section 8.2.3, the filtering process proves to be crucial in greatly reducing unnecessary entries into the database, permitting faster execution of voting and hence of the object recognition algorithm. By reducing the number of entries, the size of the database is also reduced which allows more models to be stored in the database and therefore allows a larger rectangular region of the slide to be covered even under tight computer RAM constraints. In addition to the area and eccentricity filters, other types of filters may be used to further reduce non-bacillus objects. Although this would further benefit the database size, it is likely to lower the discriminative power of the algorithm (Section 8.4.1.3) since query images would have fewer feature points and hence the likelihood of misses and false matches would increase.

9.2 Online localisation stage - Object recognition

The object recognition algorithm was tested on both query images extracted directly from the virtual slide map and real query images to check the robustness of the algorithm to rotation, displacement, illumination changes and noise.

9.2.1 *Query images Set 1*

The hit rate obtained for *Query images Set 1* was above 90% for both slide C and slide D1. Since these query images were obtained directly from the virtual slide map, the algorithm was expected to report a hit rate of 100 % (i.e. correctly matching every single query image in this set). However, a hit rate of 100 % was not achieved because the constructed maps contained areas difficult for navigation, which included areas having no or very few bacilli or other objects retained after filtering.

The methods of verification load reduction proved to be highly effective in reducing the candidate model-basis combinations (*CMBs*) to be verified from thousands to less than 10 as shown in Section 8.4.1.2. Therefore, the verification load reduction process is crucial because:

- It eliminates many unlikely matches to the basis, *Bq*, of query image *Q*, and hence reduces the chances of false matches.

- By reducing the number of *CMBs*, only a few *CMBs* need to be verified, which speeds up the object recognition algorithm.

Most of the images that were missed or falsely matched by the algorithm contained relatively fewer bacilli and thus fewer feature points. Therefore, the likely cause for not hitting the matching model is the poor discriminative power, resulting in the matching model being eliminated during the verification load reduction process in all 10 basis attempts.

To overcome this problem and to improve the performance of the algorithm, quadtree enhancement may be employed. As shown in Figure 8.10 quadtree enhancement greatly improves the discriminative power especially for images with few feature points. However, this requires prior knowledge of the estimated region in which the query image lies. In practice, if the virtual slide map covers a small area on the actual slide, then during the auto-positioning process it would be difficult to estimate in which

quadrant the current field-of-view (which acts as the query image) lies and quadtree enhancement would be unsuitable. However, if the virtual slide map covers a large area on the actual slide, quadtree enchantment at various depths (Section 2.8.2) would greatly benefit the performance of the algorithm because it would be easier to identify visually which quadrant or even sub-quadrants the current FOV lies in and to allow only that quadrant/sub-quadrant to participate in the voting stage.

9.2.2 *Query images Set 2*

The algorithm was tested of real images obtained at different orientations and at different times. As explained in Section 5.5.1.1, these images differed from the model images in several ways: rotation and translation, illumination, and focusing. All these factors also contribute to noise in the images and hence these query images are considerably noisy relative to their matching models. Despite all these variations, the hit rate (HR) for all the sub-sets of images considered did not drop below 88%. For some sub-sets of images, a HR as high as 98% was achieved. These results show the robustness of the algorithm to large rotational and translation changes, illumination changes and noise. Considering that manual localisation of an FOV on the virtual slide map is highly time consuming and tedious, a true positive rate of 88% is acceptable and if the method is included in an automated system, it is likely to improve efficiency in TB screening

9.2.2.1 Effect on hit rate with increasing orientation angle

With slide C, as seen in Figure 8.16, a general trend of decreasing HR with increasing slide orientation was observed. One possible cause for this may be reduced discriminative power for query images at large orientation angles as illustrated in Figure 9.1.

Slide C had few bacilli per field of view and would have few feature points per query mage. As seen in Figure 9.1, if the query image, *Q*, is at 0° orientation (orange outline) then it would be fully contained in models M_1 and M_2. If there are sufficient corresponding feature points between *Q* and either M_1 or M_2, then these models would receive a significant number of votes and would not be eliminated in the verification

load reduction process i.e. the algorithm would have enough discriminative power. Hence under these conditions *Q* is likely to be successfully matched by the algorithm.

