Published in final edited form as:

Comput Methods Programs Biomed. 2012 October ; 108(1): 38–52. doi:10.1016/j.cmpb.2011.12.017.

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

Bhavin Patel and Tania S. Douglas

MRC/UCT Medical Imaging Research Unit, Department of Human Biology, University of Cape Town, Observatory 7925, Cape Town, South Africa

Abstract

We address the location of regions-of-interest in previously scanned sputum smear slides requiring reexamination in automated microscopy for tuberculosis (TB) detection. We focus on the core component of microscope auto-positioning, which is to find a point of reference, position and orientation, on the slide so that it can be used to automatically bring desired fields to the field-of-view of the microscope. We use virtual slide maps together with geometric hashing to localise a query image, which then acts as the point of reference. The true positive 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 pixel² (corresponding to 1.02 μm²). The algorithm is inherently robust to changes in slide orientation and placement and showed high tolerance to illumination changes and robustness to noise.

Keywords

Geometric hashing; microscope positioning; virtual slide map

1 INTRODUCTION

In high tuberculosis (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 low cost of equipment [1]. Automation of TB screening would speed up the screening process and prevent errors due to operator fatigue, improving efficiency and accuracy. *Mycobacterium tuberculosis* expectorated in sputum is visible 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 [2]. In ZN-stained sputum smears, acid-fast bacilli stain red against a blue background (Fig. 1).

An automated microscope for TB detection would include a motorised stage, an image capture unit and algorithms for auto-focusing [3], segmentation [4]–[7], classification [4], [5], [8] and auto-positioning.

© 2012 Elsevier Ireland Ltd. All rights reserved.

Corresponding author: Tania .S. Douglas, Tel : (+27) 21 4066541, Fax : (+27) 21 4487226, tania@ieee.org.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Microscope positioning is the process of bringing a desired region on the slide to the field-of-view of the microscope. It has several applications: for repositioning to regions-of-interest of a previously scanned slide for reviewing or re-examination; 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; and as a practical tool to assist an operator to reposition to a field-of-interest allowing him/her to manually verify the output of bacillus detection algorithms. Microscope positioning can be a time consuming, difficult and poorly repeatable task if manually performed. Existing techniques that enhance microscope positioning use specially manufactured slides. These include Field Finder slides [9] and CellFinder slides [10], which suffer from limited precision of positioning and do not achieve automated positioning [11].

Automated microscope positioning would save operators' time and provide greater accuracy and repeatability. The key step in microscope auto-positioning is finding a reliable point of reference - position and orientation - on the slide which can then be used to bring desired fields-of-interest to the field-of-view of the microscope. An automated approach to positioning uses a virtual slide map – a digital replication of the smear on the glass slide [11]–[14]. A virtual slide map is a large panorama created by capturing images of the different fields on the slide followed by stitching them. Once the current field-of-view (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 microscope, thus achieving auto-positioning.

Doerr [14] proposed a system that uses a fiduciary marker as a reference point. Fiduciary markers can be large and easy-to-track features selected near the desired feature or region or may be created prior to scanning of the slide as done in [15] using two-photon optical microlithography. The use of fiduciary markers is time consuming since the system first needs to find the fiduciary marker before it can be used as a reference point. Another drawback to the technique proposed in [15] is that it may suffer from artefacts that may arise during the creation of fiduciary markers.

We present a method that uses model-based object recognition by the geometric hashing technique [16]–[18] to achieve auto-positioning in bright-field TB microscopy. This technique has been applied to microscopy of integrated circuits [13] and microscopy of lung and prostate tissue [11]. Both these papers used a stepper motor-controlled XY stage to capture images of adjacent fields and joined them together to form a virtual slide map. No image stitching was performed. Secondly, although the virtual slide maps were constructed using real images, query images were extracted from the virtual slide map itself and image distortions were simulated. We not only apply the method to TB microscopy, a new application area, but we also test the algorithm with real distorted query images. Additionally, we test and use geometric hashing to auto-stitch overlapping images of adjacent fields to construct a virtual slide map, to accommodate the use of a microscope lacking precise XY stage movement. To the authors' knowledge, this is the first application of the geometric hashing technique 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. In this study, we also compare our method to the scale invariant feature transform, SIFT, [33], which is a widely used matching scheme.

In our algorithm, after the virtual slide map is constructed, it is broken down into models whose descriptions, invariant to similarity transformations, are stored in a hash table. Localisation is then accomplished by model-based object recognition which involves finding

a matching model to a given query field, which is accelerated by indexing. Our algorithm is robust to slide rotation and displacement and also to a wide range of illumination variations.

The paper is organised as follows: Section 2 describes our approach for localisation of a FOV which will act as a point of reference. This section has been sub-divided into an offline pre-processing stage and an online localisation stage. Experimental results obtained using our method are presented and discussed in Section 3 and conclusions are drawn in Section 4.

2 LOCALIZATION AS A MODEL-BASED OBJECT RECOGNITION TASK

Localisation involves finding the coordinates of a given field-of-view on a virtual slide. This is equivalent to matching a small partial image to a larger image. One applicable and most widely used matching technique is the scale invariant feature transform, SIFT [33]. It uses the texture of the image to extract feature points, which are referred to as keypoints, and forms an invariant (to similarity transformation) 128 dimensional descriptor vector around each detected keypoint. SIFT keypoints can be extracted from the virtual slide and stored in a database. To localise an FOV, SIFT keypoints can be extracted from it and each keypoint independently matched to the database of keypoints. The corresponding keypoint would reveal the location of the FOV on the virtual slide. However, a single FOV image of a ZN-stained sputum slide 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 a large number of keypoints and thus would have high computer RAM requirements. Additionally, a very large database of keypoints results in a high percentage of false matches of the query keypoints [33]. 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 [19].

