Classification of malaria infected blood cells using a mixture of Gaussians and morphology

Oliver Partida

Supervisor:

Dr. Jose Mira Mira, U.N.E.D

October 3, 2006

Abstract

The diagnosis of malaria is undertaken by counting the number of infected cells in the blood stream. The counting is done manually by an expert using a microscope. This task is usually very time-consuming and accuracy in the diagnosis requires certain expertise and conditions. An accurate automatic system for diagnosis of malaria is needed and would be a valuable tool reducing the time needed for diagnosis and helping in areas where human expertise is demanded. In this work an attempt has been made to automatically count the total number of blood cells and the number of cells infected with the malaria parasite from an image of a patient's blood on a microscope slide. Images are pre-processed in order to reduce the variability due to diverse factors like lighting conditions, experience of the laboratorian or the quality of the microscope. In this work a method to reduce this variability is devised helping to improve the overall performance. A mixture of Gaussians model is used to segment the images and a granulometry analysis is carried out to obtain the average size of cells. Finally parasites and non-infected cells are counted.

Contents

List of Figures

Chapter 1

1.1 Introduction

Approximately 30 million people worldwide are affected by malaria and between 1 and 1.5 million people die from it every year. Malaria is caused by protozoan parasites of the genus Plasmodium. Four species of plasmodium can produce the disease in its various forms:

- Plasmodium falciparum
- Plasmodium vivax
- Plasmodium ovale
- Plasmodium malaria

A mosquito infected with malaria carries Sporozoites, when the mosquito bites a person these Sporozoites enter the blood stream and reach the liver, there they divide repeatedly and new spores called Merozoites infect the red blood cells. Within the red blood cell, asexual division starts and the parasites develop

through the stages of rings, trophozoites, early schizonts and mature schizonts. Mature schizonts consist of thousands of Merozites which are released and infect new red blood cells. A small proportion of the Merozoites in the red blood cells undergo transformation into female and male Gametocytes. The Gametocyte is the form that infects the mosquito when it bites and the cycle begins again. Microscopic examination of a drop of patient's blood is the main method for laboratory confirmation of malaria. Prior to examination, the specimen is stained (most often with Giemsa stain) to give the parasites a distinctive appearance. Experts look for parasites in the red blood cells in all their different stages. Figures 1.1, 1.2 and 1.3 show the shapes of the parasites during their 3 main stages. Unfortunately this method relies on the expertise of the laboratorian and on the quality of the equipment to obtain good results. In some areas, experts are in demand, microscopists are undertrained and overworked, equipment is unreliable and a delay in results may mean incorrect treatment.

The aim of this system is to automatically count the number of red blood cells and detect the parasites using a scan of a color photograph of stained malarial blood from a microscope. In developing an automatic system for counting malaria infected blood cells from images, four main problems have to be solved. First of all, the images present high variability and noise. This is due to different factors that may affect the quality of the images obtained and therefore a preprocessing step is necessary. The second problem is the segmentation of the image. In this step, basically three classes need to be recognized: background,

Figure 1.1: Trophozite

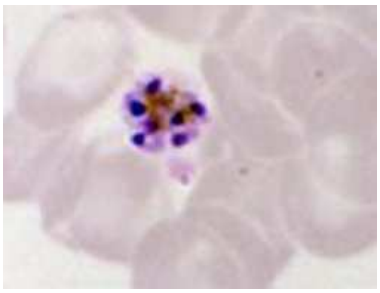

Figure 1.2: Schizont

Figure 1.3: Gametocyte

uninfected cells and parasites. These classes are not always easy to differentiate specially uninfected areas of the cell and parasites. The third problem to be tackled is the counting of cells. The difficulty of this step depends directly on the success of the segmentation. If good segmentation of the image is achieved, the counting of the cells and parasites should be easy. The main problem arises when cells overlap making the task difficult even for human experts. Finally, parasites have to be classified. We find in the literature different approaches aimed at solving each of these problems. In [RDKJ01] a combination of morphological operators and regional extrema is used. Bright objects are identified using regional maxima and morphology is used to remove the noise and segment the image. Classification of parasites is made using color similarity: the color

histograms of the parasites present in the image are compared to the color histograms of parasite samples. The type of the parasite is determined according to the parasite sample which it is most similar to.

In [TDK05b], images are first normalized to reduce variability due to different lighting conditions. Segmentation is carried out using a probabilistic model of the stained/non-stained pixels and morphology. In the following section each of these systems is briefly described.