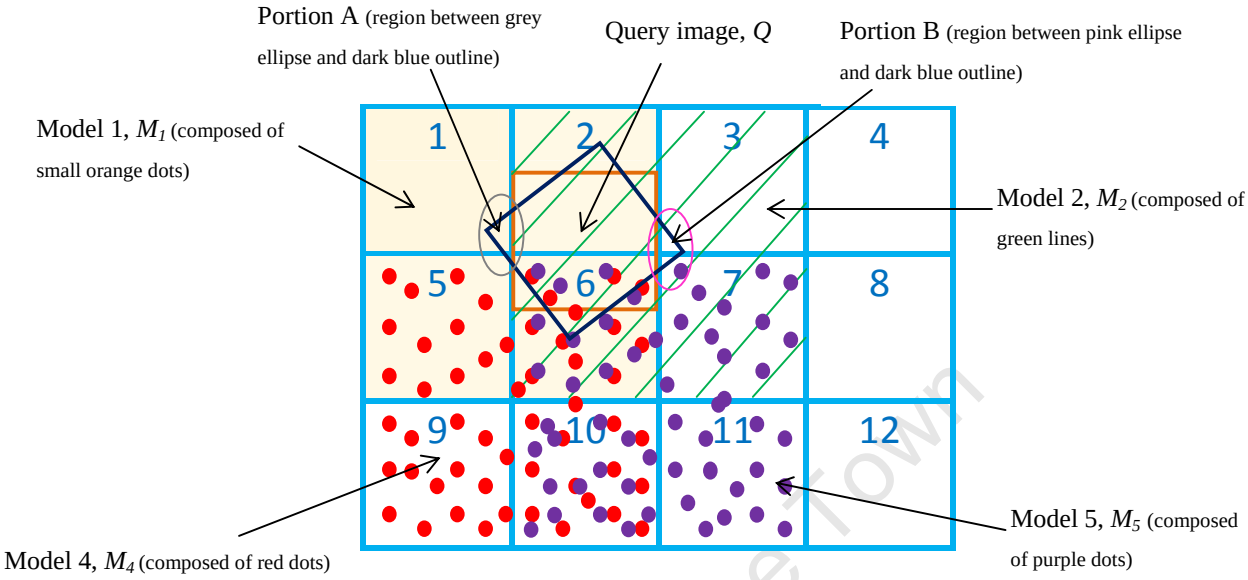


Figure 9.1: Illustration of query image with a large rotational change relative to its matching model.

For an orientation other than 0°, *Q* (outlined in dark blue) would be largely contained in these models but no longer fully contained in them. For a given orientation (other than 0°) of *Q*, Portion A represents the portion of *Q* that does not lie in model M_2 while Portion B represents the portion of *Q* that does not lie in model M_1. As the orientation angle of *Q* increases, these portions become bigger which means less of *Q* is contained in any one of these models. Hence as the orientation angle increases, the number of corresponding feature points between *Q* and either M_1 or M_2 decreases. Therefore, there would be insufficient feature points to ensure that these models do receive a significant number of votes. Owing to the reduced discriminative power, these models would be eliminated during verification load reduction and hence *Q* at large orientation angles is unlikely to be matched by the algorithm. This is the likely reason why the hit rate drops for slide C as the orientation angle increases.

With the sub-sets of images of slide D1, the hit rate did not drop below 92%. This is because query images of slide D1 contained relatively more bacilli than those of slide C

and hence more feature points which resulted in sufficient discriminative power even at high orientation angles.

9.2.2.2 Effect on hit rate with varying image quality

Another factor affecting the performance of the algorithm is the quality of the images which affects the segmentation results. Image segmentation was performed using the quadratic classifier (Khutlang et al. 2010) after which feature points were extracted (Sections 4.3 and 4.4). Segmentation results of the same field-of-view captured at different times and under different conditions vary and hence the number of feature points extracted from those FOV images varies. Therefore, a real query image may contain additional or fewer feature points (relative to the matching model) that would affect the voting process and result in a false match or a miss. Thus the segmentation results, which vary with the quality of images, greatly affect the performance on the object recognition algorithm. This may explain the sudden change in hit rate with the sub-sets slide C - 23.72 (Figure 8.16) and slide D1 - 6.43 (Figure 8.17).

9.2.2.3 Image registration

The algorithm determines the matching model to the query image and computes the registration parameters simultaneously. The quality of the estimated transformation was measured using the mean square error, which incorporates all the errors in all the registration parameters. It was also compared to those obtained using the SIFT scheme. Computation of the registration parameters using SIFT can be performed only after the algorithm has found the matching model (Section 5.5.2.3).