To simplify and permit the execution of the matching process under computer RAM constraints, model-based recognition was adopted. Model-based recognition involves finding a correspondence between a query object and a set of pre-defined objects referred to as 'models' [18]. Models can be defined in terms of various properties such as shape, colour, texture, etc. These properties are also used to find a good correspondence. The FOV localisation process can be formulated as a recognition task as follows: the slide is first scanned sequentially and the captured images of the different fields stitched together to form a virtual slide map. The slide map is broken down into smaller portions which act as the models. The FOV acts as the query image. Localisation of the FOV is then preformed by finding the matching model which is essentially the same as associating a query object to a known model in model-based object recognition. Model-based recognition can be divided broadly into two stages; namely an *offline pre-processing stage* which involves the appropriate preprocessing of models and their storage in a suitable database; and the *online localisation stage*, which involves finding the best matching model to the query image.

2.1 Offline pre-processing stage

To generate the models, the virtual slide map first must be constructed. Images of adjacent fields were captured with some overlap and stitched to construct the map. The stitching method used is based on geometric hashing, which is also used for auto-positioning.

We define a tile as a region which is the same size as a single FOV and does not overlap adjacent regions (Fig. 2 for example shows a map with 12 tiles). The slide map was decomposed into models, each model having the size of 4 adjacent tiles and overlapping the adjacent model by 2 tiles (Fig. 2). This ensures that any FOV, which will be equal in size to one tile, selected during the *online localization stage* will be entirely contained in at least one model provided the orientation of the slide is unchanged. For a different slide

orientation, the FOV would still be largely contained in at least one model. In practice, this provides freedom to the operator to select any field on the slide for localization. Once the matching model to the FOV is determined, the transformation mapping between the two will reveal where exactly the FOV lies in that model.

The query image and all the model images originate from the same slide. Therefore, if Q is the query image and the predefined objects (the model set) are $(M_1, M_2, M_3, \dots, M_n)$, then it is assumed that there is a M_i from that set that represents Q . In microscopy, typical image variations include illumination changes and geometrical changes. Geometric changes are due to improper placement of the slide in the slide holder resulting in changes such as translation, rotation and scaling which are collectively known as similarity transformation. Owing to these image variations, M_i may be required to undergo a transformation to match Q . To ensure the recognition system is robust to image variations, each model was invariantly described using invariant coordinates in coordinate frames formed by its own feature points - this is known as geometric hashing.

Let $\{m_1, m_2, \dots, m_k\}$ 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 Fig. 3. 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 .

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:

$$m_j = \frac{m_1 + m_2}{2} + \alpha_j(m_2 - m_1) + \beta_j(m - m_1) \quad (1)$$

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 [18]. The coordinates (α_j, β_j) remain unchanged regardless of the linear transformation applied on model, M and hence the co-ordinates (α_j, β_j) are referred to as invariant co-ordinates under similarity transformations.

The entry $(M_i, m_1, m_2) - M_i$ represents the model number from which m_1 and m_2 originate from - was then stored in the hash table at indices $(\alpha_j, \beta_j); 3 \leq j \leq k$ where k is the number of feature points in the model image. This was done for every possible ordered pair of points (bases) in the model image. If the number of feature points in a model M is denoted as N_i ,

then the number of different bases in that model is equal to $\binom{N_i}{2}$ [13]. The entries in the hash table are an invariant representation of the model image. The entire process was performed for all the model images of the virtual slide map. 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, m_μ, m_ν) .

To generate an invariant description as explained above, feature points must be extracted from an image. Due to possible variations in the illumination conditions and orientation of the slides, the feature points extracted must be invariant to these changes. This was achieved using the medial axis transform [20] subsequent to image segmentation.

Image segmentation—Image segmentation was performed using a quadratic pixel classifier due to its high performance in segmentation of ZN sputum smear images as shown

by Khtulang et al.[7]. In ZN sputum smear images, acid-fast bacilli stain red against a blue background (Fig. 1). Knowing this a priori information in addition to a query pixel's value in the three channels of the RGB colour space, the quadratic classifier was used to classify the pixel either as a bacillus pixel or a non-bacillus pixel. The classifier first requires training using image pixels. Pixels of bacilli in the focal plane were labelled as +1 while a subset of background pixels were labelled as -1. Several training images were used. A quadratic mapping between the training pixels and their labels was then established. The discrimination between the two classes was drawn using the class mean and covariance matrices. The quadratic decision function (Equation 2) was then used to assign labels to query image pixels based on the inequality, classifying it as either a bacillus or non-bacillus pixel.

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

where X is an query pixel feature vector, M is the mean vector, Σ is the covariance matrix, and P_1 and P_2 are prior probabilities of the classes [7].

The resulting segmented images (which were binary images) composed numerous non-bacillus objects. A high number of objects would considerably increase the number of feature points per image resulting in a steep increase in the number of different bases,

$\left(\begin{matrix} N_i \\ 2 \end{matrix} \right)$, and hence in the number of entries per image. Model pre-processing would therefore take longer and would require a larger RAM space. To reduce the number of the segmented objects, we considered only TB bacillus objects. Since TB bacilli are long and thin, area and eccentricity filters [5], [7] were used to eliminate most of the non-bacillus objects. 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.