1.2 Literature review

1.2.1 A morphological approach to malaria blood analysis

In [RDKJ01], morphological operators are the main tool used to identify infected red blood cells and classify the different parasites. The process is divided into four main steps: preprocessing, detection of parasites and white cells, cell segmentation and classification of parasites. The segmentation is based on prior knowledge of the problem: parasites and white cells will appear brighter in the images because the Giemsa staining solution highlights the DNA present in these objects and also white cells are much bigger than red blood cells and do not contain parasites. A first segmentation is then performed using a threshold that discards objects not bright enough and detects potential parasites and white cells. This first segmentation produces a marker image of parasites and white blood cells. The hue-saturation-value color space is analyzed. As we saw before, images contain noise, which is smoothed by applying a median filtering

using a 5 X 5 window on the hue and saturation components and small objects are removed from the image by applying a morphological area closing. To automate the selection of the threshold regional maxima and granulometry are used. Regional extrema are features of greyscale images used to mark objects: maxima for bright objects and minima for dark objects. A regional maxima is a set of connected pixels with the same grey level from which it is impossible to reach a point with higher level without descending. The connectivity of the pixels is defined by a structuring element. Regional maxima are found in both the H and S images obtaining the images MH and MS. Regional maxima is then the intersection of both images.

$$MHS = MH \cap MS$$

Granulometry is used in order to find the size distribution of the cells, the greatest size estimated from this distribution is used as the radius r of the structuring element that defines the connectivity. The value of the thresholds used μH μS to detect bright objects is the average value of the pixels marked as regional maxima according to the connectivity defined by the structuring element with radius r:

$$\mu H = \frac{1}{sum(MHS)} \sum_{p} H(p),$$

and

$$\mu S = \frac{1}{sum(MHS)} \sum_{p} S(p),$$

where H(p) and S(p) are the grey levels of the pixel p in the H and S images. The image identifying all the parasites and white blood cells THS is then obtained by thresholding H and S and computing the intersection:

$$THS = TH \cap TS$$

where TH and TS are the thresholded images by means of μH and μS. Bright objects identified in the previous step could be white blood cells or parasites. The size of cells found using granulometry is used to detect white blood cells and remove them from the image. The resulting image contains red blood cells and background. A second segmentation step is performed in order to isolate red blood cells. First the gradient of the image gives a rough localization of the red cell contours and then a watershed segmentation is applied using the gradient image as a marker. This gives a partial segmentation with some compound cells that have to be divided into their components. In order to identify composite cells a parameter measuring the roundness of the cell is computed. Composite cells are then separated into their components by applying a morphological opening with structuring element of size the smallest size of red blood cells, and computing the gradient of the resulting image to localize the red blood cell contours.

Classification of the parasites is achieved by two different methods:

- Morphology. Shape of the different kind of parasites is described using morphology. A skeleton of the parasites is constructed and used to classify them. Skeletonisation is a morphological operation that simplifies

the shapes of objects by reducing them to thin lines that summarize the information of the original object while preserving its topology.

- Color similarity. Samples of parasites are taken and used as prototypes to retrieve similar objects from the image. The samples are described using color histograms and histograms of objects from the image are compared to the prototypes and classified accordingly. In this way, given a object O, its chromatic content is expressed through a set of histograms $h_i(O), i = 1, ..., 4$, the ith histogram represents the chromatic content of the ith region of the object O. At the query time, the histograms of the query object Q, $h_i(Q), i = 1, ..., 4$ are matched against the histograms $h_i(O)$ of each object O present in the image. To evaluate the match, the histograms intersection is used:

$$d(h_i(Q), h_i(O)) = 1 - \frac{1}{2T} \sum_{j=1}^{m} \left| h_i^j(Q) - h_i^j(O) \right|$$

where T = $\sum_{j=1}^{m} h_i^j(O)$ and m = 256 is the number of bins.

The similarity is then the average of the four regions:

$$\frac{1}{4} \sum_{i=1}^{4} d(h_i(Q), h_i(O)).$$

1.2.2 A combination of morphology and probability to malaria blood analysis

In [TDK05b] a probabilistic approach to parasite detection is attempted. In order to detect the parasites a conditional probability density function of stained/non-stained pixels is estimated. The probability density is used to classify the pixels

of new images as stained/non-stained. The system performs three steps: color normalization, stained/non-stained color classification and cell identification.

A color normalization is applied to the RGB images in order to decrease the effect of different light source characteristics. Images taken under an unknown illuminant source $\mathbf{p^u}$ are mapped to a canonical known illuminant response $\mathbf{p^c}$ by means of a diagonal matrix M, $\mathbf{p^u} = \mathbf{M}\mathbf{p^c}$. The color constancy algorithm chosen is grey world and the scale factor is $\frac{p^c}{p^u}$. The values m_{11}, m_{22}, m_{33} of the diagonal matrix $\mathbf{M}$ are therefore obtained as follows:

$$m_{11} = \frac{nu \sum_{Ic} I_r^c}{nc \sum_{Iu} I_r^u} \qquad m_{22} = \frac{nu \sum_{Ic} I_g^c}{nc \sum_{Iu} I_g^u} \qquad m_{33} = \frac{nu \sum_{Ic} I_b^c}{nc \sum_{Iu} I_b^u}$$

where I is the image, nu is the number of pixels of the image taken under unknown illuminant source, nc is the number of pixels of the image taken under known illuminant conditions, subscripts(r,g,b) represent the RGB channels of the image and subscripts(c,u) stand for canonical and unknown respectively. Pixels are assigned to one of two classes w1, w2: stained and non-stained:

$$\mathbf{rgb} \in w1 \qquad if \qquad \frac{p(\mathbf{rgb} \mid w1)}{p(\mathbf{rgb} \mid w2)} \geq \theta.$$