For a given sub-set of images in *Query images Set 2*, the average registration angle parameter computed by the algorithm and the SIFT scheme were found to be comparable to that established manually using the *cpselect* tool in MATLAB (
Table 8.17 and Figure 8.14). The average scale parameter was found to be 1.000 with a standard deviation of < 0.002 in all the cases which was expected since all images were captured at the same magnification of 40x. The translation parameters vary from query image to image since a model is 4 times as large and hence the query image could lie

any where within the matching model. All the registration parameters' errors are included within the mean square error.

The mean square error represents the positional discrepancies between the feature points in the matching model and in the transformed query image (Section 2.8.3.2). As explained in Section 9.2.2.2, the segmentation results of the same field-of-view captured at different times vary and hence the number of feature points extracted from those FOV images varies and there are positional discrepancies between corresponding feature points in those images. Therefore, the mean square error also incorporates the positional discrepancies introduced by the image segmentation technique.

The average mean square error computed by the algorithm for a given sub-set of images did not exceed 14 pixels^2 (1.02 μm^2); corresponding to a root mean square error of 3.7 pixels. This means that a feature point in the registered query image lies within a radius of 3.7 pixels of the corresponding feature point in the matching model image. The SIFT scheme on the other hand offered a more accurate estimation of the transformation with a mean square of less than 1 pixel^2. However, considering that a given FOV image measures 1030 x 1300 = 1.34 x 10^6 pixels^2, a mean square error of 14 pixels^2 amounts to only 14 / 1.34 x 10^6 = 0.001% error. Since the conventional method for bacillus detection is visual (Section 8.4.3), and there is no visible difference at this level of error, it can be concluded that the error is acceptable; if the method is to be included in an automated system, it will be judged by the final outputs of such a system.

If more accuracy is desired, the SIFT scheme can be used. However, as mentioned above this is an additional step and hence additional time is required since the SIFT can be used to estimate the transformation only after the matching model has been found by the algorithm. Therefore, the increase in accuracy may not be fully justified depending on the application.

9.2.3 Processing time

The time taken by the algorithm to localise a query image ranged from 27s to 200s. Although it may be argued that a human can bring a desired field on the slide to the field-of-view of the microscope faster than the algorithm, the algorithm is likely to outperform a human when multiple regions-of-interest on the same slide need reviewing or re-examination on one reload of the slide. This is because once the current FOV is localised, it acts as a global point of reference for the entire slide which can be used to easily and quickly bring any desired fields to the FOV of the microscope. A human on the other hand would have to manually search each time a different field needs to be brought to the FOV of the microscope.

In practice, in the event of a miss by the algorithm, i.e. failure to localise the current field-of-view in 10 basis attempts, the algorithm can be adapted so that if the current field-of-view (the query image) is not matched in 10 basis attempts, the microscope is directed to move to a different field. The image of the new field would act as a new query image which is fed to the object recognition algorithm and localisation re-preformed.

The results of the various tests performed for object recognition show that the algorithm is inherently insensitive to changes in slide orientation and placement, which are likely to occur in practice as it is impossible to place the slide in exactly the same position on the microscope at different times. It also has high tolerance to illumination changes and it is robust to noise.

The object recognition results using slide D1, which was automatically stitched using the GHS scheme, were comparable to those of slide C which was manually stitched. This further shows that the GHS auto-stitching scheme is suitable for automatically stitching a large number of microscopy TB images to construct large virtual slide maps for the application in auto-positioning.

9.3 Auto-positioning to a desired field on the slide

This section aims to explain how the methods and algorithms developed can be used for auto-positioning in TB microscopy.

During the scanning process, the slide was placed parallel to the XY stage (Section 4.1.1) and this orientation represents 0°. Under this setting, the orientation of the XY stage relative to the constructed virtual slide map (blue frame) is shown in Figure 9.2.

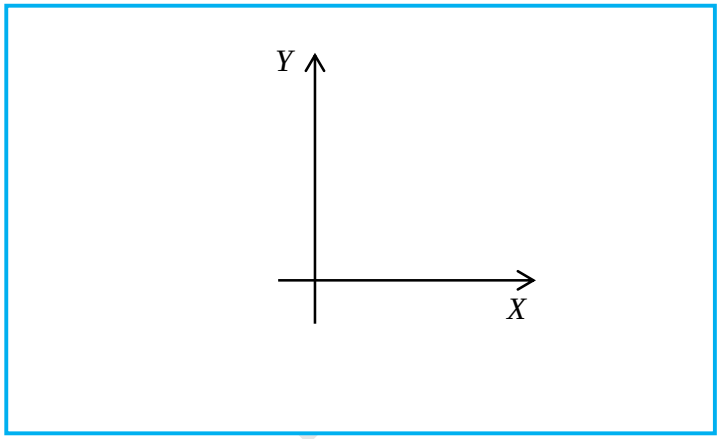


Figure 9.2: Orientation of the XY stage relative to the virtual slide map (blue frame).