Extraction of feature points—Feature points were extracted via topology skeletons of the objects in a segmented image. A topology skeleton constitutes a homotopic and thin version of the shape of an object and hence acts as a shape descriptor [21]. We applied the medial axis transform, which is a shape representation technique in which foreground regions in a binary image are reduced to topology skeletons [21]. It largely preserves the extent and connectivity of the original regions while throwing away most of the original foreground pixels [22].

The resulting branch points of the topology skeletons of the objects 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 subsequent invariant description of the image. Fig. 4 summarizes the process for feature point extraction on a zoomed sub-image. The red dots represent the extracted feature points.

2.2 Online Localisation

The *online localisation stage* comprises three steps namely processing of the FOV image, indexing and voting and finally verification.

Processing the FOV image (query image)—To localise a FOV image on the virtual slide map, it was first segmented and k feature points extracted exactly as done for the models. It was then represented similarly to the model representation but using only a single arbitrary basis pair of feature points (q_1, q_2) . This is because (q_1, q_2) will already have a matching basis pair of points in the database since all possible basis pair combinations were considered during pre-processing of the models. The invariant co-ordinates (α_j, β_j) of every other feature point $q_j, 3 \leq j \leq k$, in the FOV image are then computed in the coordinate frame formed by (q_1, q_2) .

Indexing and voting—The voting step is 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 all the entries, which are in the form (M_j, m_μ, m_ν) , in them.

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 bins of the hash table. However, the ‘correct’ bins are in the neighbourhood of these ‘wrong’ bins [18]. In order to ensure the ‘correct’ bin was included in the voting, instead of extracting only the bin at index (α, β) , entries in the neighbouring bins were also considered for voting.

The process was repeated for all the coordinates (α_j, β_j) of every feature point, $q_j, 3 \leq j \leq k$. The best matching model to the query image is expected to receive a significant number of votes but will not necessarily receive the highest number of votes, due to the presence of noise and outliers in the query image. Therefore, all the entries, which are in the form (M_j, m_μ, m_ν) , that accumulated a significant number of votes were considered and verified. If v_{max} was the maximum vote accumulated, then only entries that received at least 40% of v_{max} votes were considered. These formed the candidate model-basis combinations (*CMBs*) that are possible matches to the query image Q . This candidate list can be referred to as *CL1*. A single model, M_j , can have several (m_μ, m_ν) bases in it that may be candidate matches but only one or none will be a best match to (q_1, q_2) of Q . Therefore all the *CMBs* in *CL1* need to be verified. We extend the scheme to further reduce the verification load by employing scale and orientation filters.

To match (Q, q_1, q_2) , a *CMB*, for example (M_5, m_6, m_9) , may have to undergo a similarity transformation, T , which is a combination of translation (t_x and t_y corresponding to the x and y translation respectively), rotation (angle θ) and isotropic scaling (scale factor, s). 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: $M = \{m_j / j = 1, 2, \dots, k\}$ where m_j are feature points in the matching model and are described with coordinates x and y and denoted as $m_j = (x_j, y_j)$. $Q = \{q_i / i = 1, 2, \dots, r\}$, where q_i are the corresponding feature points in the query images and are described with co-ordinates x and y and denoted as $q_i = (x_i, y_i)$. Then $Q = T(M)$ is essentially equivalent to $q_i = Tm_j$ resulting in:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = T \begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_j \\ y_j \end{pmatrix} \quad (3)$$

To solve for the parameters of the similarity transformation T , a minimum of two point-to-point correspondences need to be known [23]. For the given example, (q_1, q_2) and (m_6, m_9) are the two point-to-point correspondences (the only point-to-point correspondences known at this stage) which are used to estimate the parameters. These parameters are estimated for every *CMB*. *CMBs* that are highly unlikely to be matches to (Q, q_1, q_2) were filtered out by employing angle and scale filters. The thresholds of the filters 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 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. A scale factor range of 0.85 – 1.5 was used.

Rotational angle, θ : Manual slide placement can result in rotational angles between query image and model image of up to 12 degrees [11]. 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.

Therefore, a *CMB* was discarded if it's computed scale factor did not fall between 0.85 and 1.5 or the magnitude of its computed angle was greater than 30°. The remaining *CMBs* were further verified.

Verification—This is the final step in the recognition algorithm as it reveals the best matching model and the transformation relating it to the query image. For a given *CMB*, the estimated T is applied to the corresponding model. Every model point will correspond to the closest point, q_i' , in Q . That is:

$$q_i' = \arg \min_k (d(q_k, Tm_j)) \quad (4)$$

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 Tm_j needs to be checked. If the Euclidean distance between correspondences is below a threshold value i.e. $d(q_i', Tm_j) < t$, then those corresponding points were considered as inliers. Otherwise they were classified as outliers, which are inconsistent feature points in Q that do not have a corresponding point in the model. The threshold distance, t , was empirically determined and the value used was 10 pixels [24].

To speed up this process of finding the correspondences, Voronoi Tessellation [25] of the point patterns was employed. Constructing a Voronoi tessellation on feature points 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 Tm_j . The centre point, q_i , of that polygon will be the corresponding point to m_j .

The *CMB* that results in the highest number of inliers is expected to be the best matching model to Q . However, in practice, due to noise and errors introduced by the feature detector, some feature points (outliers) in the Q might be mistakenly reported and will wrongly match or not match any model point. These outliers severely disturb the point-to-point correspondences and therefore need to be identified and discarded to allow retrieval of the correct matching model and optimal estimation of the transformation [24].