θ is formulated with application dependent costs C_{ij} and a priori probability of classes P(w1),P(w2):

$$\theta = \frac{(C_{12} - C_{22})P(w1)}{(C_{21} - C_{11})P(w2)}.$$

The probability distribution of the pixels belonging to each of the classes is modeled using histograms. Images are labeled one by one and the histograms for each class constructed. In RGB space each channel is quantized to 256

levels which would require 256^3 separate bins. In order to avoid this complexity, different quantization levels are used 16, 32, 64, 128 and ROC curves used for model selection. Only parasites contained in red blood cells are important. After detecting stained pixels a segmentation of the whole image *Seg* is obtained using [TDK05a]. White blood cells are bigger than red blood cells therefore a stained connected component is classified as a parasite if it is contained in a red blood cell in *Seg*.

1.3 Contributions

In this work, a mixture of Gaussians, trained using a variation of the Expectation-Maximization algorithm, is used to segment images of cells infected with malaria. A method to reduce the variability of the data is devised. Color of the different classes important for segmentation are made more uniform among the images and consequently segmentation performance is improved.

Chapter 2

Classification of malaria infected blood cells using a mixture of Gaussians and morphology

2.1 Background

As we can see the problem of parasite detection and cell counting is a question of learning what a parasite and a cell are and then using that knowledge to analyze new images. Also, good preprocessing of the images is needed in order

to reduce the complexity of the problem. This complexity arises from the high variability that the images might present which makes recognition a very difficult task. There are two main techniques used in machine learning to learn from data: supervised learning and unsupervised learning. In the first one we tell the learning system what class each object belongs to and the system builds an internal representation that allows it to identify and categorize new objects. In unsupervised learning the learning system is not told the classes of the objects and the system tries to find different classes hidden in the data based on a similarity function. The system that constructs a probability distribution of the stained/non-stained pixels described previously is an example of supervised learning. The expert labels the pixels as belonging to one of two different classes and the system learns a function that assigns objects to one of the classes. In that system the function is a probability distribution and new objects, in this case pixels, are assigned to the most probable class. The problem with this approach is that a comprehensive set of examples is needed in order to build a reasonably accurate probability distribution. Quantization levels are used but this might cause accuracy reduction. The problem is not easily solved by using supervised learning methods due to the difficulty of coding each of the different objects that need to be segmented in the image. Using supervised learning techniques would require first a codification of the classes required to be recognized to train our models. The problem arises when we want to classify a new image because we do not know where the important objects are in the image.

To overcome these difficulties, in our work, we attempt an unsupervised learning approach to image segmentation. We assume that pixels belonging to the different classes have distinctive characteristics and attempt to capture these dissimilarities using a mixture of Gaussians model.

2.1.1 Unsupervised learning

Unsupervised learning is a sub-field of machine learning which is a field of research devoted to the formal study of learning systems. The main goal of a learning system is to build a representation of input data that can be used to solve various tasks like decision making and prediction. In the case of unsupervised learning, the input data is unlabeled in the sense that the system does not know beforehand that a particular description belongs to a certain class. Unsupervised learning finds patterns in the data beyond what could be considered unstructured noise. Unsupervised learning, typically, treats input objects as a set of random variables and builds a joint density model for the data set. Sometimes the patterns hidden in the data are easily modeled using simple functions but sometimes in order to describe the pattern we need more complex and powerful functions that are capable of explaining most of the data. Mixture of Gaussians models are very useful models because they combine a simple mathematical representation, with parameters that are simple to learn, with the ability to represent almost any probabilistic model with any complexity.

Mixture of Gaussians and EM

Mixture of Gaussians models allow us to model almost any non-Gaussian probability distributions using the well-known Gaussian form. The probability of a data point y under a mixture of Gaussians distribution with covariance matrices Σ, means matrix μ and prior probability of each Gaussian π is a weighted sum of Gaussian densities:

$$p(y \mid \mu, \Sigma, \pi) = \sum_{i=1}^{k} \pi_i p(y \mid \mu_i, \Sigma_i), \tag{2.1}$$

where

$$p(y \mid \mu_i, \Sigma_i) = \mid 2\pi\Sigma_i \mid^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y-\mu_i)^\top \Sigma_i^{-1}(y-\mu_i)\right).$$

The **EM** algorithm is an algorithm for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. It can be used for finding the parameters of the mixture model where the latent variable indicates which component of the mixture each data point came from. Given a dataset y = ($y^{(1)}$,...,$y^{(n)}$) the likelihood of Θ given y is

$$p(y|\Theta) = \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i p(y^{(c)} \mid \mu_i, \Sigma_i) \tag{2.2}$$

$$= \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i \mid 2\pi\Sigma_i \mid^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y^{(c)}-\mu_i)^\top \Sigma_i^{-1}(y^{(c)}-\mu_i)\right)$$