A desired field (*DF*) can be virtually marked on the virtual slide map. The slide is placed onto the slide holder. To bring the *DF* to the field-of-view of the microscope, two features need to be known as explained in Section 2.3; namely a point of reference and the co-ordinates of the desired field relative to that reference. Finding a point of reference is the core component in achieving auto-positioning.

9.3.1 Current field-of-view as a point of reference

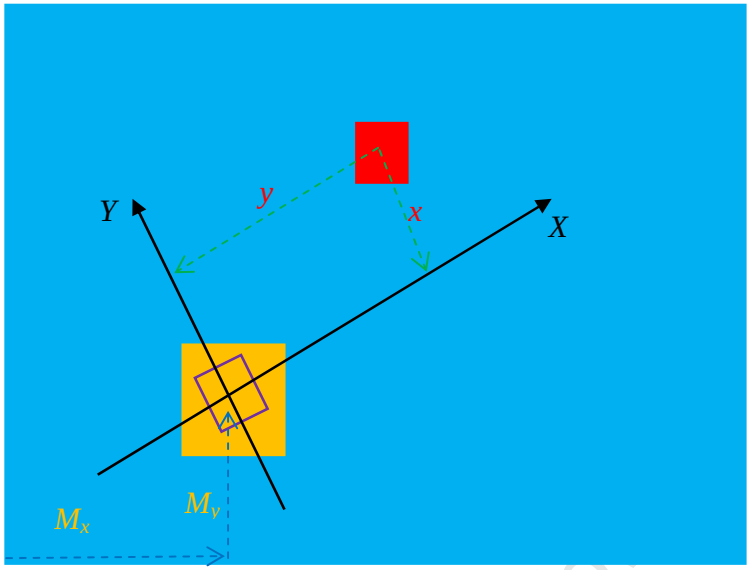
For the algorithm developed, the current field-of-view acts as a point of reference. This requires the localisation of the current FOV on the virtual slide map. It is achieved by determining the best matching model to the query image (current FOV, Figure 9.3) and the similarity transformation linking the best matching model to the query image, as explained in Chapter 5.

9.3.2 Coordinates of DF with respect to the current FOV

The angle parameter of the computed least-squares-fit transformation is used to re-orientate the virtual XY stage on the virtual slide map. The virtual XY coordinate frame is then placed with its origin at the middle of the query image which is overlayed on the matching model on the virtual slide map as illustrated in Figure 9.3. Using the re-oriented virtual XY stage on the virtual slide map, the coordinates *(x, y)* of the desired field (*DF*) relative to the just localised FOV (query image) can be obtained as shown in Figure 9.3.

These co-ordinates can be fed to the respective motors to automatically and accurately drive the XY stage such that *DF* (red in Figure 9.3) is brought to the field-of-view of the microscope, hence achieving auto-positioning.

For this study, a conventional microscope lacking a mortised XY stage was used. In the absence of a motorised XY stage, user interaction is required to manually move the XY stage to bring the *DF* to the field-of-view of the microscope. Accurate sub-millimetre manual movement of the XY stage is almost impossible and highly unrepeatable and therefore, this final step of auto-positioning can not be achieved using a conventional microscope.



	Complete virtual slide map, V_n
	Virtually marked Desired Field (*DF*)
	Outline of the current field-of-view as seen by the camera. Its image forms the query image, *Q*, which is localised by finding the best matching model
	Best matching model, *M*, to the query image, *Q*. *M* is has coordinates (M_x, M_y) on the virtual slide map.

Figure 9.3: Illustration of auto-positioning.

10. Conclusions and Recommendations

The aim of the project was to develop a method suitable for auto-positioning for bright field microscopy for TB detection, which uses ZN-stained sputum smear slides. The biggest task in auto-positioning is generating a suitable point of reference regardless of the slide displacements and orientation (similarity transformation), due to improper placement of the slide under the microscope usince the slide, and other microscope conditions . This was successfully achieved by jointly using a virtual slide map and an object recognition algorithm. With this method, any given filed-of-view can act as a point of reference by localising it - using the object recognition algorithm - on a previously constructed virtual slide map; the virtual slide map being a digital replication of the actual slide. The coordinates of a desired field relative to the just localised FOV (point of reference) can be determined on the virtual slide map and fed to the motors of the XY stage to bring that field to the field-of-view of the microscope, hence achieving microscope auto-positioning. In the absence of a motorised stage like in the conventional microscope used, this final component, which is trivial with a motorised stage, can not be achieved since manual movement of the stage in inaccurate and not repeatable.