We use a robust estimator, the RANSAC [26], to make the localization algorithm robust to outliers. For a given *CMB*, the set of point-to-point correspondences $q_i \leftrightarrow m_j$ that has been already determined acted as the putative set of correspondences which is essential for RANSAC. A sample of two point-to-point correspondences - two because that's the minimum number of correspondences required to compute a similarity transformation - was randomly selected from the set and an approximation $\hat{H}$ of the desired similarity transformation, T , was 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 $d(q_i', \hat{H}m_j) < t$ where $\hat{H}m_j$ is a transformed model feature point and q_i is the feature point in Q closest to $\hat{H}m_j$, and t is the threshold distance which was set to 10 pixels. This process is repeated several times with different samples and the $\hat{H}$ which provides the largest number of inliers was selected as the best estimate transformation, $\hat{H}_{best}$, between that *CMB* and the query image. The precision of the transformation was further improved by re-computing it using only the inliers corresponding to $\hat{H}_{best}$. This was done by the least squares fit [24]. The resulting transformation, T , is the optimal transformation linking that *CMB* to the query image.

The entire verification process was performed for each and every *CMB* to determine the optimal number of inliers and optimal transformation between the query image and that *CMB*. The *CMB* that shared the highest number of inliers with the query image (in case of a tie, the *CMB* having the minimal standard deviation of inliers is chosen) was declared the best matching model-basis combination (*BMMB*) to the query image and the corresponding T the optimal transformation relating the two. Therefore, the algorithm produces the best matching model and the registration parameters simultaneously. The registration parameters show where exactly the query image lies in the matching model which is four times as large.

Due to noise in the query image, it is possible that one of the basis points (q_1, q_2) that was selected will be reported by mistake. Consequently, it will not have a matching model point and hence the algorithm may fail to find a matching *CMB*. Furthermore possible inaccuracies (induced by noise) in the locations of the other feature points will have similar effects. When this occurred, another attempt was made by re-executing the algorithm with another arbitrary basis (q_i, q_j) from the query image. A maximum of 10 attempts were allowed to accommodate considerably noisy images. If the query image was not matched within this maximum allowable number of attempts, it was declared a miss.

In summary, the algorithm comprises the offline preprocessing stage and the online localisation stage. In the offline pre-processing stage, images of the different fields of a given slide are acquired after which they are stitched together to construct a virtual slide map. The map is then decomposed into models. Invariant descriptions of the models, obtained using the features points extracted by the medial axis transform subsequent to image segmentation, are then stored in a hash table. In the online localisation of a query image, feature points are extracted as done for the models and an invariant description of the query image is obtained using a single arbitrary basis pair of points. The resulting description is used in the indexing and voting stage to produce potential matching *CMBs*. These *CMBs* are then verified, with the help of Voronoi tessellation and RANSAC, to establish the best matching model and also the transformation relating that model and the query image.

3 EXPERIMENTS

3.1 Image acquisition

We used a conventional bright field microscope, Zeiss Axioskop 2 with a 40x objective lens at 0.75 numerical aperture for image acquisition. A rectangular section of the smear on a slide was marked and the slide placed in the slide holder so that it was aligned to the x -axis of the xy stage. Both stage movement and focusing were manually performed. The accompanying Axiovision 4.7 software allows real time display of the field-of-view which was captured at 1030×1300 pixels using the attached Axiocam HR digital colour camera. Each pixel measured $0.27 \mu\text{m} \times 0.27 \mu\text{m}$. Colour images with red, green and blue planes (RGB) were used in the experiments. Three slides, positive for TB, were used for this study – slide A, slide B and slide C. Slide A had relatively fewer bacilli per field. Images of slide A's fields were acquired sequentially with a 5–15% overlap between adjacent images and

were subsequently stitched manually using Photoshop 7.0 to construct its virtual slide map. Images of slide B and slide C on the other hand were automatically stitched to construct its virtual slide map and, to ensure good stitching results, images were acquired with a 30–50% overlap. Slides A and B were used to validate the auto-positioning method while slide C was used only to compare two auto-stitching methods.

All the image processing algorithms were developed on a 32-bit quad-core desktop computer with 2.67 GHz Intel processors and 2.96 GB of RAM using MATLAB R2007b, its image processing toolbox and the PRtools toolbox [27].

Construction of virtual slide map by auto-stitching—In [11] and [13], an automated XY stage was used to scan the slide and the images were simply joined together to form the virtual slide map. If the automated XY stage is not accurate, then simple joining of two adjacent images can result in repetition of overlapping features in the resulting image. In their case, since the query images were extracted from the virtual slide map itself, the repetition of the overlapping features would also exist in the query images and hence the matching process will be unaffected. Our microscope lacked an automated XY stage - it is impossible to guarantee manual capture of adjacent images without any overlap. Simple joining of adjacent images would result in a virtual slide with many repeated features between adjacent images. A **real** query image (captured at a different time to scanning) spanning two adjacent images of the virtual slide, would not contain these repeated features and hence the matching process would be impaired. Thus image stitching was adopted to ensure the constructed virtual slide map was an accurate replication of the actual slide. A small region of a smear resulted in hundreds of images prompting the use of auto-stitching to construct the virtual slide map.

Auto-stitching requires automatic identification of the overlap region between adjacent images. Two matching schemes, namely geometric hashing scheme (GHS) and the scale invariant feature transform (SIFT) were considered and compared for auto-stitching images [28]. GHS was considered because the developed object recognition algorithm for auto-positioning was based on the GHS scheme itself. The SIFT scheme was considered as it is a widely used matching scheme in photography [29] and also because Ma et al [30] showed that the Autostitch software, which uses the SIFT matching scheme, is suitable for stitching microscopy images.