In some applications it is possible to make the maximization of (2) easier if our data is considered to have missing values. In the case of mixture of Gaussians, these values could indicate which Gaussian each data point came from. EM

alternates between 2 steps called the Expectation and Maximization steps respectively. In the E step an optimal lower bound of the likelihood is computed, this optimal lower bound touches the objective function at the current estimate of Θ. In the M step this lower bound is maximized. These 2 steps guarantee convergence to a maximum of the likelihood function. Any probability distribution q(x) over the hidden variables or missing values can be used to find a lower bound to the likelihood using also Jensen's inequality. By Jensen's inequality we know that

$$\sum_x p(x)f(x) \geq f\left(\sum_x p(x)x\right)$$

that is to say the average of the function is greater than or equal than the function of the average for convex functions. When maximizing the likelihood it is common practice to maximize the log of the likelihood instead of the likelihood in order to avoid numerical problems, in that case the function is concave therefore

$$\log \sum_x q(x)\frac{p(x, y \mid \Theta)}{q(x)} \geq \sum_x q(x) \log \frac{p(x, y \mid \Theta)}{q(x)}.$$

In order to find the optimal lower bound with respect to q(x) given the current settings of the parameters Θ. Introducing a Lagrange multiplier λ to enforce the constraint $\sum_x q(x) = 1$ the objective function becomes

$$F(q(x)) = \lambda\left(1 - \sum_x q(x)\right) + \sum_x \log p(x, y \mid \Theta) - \sum_x q(x) \log q(x)$$

Taking the derivative

$$\frac{\partial F}{\partial q(x)} = -\lambda + \log p(x, y \mid \Theta) - \log q(x) - 1$$

therefore

$$q(x) = p(x \mid y, \Theta)$$

In the E step we just set q(x) $\leftarrow p(x \mid y, \Theta)$. In the Gaussian mixture model the hidden variable $x^{(c)}$ indicates which component observation $y^{(c)}$ belongs to, therefore:

$$r_k^{(c)} = q(x^{(c)} = k) \propto \frac{\pi_k}{\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}(y^{(c)} - \mu_k)^2\right\}, \sum_k r_k^{(c)} = 1$$

In the M step we maximize the optimal lower bound with respect to the parameters of the model to obtain a new set of parameters Θ^{new}

$$E = \sum q(x) \log[p(x \mid \theta)p(y \mid x, \theta)]$$

Taking derivatives we obtain Θ^{new} of the mixture model:

$$\mu_k = \frac{\sum_c r_k^{(c)} y^{(c)}}{\sum_c r_k^{(c)}} \tag{2.3}$$

$$\sigma_k^2 = \frac{\sum_c r_k^{(c)} (y^{(c)} - \mu_k)^2}{\sum_c r_k^{(c)}} \tag{2.4}$$

$$\pi_k = \frac{1}{n} \sum_c r_k^{(c)} \tag{2.5}$$

2.1.2 Mathematical Morphology

Mathematical morphology is the analysis of signals in terms of shape. Morphological operators are defined as operations carried out on point sets of any dimension. A small point set called structuring element is moved systematically across the original dataset in order to obtain another point set result of the operation. The primary morphological operators are:

- Dilation. The dilation X $\oplus$ B is the point set of all possible vector additions of pairs of elements, one from each of the sets X and B. In a binary image, objects of the same size and shape as the structuring element are made bigger.

- Erosion. X $\ominus$ B is defined as the set of points p $\in$ X for which all possible p + b, b $\in$ B, are in X. In a binary image the structuring element is fit inside the shapes present in the image and all the pixels belonging to the shape but outside the structuring element are removed. Shapes for which the structuring element does not fit inside remain unchanged.

More complex morphological operations are defined as a combination of erosion and dilations. Opening is and erosion followed by a dilation X $\circ$ B = (X $\ominus$ B) $\oplus$ B and has the effect of removing elements from the image smaller than the structuring element. Closing X $\bullet$ B = (X $\oplus$ B) $\ominus$ B is a dilation followed by an erosion an is the dual transformation of opening $(X \bullet B)^c = X^c \circ$ B

Granulometry

An important morphological operation is granulometry. Granulometry was first used as a tool for the study of porous materials. In order to study the size distribution of objects in an image we apply a sequence of openings increasing the size of the structuring element. Each opening removes pixels from the image and by computing the difference in number of pixels in the image between two consecutive openings we can estimate the sizes of the particles present. If we define the number of pixels in an image I after applying the opening with

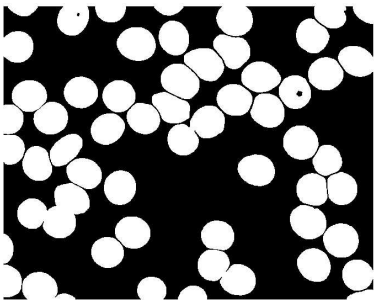

Figure 2.1: Cells binary image

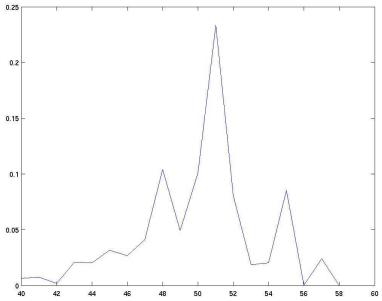

Figure 2.2: Granulometric pattern function

structuring element k as $S(k) = sum(\psi_{ks}(I))$, $k = 1, 2, ..., \psi$ is the opening operation, then the granulometric pattern function can be defined as

$$P(k) = N(k+1) - N(k), \tag{2.6}$$

$$N(k) = 1 - \frac{S(k)}{sum(I)} \tag{2.7}$$

Figure 2.2 shows the granulometry pattern function of a binary image, shown in figure 2.1, that contains particles of sizes 30, 40 and 50. The pattern function has peaks at these values.