The time taken by the algorithm may be reduced by re-writing the algorithm with attention paid to implementing routines in a time efficient manner. The use of object-oriented programming could be explored to limit looping in algorithms. All algorithms were written in MATLAB; a future development might be to convert all software to another programming language that would allow faster execution such as the C programming language.

Additional time gains would be achieved if parallel processing is employed during the voting stage. This would not only reduce the time spent in navigation but would also address storage space and RAM limitations since the database (which comprises 4 hash tables) would reside on multiple computers. Consequently, it would allow the model representation and storage of larger virtual slide maps covering the entire sputum smear on the actual slides.

The ZEISS Axioskop 2 was used to capture images of the different fields on the actual slide to construct a virtual slide and the same microscope was used to capture sets of query mages on that slide. The flexibility of the algorithm can be investigated by studying its performance using field-of-view images (query images) captured from a microscope different to the microscope used to construct the virtual slide map. However, since digital cameras capture and register light differently, bacilli and background may have different colour properties in images obtained with different cameras. This is likely to significantly affect the segmentation results and hence the location and number of feature points extracted, and therefore likely to degrade the ability of the algorithm to localise query images obtained from a different microscope. The normalisation of colour images from different microscopes will allow the generalisation of algorithms for different platforms and save time in the construction of another virtual slide map of the same slide every time a different microscope needs to be used. Other more robust segmentation techniques could also be explored.

All the images in the experiments were captured at the same magnification of 40x. The capability of the algorithm to localise query images captured at different magnification levels can be investigated. The geometric hashing technique used is theoretically invariant to scale. However, as shown in (Khutlang 2009), segmentation results of images acquired at different magnification levels vary considerably and hence this would affect the performance of the object recognition algorithm. Segmentation techniques robust to scale may be explored to make the algorithm robust to scale and hence allow TB screening to be performed at different magnification levels.

The algorithm's performance was relatively low for query images with relatively fewer bacilli and hence fewer feature points which reduced the discriminative power of the algorithm. Techniques of extracting stable feature points from the background of these images might be explored which would greatly benefit the algorithm, especially for TB sputum slides that contain no or very few bacilli per field. Ideally, the feature extraction techniques should be capable of extracting not too few but also not too many feature points per image otherwise computing requirements would escalate steeply. Methods to

extract the best *C* feature points, where *C* is a constant, from every image can be investigated.

Image matching techniques such as template matching and sub-graph matching can be explored to improve the system's performance. The use of fuzzy logics can also be investigated.

The virtual slide map generation and the object recognition algorithms developed may be used in combination for auto-positioning in TB microscopy to provide a means for technicians to verify the results of automated bacillus detection algorithms and to perform TB screening quality control tests. The methods may also be used to compare bacillus detection accuracy in the same field-of-view at different settings of a microscope or across microscopes.

References

Autostitch. 2010, *Autostitch: A new dimension in automatic image stitching*. Available: http://cvlab.epfl.ch/~brown/autostitch/autostitch.html#publications [2010, 06/10].

Bay, H., Tuytelaars, T. and Van Gool, L. 2006, "SURF: Speeded up robust features", *Proceedings of the Ninth European Conference on Computer Vision*, Graz, pp. 404-417.

Begelman, G., Lifshits, M. and Rivlin, E. 2006, "Visual positioning of previously defined ROIs on microscopic slides", *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 42-50.

Blum, H. 1967, "A transformation for extracting new descriptors of shape", *Models for the Perception of Speech and Visual Form*, vol. 19, no. 5, pp. 362-380.

Bouix, S. and Siddiqi, K. 2000, "Divergence-based medial surfaces", *Proceedings of the European Conference on Computer Vision*, Dublin, pp. 603-618.

Brown, M. and Lowe, D.G. 2007, "Automatic panoramic image stitching using invariant features", *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59-73.

Brown, M. and Lowe, D.G. 2003, "Recognising panoramas", *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Nice, pp. 1218.

Cheng, F.H. 1996, "Point pattern matching algorithm invariant to geometrical transformation and distortion", *Pattern Recognition Letters*, vol. 17, no. 14, pp. 1429-1435.

Costa, M.S., Haralick, R.M. and Shapiro, L.G. 2002, "Optimal affine-invariant point matching", *Proceedings of the Tenth International Conference on Pattern Recognition*, New Jersey, pp. 233-236.