The systematic acquisition of images enabled stitching to be performed in the sequence as shown in Fig. 5 where the black outline represents the rectangular region on the slide and the red arrows illustrate the direction of image stitching. I_{rk} is the last image of row k and the total number of rows of images = $R = rk + 1$.

Let the captured images be labeled 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, V_n , which was constructed by sequentially stitching the next image, I_{j+1} , to the current partial virtual slide map, V_j .

$$V_{j+1} = V_j + I_{j+1}, \quad 0 \leq j < n \quad (5)$$

The symbol '+' in this context refers to stitching of images. 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 formed the first partial virtual slide map V_1 .

Automatic construction of the virtual slide map requires automatic identification of the overlap region between V_j and I_{j+1} . However, for larger values of j , V_j would be very large

relative to the image I_{j+1} and therefore the matching process would be prone to errors. The matching process would be more accurate and faster if a smaller portion of V_j were considered in the matching process.

Image stitching using a small portion of V_j —Since prior knowledge was available as to which image needs to be stitched to which image, the portion-of-interest, POI_j , of V_j that shares an overlap region with the image, I_{j+1} , to be stitched was known and could be extracted from V_j . Using the sequence shown in Fig. 5, 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 Fig. 6. 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.

Prior to the scanning process, the camera was well aligned to the xy stage i.e. objects moved parallel to the x and y -axes of the field-of-view. This guaranteed the correct movement of the stage, and 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 was 1.

Therefore, the orientation and the scaling could 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} consisted only of translation (parameters t_x and t_y).

SIFT: SIFT keypoints were extracted directly from POI_j and stored in a database. Similarly, SIFT keypoints were extracted from I_{j+1} followed by matching each keypoint independently to the database of keypoints. The matching keypoints formed a putative set of correspondences between POI_j and I_{j+1} . The random consensus sample (RANSAC, [26]) was executed to eliminate highly mismatched keypoints (outliers) and the resulting inliers obtained were used to re-compute a better estimation of the translation parameters of the transformation in the least-squares sense. Using this transformation mapping, image I_{j+1} was stitched by replacing the common overlapping region between it and POI_j by only that of Image I_{j+1} . The process was repeated until the entire virtual slide map, V_n , was constructed.

GHS: Using geometric hashing, at any given time, POI_j acts as the one and only model while I_{j+1} acts as the query image. Since multiple models are not involved, all we are doing is finding which model basis, (m_u, m_v) , in POI_j best matches the chosen query basis (q_1, q_2) in I_{j+1} . POI_j was processed and stored in the hash table. To execute the registration process, I_{j+1} was segmented, filtered, and feature points extracted similarly to POI_j . An arbitrary basis, (q_1, q_2) , 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. For configuration 1, both the chosen points q_1 and q_2 were located in the left half of I_{j+1} since only the left half of I_{j+1} overlaps with POI_j . Similarly, for configuration 2, both the chosen points were located in the upper half of I_{j+1} and for configuration 3, one of the chosen points was located in the left half while one in the upper half of I_{j+1} .

The best matching model basis (m_u, m_v) was found as explained in Section 2.2. Using the resulting optimal transformation, T , image I_{j+1} was stitched to POI_j , which is part of V_j . The common overlapping region between I_{j+1} and POI_j was replaced by only that of Image I_{j+1} . The process was repeated until the entire virtual slide map, V_n , was constructed.

Comparison of the auto-stitching schemes—A rectangular region measuring 1.6×1.1 mm on slide C was scanned which produced 62 images. Two virtual slide maps were constructed, one with SIFT, named SIFT VS, and the other using GHS, named GHS VS. The resulting SIFT VS measured 4049×5976 pixels while the GHS VS measured 4044×5979 pixels. The almost identical sizes indicate very little discrepancies exist between the two VSs. Furthermore, there were no detectable differences at and near the visible seam lines upon visual comparison of the two VSs. Fig. 7 shows the two VS's.

If the stitching quality of both the methods is same then corresponding triangles - defined by 3 pairs of corresponding points - in the two VSs are expected to be congruent [30]. A triplet of points was selected in the SIFT VS and the corresponding points in the GHS VS 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 have the same size and shape and hence they also have the same perimeter. Therefore, perimeters of the corresponding triangles were also compared.

A set of 1140 corresponding triangles in the SIFT VS and the GHS VS were used for this analysis. The triangles varied in size and covered various regions in the two VSs. Table 1 shows the results obtained for 5 randomly chosen pairs of corresponding triangles.

As seen in Table 1, the discrepancies between the ratios of the triangles from the two VSs 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.

In conclusion, the GHS and SIFT auto-stitching schemes showed negligible differences, indicating high similarity of stitching quality between the two.

Slide B's virtual slide map was constructed using the GHS scheme; the GHS auto-stitching scheme was selected over the SIFT auto-stitching scheme in order to maintain consistency since the object recognition scheme developed for auto-positioning was based on GHS scheme.

Fig. 8 shows the virtual slide maps of slide A (manually stitched using Photoshop 7.0) and slide B. 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.

Table 2 summarises different properties of the two virtual slide maps. The average number of feature points per model image of slide A was 90 while that from slide B was 140. The large difference was mainly because Slide A had relatively fewer bacilli per field.

3.2 Object recognition 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. Localising the query image on the virtual slide map is equivalent to finding the model in the database that best matches the query image. Hence the performance of the localisation task was evaluated as follows: If the algorithm reported the correct matching model, it was declared a true positive; if it reported an incorrect model, it was declared a false positive and if it did not find a matching model in 10 attempts, it was declared a 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 could also be identified without difficulty.