2.2 Methodology

In this work a probabilistic approach is used. We assume that pixels belonging to different regions in the image follow a similar probability distribution and therefore attempt to model that distribution using a mixture of Gaussians. Each Gaussian of the mixture will represent a class c and each pixel will be assigned to the class for which $p(x \mid c)$ is highest. Prior to segmentation, preprocessing of

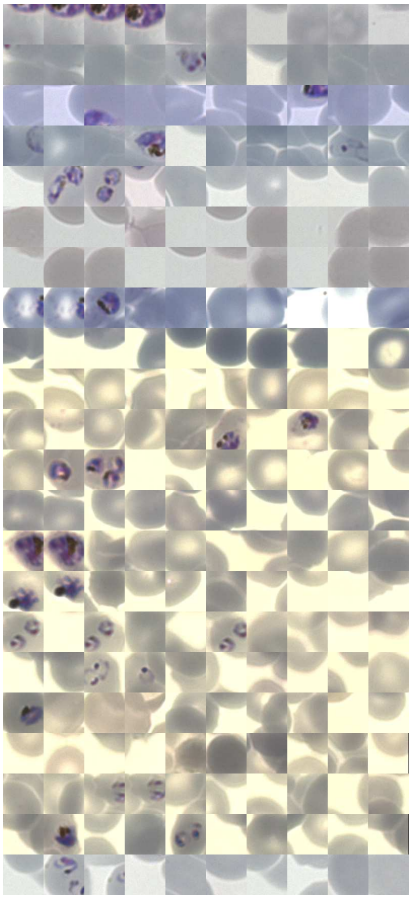

Figure 2.3: Unnormalized image

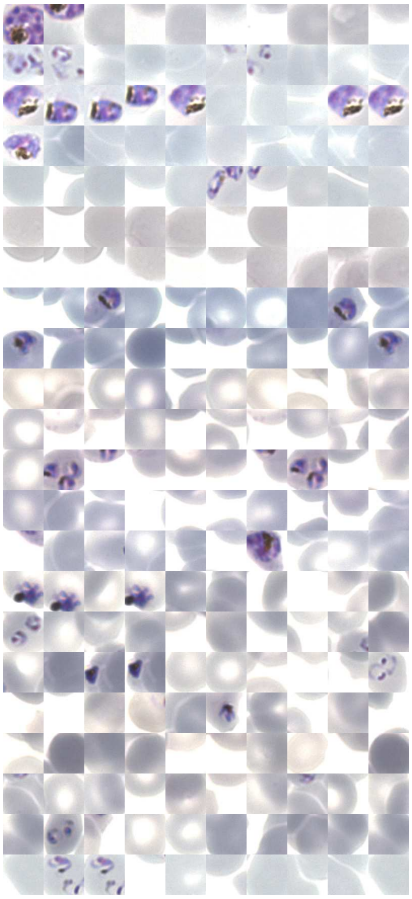

Figure 2.4: Grayworld normalized image

the image is performed in order to reduce the variability present in the images. After the segmentation step, pixels have been labeled as background, infected or parasite and in the final step parasites are detected and a count of the total number of cells is carried out. This system performs three major steps:

- Preprocessing of images for variability reduction

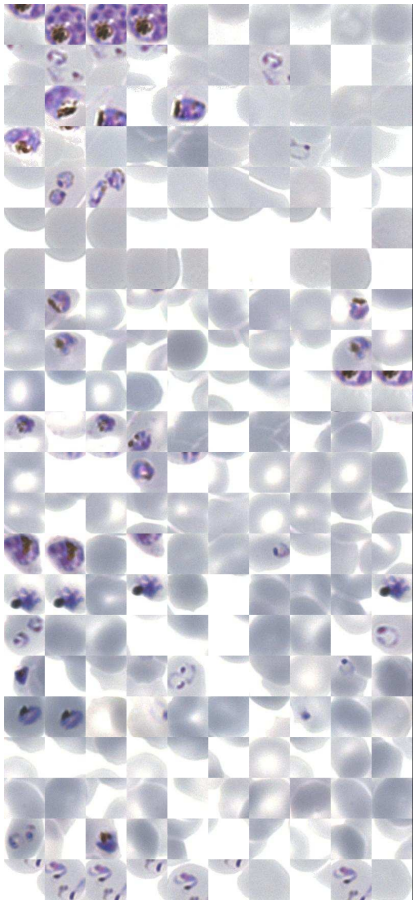

Figure 2.5: Color tuned image

- Segmentation of images for parasite and uninfected red blood cells detection

- Total number of cells and parasites counting

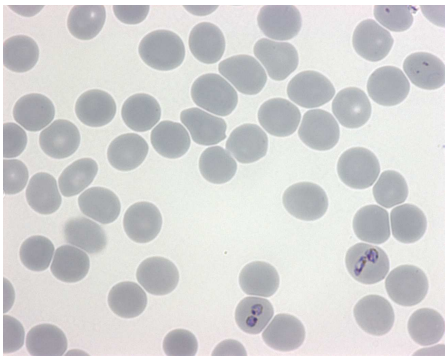

Figure 2.6: Original image

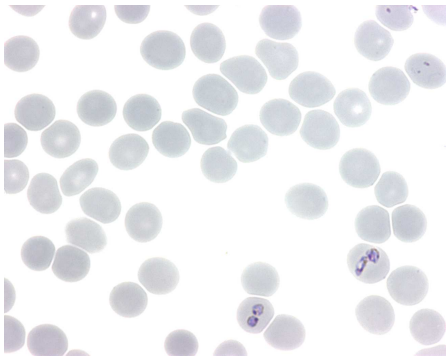

Figure 2.7: Grayworld normalized image

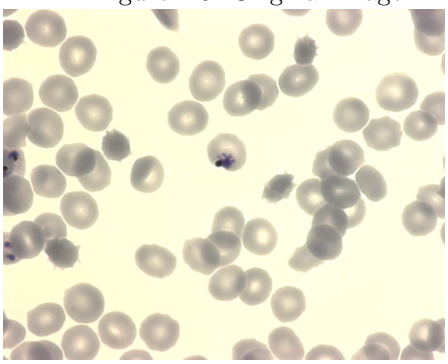

Figure 2.8: Original image

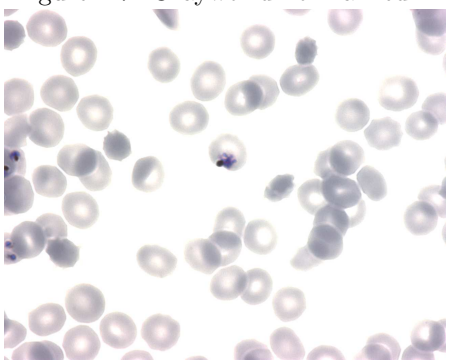

Figure 2.9: Grayworld normalized image

2.2.1 Preprocessing of images for variability reduction. Color normalization

The aim of pre-processing is to improve the image data by suppressing unwanted distortions or enhancing some image features important for further processing. In this work, preprocessing of the images is necessary in order to reduce the variability due to different factors. In the process of obtaining an image a number of variables influence the result and account for the variability present in the images. The experience of the laboratorian is an important factor because

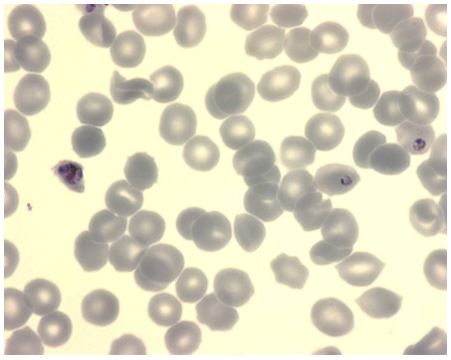

Figure 2.10: Original image

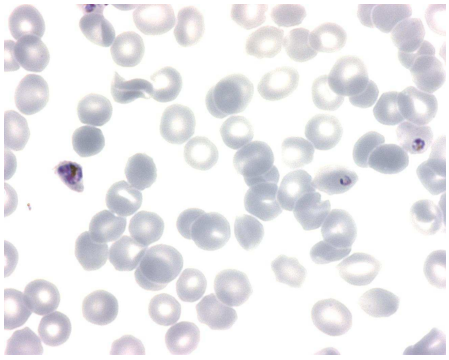

Figure 2.11: Color tuned image

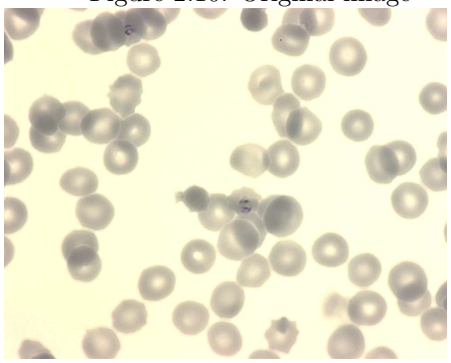

Figure 2.12: Original image

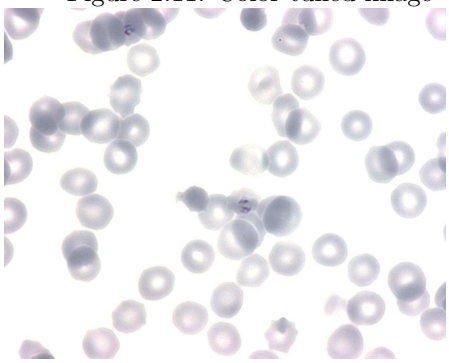

Figure 2.13: Color tuned image

images will vary depending on the quality of the preparation of the slide and the capturing of the images. Also the quality of the microscope and the capturing device and the lighting conditions will influence the result. This variability makes the recognition task very difficult and finding good methods to reduce it may help to solve the recognition task. In this work three normalization procedures are attempted: grey-world, color tuning and comprehensive. This methods are described in the following sections.