Costantino, S., Heinze, K.G., Martínez, O.E., De Koninck, P. and Wiseman, P.W. 2005, "Two-photon fluorescent microlithography for live-cell imaging", *Microscopy Research and Technique*, vol. 68, no. 5, pp. 272-276.

De Berg, M., Cheong, O., Van Kreveld, M. and Overmars, M. 2008, *Computational Geometry: Algorithms and Applications*, Springer-Verlag New York Inc.

Dee, F.R., Lehman, J.M., Consoer, D., Leaven, T. and Cohen, M.B. 2003, "Implementation of virtual microscope slides in the annual pathobiology of cancer workshop laboratory", *Human Pathology*, vol. 34, no. 5, pp. 430-436.

Doerrer, R. 2007, *System and Method for Re-locating an Object in a Sample on a Slide with a Microscope Imaging Device*, United States Patent Application 20070076983.

Duin, R., Juszczak, P., Paclik, P., Pekalska, E., De Ridder, D. and Tax, D. 2004, *Prtools4, a matlab toolbox for pattern recognition, 2004*, .

Electron Microscopy Science. 2010, *Electron Microscopy Science*. Available: http://www.emsdiasum.com [2009, 09/30].

Fischler, M.A. and Bolles, R.C. 1981, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381-395.

Fitzpatrick, J.M. and West, J.B. 2002, "The distribution of target registration error in rigid-body point-based registration", *IEEE Transactions on Medical Imaging*, vol. 20, no. 9, pp. 917-927.

Forero, M.G., Cristobal, G. and Desco, M. 2006, "Automatic identification of Mycobacterium tuberculosis by Gaussian mixture models", *Journal of Microscopy*, vol. 223, no. 2, pp. 120-132.

Gonzalez, R.C., Woods, R.E. and Eddins, S.L. 2004, *Digital image processing using MATLAB*, Prentice Hall Upper Saddle River, NJ.

Graham, M.D. and Cook, D.D. 1985, *Automated microscopy system and method for locating and re-locating objects in an image*, United States Patent Application 4513438.

Hänscheid, T. 2008, "The future looks bright: low-cost fluorescent microscopes for detection of Mycobacterium tuberculosis and Coccidiae", *Transactions of the Royal Society of Tropical Medicine and Hygiene*, vol. 102, no. 6, pp. 520-521.

Harris, C. and Stephens, M. 1988, "A combined corner and edge detector", *Proceedings of the Alvey Vision Conference*, Manchester, pp. 50.

Hartley, R. and Zisserman, A. 2003, *Multiple View Geometry in Computer Vision*, Cambridge University Press New York, NY, USA.

He, J., Zhou, R. and Hong, Z. 2003, "Modified fast climbing search auto-focus algorithm with adaptive step size searching technique for digital camera", *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 257-262.

Hugin. 2010, *Panorama Photo Stitcher*. Available: http://hugin.sourceforge.net/ [2010, 06/10].

Khutlang, R. 2009, *Image Segmentation and Object Classification for Automatic Detection of Tuberculosis in Sputum Smears*, MSc Thesis. Department of Human Biology, University of Cape Town.

Khutlang, R., Krishnan, S., Dendere, R., Whitelaw, A., Veropoulos, K., Learmonth, G. and Douglas, T. 2010, "Classification of Mycobacterium tuberculosis in images of ZN-stained sputum smears", *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 4, pp. 949-957.

Lafore, R. 1999, *Sam's Teach Yourself Data Structures and Algorithms in 24 Hours*, Sam's Publishing. Indianapolis, Indiana, USA.

Lamdan, Y. and Wolfson, H.J. 1988, "Geometric hashing: A general and efficient model-based recognition scheme", *Proceedings of the Second International Conference on Computer Vision*, New York, pp. 238-249.

Lamdan, Y. and Wolfson, H. 1991, "On the error analysis of geometric hashing", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, pp. 22-27.

Lifshits, M., Goldenberg, R., Rivlin, E., Rudzsky, M. and Adel, M. 2004, "Image-based wafer navigation", *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 3, pp. 432-443.

Loncaric, S. 1998, "A survey of shape analysis techniques", *Pattern Recognition*, vol. 31, no. 8, pp. 983-1001.

Lowe, D.G. 2004, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110.

Ma, B., Zimmermann, T., Rohde, M., Winkelbach, S., He, F., Lindenmaier, W. and Dittmar, K.E.J. 2007, "Use of autostitch for automatic stitching of microscope images", *Micron*, vol. 38, no. 5, pp. 492-499.