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

For the evaluation of the image registration, only true positives were considered. We used the mean square error (ϵ_{avg}) and root mean square ($\sqrt{\epsilon_{avg}}$) error which are commonly used to evaluate registration error between two images [31]. Cheng [23] refers to this as the average pairwise error and used it to determine the matching error between two point patterns. The algorithm produces the *BMMB*, the optimal transformation, T , and also for that T , the point-to-point correspondences, $q_i \leftrightarrow m_j$ between the *BMMB* and the query image. The mean square error was computed as:

$$\epsilon_{avg} = \frac{1}{n} \sum_{j=1}^n \|T(m_j) - q_j\|^2 \quad (6)$$

where n is the number of inliers between *BMMB* and Q . The mean square error incorporates the errors in all the registration parameters. The square root of ϵ_{avg} gives the root mean square error which is equivalent to the fiducial registration error [32].

3.3 Algorithm performance

The algorithm was tested on two sets of query images, namely *Query set 1* and *Query set 2*

Algorithm performance with Query set 1—*Query set 1* comprised images extracted directly from the virtual slide maps themselves. To form a query image, Q , a region measuring 1030×1300 pixels (i.e. the same size as a tile image) was randomly selected from the virtual slide map. For each slide 250 query images were extracted. Since the query image emerges from the virtual slide itself, it is completely noise-free relative to the matching model and there will be a model in the database that will exactly match Q . Therefore, the rotational angle and scale factor between these query images and their matching models could be expected to be 0° and 1 respectively. Tables 3 – 5 summarise the algorithm's performance with *Query set 1*.

In all cases, the scale parameter of the optimal transformation, T , relating the query image to the matching model on average was 1.000 with standard deviation < 0.001 .

Query set 1 was further used to assess the discriminative power of the algorithm. In the verification stage, all *CMBs* are compared to the query image to find the *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 a significantly higher number of votes than many other *CMBs* in *CL1*, 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. The discriminative power can be defined in terms of the position of the *BMMB* in the sorted *CL1*. The higher the *BMMB* in this list, the greater the discriminative power of the algorithm [13]. We used *Query set 1* to assess the discriminative power of the algorithm since these images were noise-free relative to the models. We studied the variation of the algorithm's discriminative power with the number of feature points in query images. The algorithm was highly discriminative (the best matching *BMMB* was near the top of the sorted candidate list (*CL1*)) when the query image contained about 23 feature points (Fig. 9). An additional feature point further improved the

discriminative power since each additional feature point provided supplemental evidence for the presence of the correct model.

Discussion on Query set 1 results—As seen in Table 3 the hit rate achieved for *Query set 1* was above 90% for both slide A and slide B. 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, this was not achieved because the constructed maps contained areas difficult for navigation, which included areas having no or very few bacilli and hence containing few feature points. The algorithm produces the *BMMB* provided it received a significant number of votes. With few feature points involved, the voting results would be poor (the *BMMB* rank will not be near the top in the sorted *CLI* as seen in Fig. 9) and this would mean the *BMMB* is highly likely to be eliminated before the verification stage (votes received by *BMMB* $< 0.4v_{max}$).

Algorithm performance with Query set 2—Unlike the previous papers [11],[13], which only tested with synthetic query images extracted from the virtual slide map itself, we also tested the algorithm robustness to image variations using distorted real images that were obtained at a different time to the scanning process. We called this set of images *Query set 2*. This set of query images comprised several sub-sets of images where each sub-set was obtained at a different time and different orientation, θ_c , 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 (1030×1300 pixels) lying within the rectangular mark on the slide were then captured at 40x magnification. About 150 images were captured for each sub-set.

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 set 2* were considerably distorted relative to the model images.

Since a sub-set of query images was formed by obtaining images at a particular orientation, say θ_c , of the slide, the registration angle parameter for any image in that sub-set is expected to be θ_c . Moreover, 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 sub-set of query images. This is because the models are four times larger in size than any query image (size of one tile) and hence the position of the query images in their respective matching models will differ.

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 *cpselect* tool, 10–15 corresponding

points between a query image and its matching model were manually selected. Although, two corresponding pairs of points are sufficient to compute a similarity transformation, 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. This process was done for all the 5 pairs (query image – model image) and an average orientation angle was computed. This orientation angle acted as the ground truth angle for the given subset. The ground truth orientation angle of each of the subsets was obtained in this manner. We named the sub-set of a given orientation based on the slide they were obtained from and the ground truth orientation angle computed with the *cpselect* tool. For example, slide B - 10.58 represented the sub-set of images obtained using slide B and at orientation 10.58°. We compare the angle parameter reported by our algorithm to the ground truth angle of the respective sub-set.

For the different sub-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. To illustrate this visually, an example image from each sub-set and the corresponding matching model are shown in Fig. 10 and Fig. 11. Fig. 10 shows one image from every sub-set of images considered for slide B and the matching model in slide B. Similarly, Fig. 11 shows one image from every sub-set of images considered for slide A and the matching model of slide A. An image from *Query set 1* is also included for each of the slides for comparison.

The performance of the algorithm for *Query set 2* is shown in Fig. 12. The performance on the individual slides is illustrated in Fig. 13 and Fig. 14.

The false positive rate was 0 for all slide B orientations. As seen in Fig. 15 the average registration angle parameter computed by the algorithm was comparable to that established manually using the *cpselect* tool in MATLAB (a maximum error of 0.05° only). 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. Errors in all the registration parameters are included within the mean square error. Fig. 16 illustrates the variation of the average mean square error with slide orientation.