Grayworld normalization

Grayworld normalization assumes that the average of any scene is grey and any variations are due to illumination effects. In order to remove these variations from an image I we obtain the values of each channel (RGB) of the normalized image using the following formula:

$$I_{ij} = \frac{NI_{ij}}{\sum_{ij} I_{i,j}} \tag{2.8}$$

where N is the number of pixels in the image. We can see that the mean of each channel is 1. Figures 2.6 and 2.8 show the images before normalization and Figures 2.7 and 2.9 show the same images after normalization using this method.

Cell color tuning

After normalizing the images, the color of the cells from different images presents still some variability. In order to make the color of the cells more uniform two steps are carried out:

- **Separation of cells and background.** After normalization, the background is easily separated from the rest of the objects in the image. To achieve that, a mixture of two Gaussians is first trained using a sample of background and a sample of a cell. This model is then used to label the images. A binary image is obtained with one representing cells and zero representing background. These binary images are used to obtain the RGB values of the color of the non-background pixels.

- **Tuning of cell color.** The mean m of the cell pixels of each image and the grand mean M, which is the mean of the cell pixels of all the images, are computed. The new values of the pixels are transformed as follows:

$$I_i = I_i - m_i + M$$

The color of uninfected cells and parasites is made more uniform among all images. This should help to clusterize more easily the objects we are interested in. Figures 2.10 and 2.12 show the images before normalization and Figures 2.11 and 2.13 show the same images after normalization using this method.

Comprehensive normalization[FSC98]

Comprehensive normalization attempts to normalize away variation due to illuminant color and lighting geometry together at the same time. First the variation due to lighting geometry is normalized by using:

$$r = \frac{r}{r+g+b}, g = \frac{g}{r+g+b}, b = \frac{b}{r+g+b},$$

where r, g, b are the values of each of the RGB channels of the pixels. Then variation due to illuminant color is removed by using:

$$r_n = \frac{Nr_n}{r_1 + ... + r_N}, g_n = \frac{Ng_n}{g_1 + ... + g_N}, b_n = \frac{Nb_n}{b_1 + ... + b_N},$$

where N is the total number of pixels. This process iterated until each normalization step is idempotent. After normalizing the images using this method the distinction between background and uninfected cells was not very clear and attempts to segmenting the image afterwards were not successful.

2.2.2 Segmentation of images for parasite and uninfected red blood cells detection

Image segmentation is one important step leading to the analysis of image data. Its main goal is to divide an image into parts that have similar characteristics. There exist different methods for image segmentation: global knowledge about an image, edge-based and region-based segmentation. In the first group, thresholding is the main algorithm. Knowledge about the characteristics of the objects present in the image like their brightness, light reflectance of surface can be used to build a histogram and based on that information set a threshold to identify the different objects. Edge-based methods for segmentation rely on edges found by edge detecting operators in order to identify objects. Region-based segmentation methods find regions in an image normally based on homogeneity of the regions. The goal is to divide the image into regions of maximum homogeneity. In practical applications a combination of al methods is used in order to achieve a good segmentation performance, specially when dealing with images that contain noise or low quality images.

In this work we try a probabilistic approach where pixels of the image are assigned to one of three different classes: background, uninfected area, infected area. Each Gaussian component of the model is trained to recognize a specific class. The result of the segmentation step is an image where each pixel has been labeled as belonging to one of the classes. Training the model is sometimes imposible because of the nature of the data. Sometimes one Gaussian is responsible for a single data point yielding a singularity because the likeli-

hood goes to infinity as the variance goes to zero. A modification of the EM algorithm is needed in order to avoid this behavior and train the model. Some configurations are penalized and will give a lower value of the likelihood. This configurations correspond to the cases where only one gaussian is responsible for a few data points and could degenerate towards a singularity.

Penalized EM algorithm

The likelihood function for mixture of Gaussians distributions can degenerate towards infinity. The probability of a data point under a mixture of Gaussians is the sum of Gaussian distributions probabilities. For univariate models the variance parameter appears in the denominator of the formula of the probability so when the variance approaches zero the likelihood tends to infinity yielding a singularity. In order avoid this behavior a proposed solution is to assign a prior distribution over the variances [RI02]:

$$f(\sigma_i^2) = \frac{\alpha^{\beta-1}}{\Gamma(\beta-1)} \frac{1}{\sigma_i^{2\beta}} exp\left\{-\frac{\alpha}{\sigma_i^2}\right\} \tag{2.9}$$

(9) corresponds to the inverted gamma distribution which is chosen for being the conjugate prior for the variance of a scalar Gaussian density. This property allows us to easily re-estimate the formulas of the EM algorithm for the mixture of Gaussians. The new formula for the variance is:

$$\sigma_k^2 = \frac{2\alpha + \Sigma_c r_k^{(c)} (y^{(c)} - \mu_k)^2}{2\beta + \Sigma_c r_k^{(c)}}, \alpha > 0, \beta > 1. \tag{2.10}$$