Mehrotra, H., Majhi, B. and Gupta, P. 2010, "Robust iris indexing scheme using geometric hashing of SIFT keypoints", *Journal of Network and Computer Applications*, vol. 33, no. 3, pp. 300-313.

Microlab. 2010, *CellFinder Microscope Slides*. Available: http://www.antenna.nl/microlab/index-uk.html [2009, 09/30].

Mikolajczyk, K. and Schmid, C. 2001, "Indexing based on scale invariant interest points", *Proceedings of the Eighth IEEE International Conference on Computer Vision*, Vancouver, pp. 525-531.

Osibote, O., Dendere, R., Krishnan, S. and Douglas, T. 2010, "Automated focusing in bright-field microscopy for tuberculosis detection", *Journal of Microscopy*, vol. 240, no. 2, pp. 155-163.

Pal, N.R. and Pal, S.K. 1993, "A review on image segmentation techniques", *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294.

Panorama Maker. 2010, *Panorama Maker 5 pro*. Available: http://www.arcsoft.com/estore/software_title.asp?ProductCode=PMK5PRO [2010, 06/15].

Roth, P.M. and Winter, M. 2008, *Survey of appearance-based methods for object recognition*, Inst. for Computer Graphics and Vision, Graz University of Technology, Austria, Tech.Rep.ICG-TR-01/08.

Russell, M.J. 2006, *Auto-focusing and image segmentation in microscopy for automatic detection of tuberculosis in sputum smears*, MSc Thesis. Department of Human Biology, University of Cape Town.

Steingart, K.R., Henry, M., Ng, V., Hopewell, P.C., Ramsay, A., Cunningham, J., Urbanczik, R., Perkins, M., Aziz, M.A. and Pai, M. 2006, "Fluorescence versus conventional sputum smear microscopy for tuberculosis: a systematic review", *The Lancet Infectious Diseases*, vol. 6, no. 9, pp. 570-581.

Sun, T.H., Horng, H.C., Liu, C.S. and Tien, F.C. 2009, "Invariant 2D object recognition using KRA and GRA", *Expert Systems with Applications*, vol. 36, no. 9, pp. 11517-11527.

Sun, T.H., Liu, C.S. and Tien, F.C. 2008, "Invariant 2D object recognition using eigenvalues of covariance matrices, re-sampling and autocorrelation", *Expert Systems with Applications*, vol. 35, no. 4, pp. 1966-1977.

Todar, K. 2005, *Todar's Online Textbook of Bacteriology*, On-line book, www.textbookofbacteriology.net.

Veropoulos, K. 2001, *Machine Learning Approaches to Medical Decision Making*, PhD Thesis. Department of Computer Science, University of Bristol.

Veropoulos, K., Learmonth, G., Campbell, C., Knight, B. and Simpson, J. 1999, "Automated identification of tubercle bacilli in sputum. A preliminary investigation", *Analytical and Quantitative Cytology and Histology*, vol. 21, no. 4, pp. 277-282.

Weiss, M.A. 1999, *Data Structures and Problem Solving with C++*, 2nd edn, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Wolfson, H.J. and Rigoutsos, I. 1997, "Geometric hashing: An overview", *IEEE Computational Science & Engineering*, vol. 4, no. 4, pp. 10-21.

World Health Organisation. 2008, *Global tuberculosis control 2008: surveillance, planning and financing*. Available: http://www.who.int/tb/publications/global_report/2008/pdf/fullreport.pdf [2010, 09/10].

World Health Organisation. 2007, *Tuberculosis Facts Sheets*. Available: http://www.who.int/mediacentre/factsheets/fs104/en/ [2010, 09/10].

ZEISS. 2010, *ZEISS Digital Imaging: AxioVision*. Available: http://www.zeiss.de/axiovision [2010, 04/15].

Zitova, B. and Flusser, J. 2003, "Image registration methods: a survey", *Image and Vision Computing*, vol. 21, no. 11, pp. 977-1000.

Appendix

Contents of the MATLAB M-files on the accompanying CD.

OFFLINE PRO-PROCESSING STAGE

FOLDER - SIFT auto-stitch scheme.

a0mother_virtual_slide_creation6.m - auto-stitches colour images (which should be copied into this folder itself) using the SIFT scheme to form a virtual slide map. It extracts the appropriate portion-of-interest (*POI*) from the partial virtual slide map and detects SIFT keypoints in it. It then detects SIFT keypoints in the image, *I*, to be added, matches the keypoints to that of *POI*, registers it and finally stitches it to the *POI*, forming the new partial virtual slide map.