3.4 Discussion

Representation of a model and hash table filling took on average 2 minutes; this is an offline stage and therefore time is not critical. The time taken for the online stage of localising a query image ranged from 27s to 200s. This time variation is due to the varying number of feature points among the query images. Although it may be argued that a human can bring a desired field on the slide to the field-of-view of the microscopes faster than the algorithm, the algorithm is likely to out perform a human when multiple regions-of-interest on the same slide need reviewing 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 and can be used to easily and quickly bring any desired fields to the FOV of the microscope. A human would have to manually search each time a different field needs to be brought to the FOV of the microscope. Furthermore, re-writing the algorithm on a faster platform such as C and/or using parallel computing techniques would considerably reduce the time taken by the 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 (illustrated in Fig. 9), resulting in the matching model being eliminated before the verification process in all 10 attempts.

The miss rate in slide A (Fig. 13) was higher than that of slide B (Fig. 14) because slide A had a relatively higher number of fields with no or very few bacilli.

Despite a wide range of real image distortions present in *Query set 2*, the hit rate (HR) for all the sub-sets of images considered did not drop below 88% (Fig. 12). For some sub-sets of images, a HR as high as 98% was achieved. 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.

The average mean square error computed by the algorithm for a given sub-set of images did not exceed 14 pixels^2 ($1.02 \mu\text{m}^2$); 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. Since the conventional method for bacillus detection is visual, 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.

The results using slide B, which was automatically stitched using geometric hashing, were comparable to those of slide A which was manually stitched. This indicates the suitability of the geometric hashing method for automatically stitching a large number of microscopy TB images to construct large virtual slide maps for auto-positioning.

The algorithm allows the current field-of-view to act as a query image and be localised after which it can be used as a reference point to bring desired fields-of-interest to the field of view of the microscope, therefore achieving auto-positioning. In practice, in the event of a miss by the algorithm, i.e. failure to localise the current FOV in 10 attempts, the algorithm can be adapted to direct the microscope to another arbitrary field. The image of the new field would act as a new query image which is fed to the algorithm and the localisation re-performed.

4 CONCLUSION

We have presented a method for microscope auto-positioning in bright-field TB microscopy, which is based on virtual slide maps in combination with geometric hashing. The method is suitable for auto-stitching multiple images to construct virtual slide maps and also for localisation of regions-of-interest. The method was tested on severely distorted real images and the results show that it 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 showed high tolerance to illumination changes and robustness to noise.

The method developed may be used 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 at different settings of a microscope or across microscopes.

Acknowledgments

This work was supported by the National Institutes of Health/National Institute of Allergy and Infectious Diseases (NIH/NIAID) under Grant R21 AI067659-01A2.

References

1. Steingart KR, Henry M, Ng V, et al. Fluorescence versus conventional sputum smear microscopy for tuberculosis: A systematic review. *The Lancet Infectious Diseases*. 2006; 6(9):570–581. [PubMed: 16931408]
2. Hänscheid T. The future looks bright: Low-cost fluorescent microscopes for detection of mycobacterium tuberculosis and coccidia. *Transactions of the Royal Society of Tropical Medicine and Hygiene*. 2008; 102(6):520–521. [PubMed: 18405928]
3. Osibote O, Dendere R, Krishnan S, Douglas T. Automated focusing in bright-field microscopy for tuberculosis detection. *Journal of Microscopy*. 2010; 240(2):155–163. [PubMed: 20946382]
4. Veropoulos K, Learmonth G, Campbell C, Knight B, Simpson J. Automated identification of tubercle bacilli in sputum. A preliminary investigation. *Analytical & Quantitative Cytology & Histology*. 1999; 21(4):277–282. [PubMed: 10560504]
5. Forero MG, Cristobal G, Desco M. Automatic identification of mycobacterium tuberculosis by gaussian mixture models. *Journal of Microscopy*. 2006; 223(2):120–132. [PubMed: 16911072]
6. Sadaphal P, Rao J, Comstock G, Beg M. Image processing techniques for identifying mycobacterium tuberculosis in ziehl-neelsen stains. *The International Journal of Tuberculosis and Lung Disease*. 2008; 12(5):579–582. [PubMed: 18419897]
7. Khutlang R, Krishnan S, Dendere R, et al. Classification of mycobacterium tuberculosis in images of ZN-stained sputum smears. *IEEE Transactions on Information Technology in Biomedicine*. 2010; 14(4):949–957. [PubMed: 19726269]
8. Khutlang R, Krishnan S, Whitelaw A, Douglas TS. Automated detection of tuberculosis in ziehl-neelsen-stained sputum smears using two one-class classifiers. *Journal of Microscopy*. 2010; 237(1):96–102. [PubMed: 20055923]
9. Electron Microscopy Science. <http://www.emsdiasum.com>
10. Microlab, CellFinder Microscope Slides. <http://www.antenna.nl/microlab/index-uk.html>
11. Begelman G, Lifshits M, Rivlin E. Visual positioning of previously defined ROIs on microscopic slides. *IEEE Transactions on Information Technology in Biomedicine*. 2006; 10(1):42–50. [PubMed: 16445248]
12. Dee FR, Lehman JM, Consoer D, Leaven T, Cohen MB. Implementation of virtual microscope slides in the annual pathobiology of cancer workshop laboratory. *Human Pathology*. 2003; 34(5):430–436. [PubMed: 12792915]
13. Lifshits M, Goldenberg R, Rivlin E, Rudzsky M, Adel M. Image-based wafer navigation. *IEEE Transactions on Semiconductor Manufacturing*. 2004; 17(3):432–443.
14. Doerrer, R. System and method for re-locating an object in a sample on a slide with a microscope imaging device. US. 20070076983. 2007.
15. Costantino S, Heinze KG, Martínez OE, De Koninck P, Wiseman PW. Two-photon fluorescent microlithography for live-cell imaging. *Microscopy Research and Technique*. 2005; 68(5):272–276. [PubMed: 16315236]
16. Lamdan, Y.; Wolfson, HJ. Geometric hashing: A general and efficient model-based recognition scheme. *Proceedings of the Second International Conference on Computer Vision*; New York. 1988. p. 238–249.
17. Lamdan, Y.; Wolfson, H. On the error analysis of geometric hashing. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; New York. 1991. p. 22–27.
18. Wolfson HJ, Rigoutsos I. Geometric hashing: An overview. *IEEE Computational Science and Engineering*. 1997; 4(4):10–21.
19. Mehrotra H, Majhi B, Gupta P. Robust iris indexing scheme using geometric hashing of SIFT keypoints. *Journal of Network and Computer Applications*. 2010; 33(3):300–313.
20. Blum H. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*. 1967; 19(5):362–380.
21. Mokhtarian F, Mackworth AK. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 14:789–805.

22. Loncaric S. A survey of shape analysis techniques. *Pattern Recognition*. 1998; 31(8):983–1001.
23. Cheng FH. Point pattern matching algorithm invariant to geometrical transformation and distortion. *Pattern Recognition Letters*. 1996; 17(14):1429–1435.
24. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press; New York: 2003.
25. De Berg, M.; Cheong, O.; Van Kreveld, M.; Overmars, M. *Computational Geometry: Algorithms and Applications*. Springer-Verlag; New York: 2008.
26. Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 1981; 24(6):381–395.
27. Duin R, Juszczak P, Paclik P, Pekalska E, De Ridder D, Tax D. *Prtools4: a MATLAB toolbox for pattern recognition*. 2004
28. Patel, B.; Douglas, T. Creating a Virtual Slide Map of Sputum Smears by Auto-Stitching. *Proceedings of the IEEE Engineering in Medicine and Biology Society*; Boston. September 2011;
29. Roth, PM.; Winter, M. 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
30. Ma B, Zimmermann T, Rohde M, Winkelbach S, He F, Lindenmaier W, Dittmar KEJ. Use of autostitch for automatic stitching of microscope images. *Micron*. 2007; 38(5):492–499. [PubMed: 17045805]
31. Zitova B, Flusser J. Image registration methods: A survey. *Image and Vision Computing*. 2003; 21(11):977–1000.
32. Fitzpatrick JM, West JB. The distribution of target registration error in rigid-body point-based registration. *IEEE Transactions on Medical Imaging*. 2002; 20(9):917–927. [PubMed: 11585208]
33. Lowe DG. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 2004; 60(2):91–110.

Fig. 1.
Image of ZN stained sputum smear

Fig. 2.
Decomposition of a virtual slide map made up of 12 tiles

Fig. 3.
Image representation using feature points (dots) in the spatial domain.

Fig. 4.
Extraction of feature points.

Fig. 5.
Image stitching sequence.

Fig. 6.
Configurations of the V_j .

Fig. 7.
Virtual slide maps using (a) SIFT and (b) GHS

(a)

(b)

Fig. 8.
Virtual slide maps of (a) slide A (b) slide B.

Fig. 9.
Discriminative power variation with number of feature points

Fig. 10.
Image variations among the different sets of images of Slide B.

Fig. 11.
Image variations among the different sets of images of Slide A.

Fig. 12.
Algorithm performance at different slide orientations

Fig. 13.
Algorithm performance at different slide A orientations

Fig. 14.
Algorithm performance at different slide B orientations

Fig. 15.

Error between the average angle reported by algorithm and that obtained using the *cpselect* tool

Fig. 16.
Variation of average mean square error with image orientation

Table 1

Quantitative comparison of SIFT and GHS auto-stitch.

Triangle	Ratio of sides in triangle	
	In SIFT VS	In GHS VS
1	1 : 2.799 : 1.799	1 : 2.800 : 1.801
2	1 : 2.119 : 1.193	1 : 2.119 : 1.193
3	1 : 4.399 : 3.791	1 : 4.404 : 3.798
4	6.667 : 6.985 : 1	6.665 : 6.984 : 1
5	2.050 : 1.395 : 1	2.049 : 1.394 : 1

Table 2

Properties of the virtual slide maps of slides A and B.

Slide	Size of smear considered on the slide (mm ²)	Overlap between adjacent images	Total number of images	Total number of models generated
A	4 × 2	5 – 15 %	127	77
B	4.5 × 2	30 – 50 %	307	84

Table 3

Algorithm performance with *Query set 1*.

Slide	Number Of images tested	Misses		False matches		Hits	
		Number	Miss rate	Number	FP rate	Number	Hit rate
A	250	5	0.02	16	0.06	229	0.92
B	250	3	0.01	2	0.01	245	0.98

Table 4

Average angle reported for *Query set 1*.

Slide	Slide Orientation (°)	<u>Angle Reported by algorithm (°)</u>	
		Avg.	Std dev.
A	0	0.001	0.060
B	0	0.003	0.053

Table 5Average mean square error for *Query set 1*.

Slide	Ms error (pixels ²)		Rms error (pixels)	
	Avg.	Std dev.	Avg.	Std dev.
A	5.99	2.98	2.45	0.68
B	7.84	3.03	2.75	0.55