FOLDER - GHS auto-stitch scheme.

a0mother_virtual_slide_creation_my_method.m - auto-stitches colour images (which should be copied into this folder itself) using the GHS scheme to form a virtual slide map. It extracts the appropriate portion-of-interest (*POI*) from the partial virtual slide map, segments and filters it; extracts feature points, represents it using the feature points and stores it in the database. It then takes in the (segmented) image, *I*, to be added and matches it to the *POI*, registers it and finally stitches *I* to the *POI*, forming the new partial virtual slide map. To speed up the process, all the images, *I*, can be segmented and filtered using the m-files in folder titled 'Image segmentation' (below) prior to constructing the virtual slide map and saved as a MAT file named segfilim.mat in this folder.

FOLDER - Decomposition of virtual slide map.

virtual_colorslide__break_downv2.m – breaks down virtual slide map into models.

FOLDER - Image segmentation.

bhav_seg.m – segments a colour image (present in this folder) using the quadratic classifier (requires PRtools toolbox).

removal_of_darkspots_by_eccentricity_plus_erosion_slide_cV2.m – filters segmented objects using area and eccentricity filters.

FOLDER - Model image feature extraction, representation and storage in database.

All the filtered segmented model images of a virtual slide map should be saved as a single MAT file named models.mat in this folder after which the following algorithms can be executed.

afeature_pointsv04.m – performs the medial axis transform and extracts feature points from each model of the virtual slide map.

bahashtable_method11_2.m – constructs a database (memory pre-allocation).

bbasis_using_cells_with_all_xypointsv1_method11_2.m – represents a model using the geometric hashing technique and stores the representation in the database – database filling.

ONLINE LOCALISATION STAGE

Before performing this task the appropriate database in folder titled 'Databases' needs to be loaded into RAM. The following can then be executed.

cROI_image_feature_points_extraction_method3.m – takes in a segmented query image, *Q*; performs the medial axis transform and extracts feature points from it. Prior to

executing this m-file, the query image needs to be segmented and filtered using the m-files in the folder titled 'Image segmentation' (above) and saved as a MAT file named querry.mat in this folder.

localiseROIv3.m – runs the entire *online localisation process* calling the following functions:

dROI_point_extract_to_check_point_method11_2.m – selects an arbitrary basis, B_q, in Q and computes the invariant coordinates of the other feature points.

evoting_process_method11_2.m – uses the computed invariant coordinates to execute the voting process. For quadtree implementation, **evoting_process_method11_3.m** is used.

fobtaining_the_basis_pair_with_most_votesmethod11_2.m – sorts the voting results in descending order of votes received – sorted candidate list *CL1* - and carries forward the top fraction of the sorted *CL1*. The new candidate list is called *CL2* which contains the *CMB*s.

hfinding_and_transfomrin_ROI_to_be_model_method11_2.m – computes the similarity transformation using corresponding basis pairs between the query image and each *CMB* in *CL2*.

ifinding_and_transfomrin_ROI_to_be_model_method11_2.m – uses orientation and scale filters to remove highly unlikely candidate matches to query image.

jto_transformation_is_right_method11_2.m and
kimage_of_trasofrmed_ROI_method11_2.m – performs Voronoi tessellation and Delaunay triangulation to produce a putative set of corresponding points between query image and each *CMB*.

L1_to_get_best_tranfomration_mappingmethod11_2.m – executes RANSAC for query image and each *CMB*, computes least-squares-fit transformation (also computes the associated registration errors) and declares the matching model. The associated RANSAC m-files contained in the folder 'RANSAC functions' are called. These RANSAC files were modified versions of those written by Peter Kovesi. Available: http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/#robust.

FOLDER - Image registration using SIFT: after the matching model of the query image has been found.

a1find_tfromlsbest_using_sift.m – takes in a query image and its matching model image (both these images need to be present in this folder), detects the SIFT keypoints in both the images and finds the matching SIFT keypoints between the two images. It then calls RANSAC to remove outliers after which it computes the least-squares-fit transformation relating the two images and the associated registration errors. The function that detects SIFT keypoints and the function that finds the matching SIFT points between two images were obtained from http://www.cs.ubc.ca/~lowe/keypoints/ which are written by David Lowe and made available for research purposes.

FOLDER - Example images, slide O.

This folder contains images of slide O to illustrate the output of various steps of the object recognition scheme. It includes the images acquired, the constructed virtual slide maps, the model images generated, the segmented model images and the filtered segmented model images. It also includes the real query images captured, the segmented query images and the filtered segmented query images.