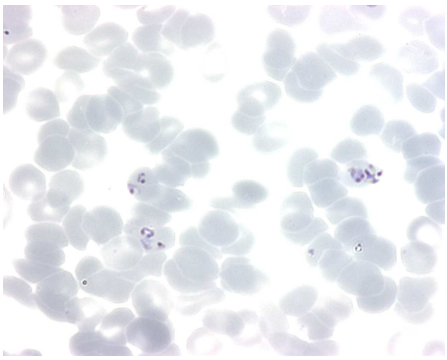

Figure 2.14: Original image

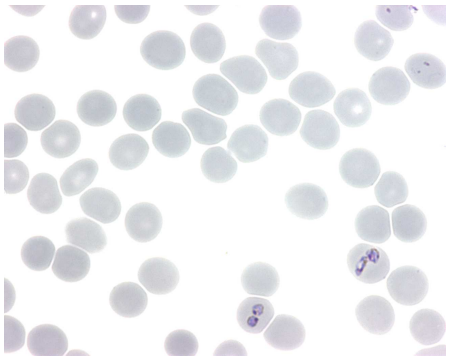

Figure 2.15: Original image

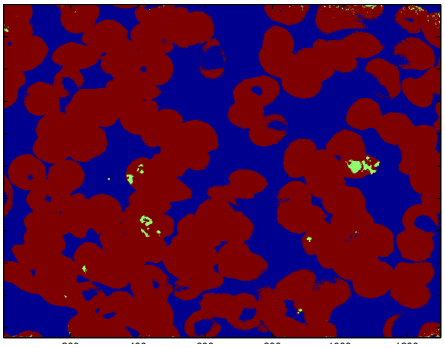

Figure 2.16: Clusters using 3 Gaussians

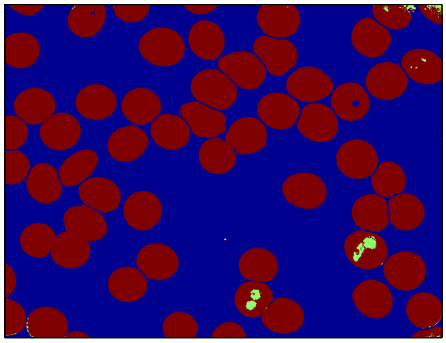

Figure 2.17: Clusters using 3 Gaussians

2.2.3 Total number of cells and parasites counting

After the segmentation process there is an image where each pixel has a label assigned to it. Figures 2.16 and 2.17 show the labels obtained after clusterizing the images shown in figures 2.14 and 2.15 respectively. In order to count the number of cells, all the objects from the image that are not background need to be obtained. We use the image obtained after segmentation and set to zero all the pixels labeled as background and to one the rest, we end up with a binary image where the ones represents pixels belonging to cells and the zeros represent

background (See figure 2.1). An important parameter for the recognition of cells is their size. To obtain the sizes of red blood cells we apply granulometry to an image where the cells do not overlap. We can now use this parameter, the labels obtained after segmentation and the binary image with the components in order to count the number of cells and the number of infected cells. From the binary image we obtain the number of connected components using the Matlab function bwlabel. This function returns another image of the same size as the original binary image where all the pixels that form part of a connected component are assigned the same number, one number for each connected component of the image. Some of the components will be too large to be a cell and that might be due to the fact that some cells are overlapping, and form a unique connected component. We can use the knowledge of the size of the cells to guess how many cells there might be in that component. In order to determine wether a cell might be infected we obtain the labels of all the pixels belonging to that cell and check how many pixels there are labeled as infected and set a threshold in order to classify that cell as infected or uninfected. Sometimes cells overlap making the counting task difficult even for humans. Normally an experienced operator would discard zones of the image where cells are too cluttered and would choose a different area to obtain a more accurate count. In some cases only two or three cells appear together and are easy to count considering the average sizes of cells and the shape of the object. In the next section we describe an attempt to imitate what a human does to separate two or more cells by studying the shape of the objects.

Separation of cells

In this work we analyze the concavity of each of the connected components found to determine if they are individual red blood cells or regions containing more than one. We use the technique described in [FKZ95]. The input to the algorithm is the binary image BW containing the connected components used to count the number of cells obtained after segmentation. Each connected component is analyzed one by one. First all the pixels that do not belong to the component to be analyzed are reset to zero, the result is an image where only the pixels of the component to be analyzed are one. The coordinates x, y of this pixels are extracted using the Matlab function find and a new image **I** containing only the component is obtained, where **I = BW(y1:y2,x1:x2)** and y1,y2,x1,x2 are the vertices of the square containing the component. A morphological gradient with structuring element of size 1 is performed on this image in order to extract the contours of the component. The concavity of the contour is then analyzed, each pixel p of the contour has a concavity value C calculated as follows:

$$C_p = \sum_{j-1}^{j+1} \sum_{x=1}^{5} \sum_{y=1}^{5} M_j \cap I$$

where M_j is a 5 X 5 mask. For the j_{th} point of the contour the value of concavity is calculated as the number of points of a 5 x 5 mask centered on j that intersects the binary shape. Finally the values of the two adjacent contour neighbors are added to the contour value. A cell is considered concave if the standard deviation of all the concavity values is higher than a threshold. Only concave cells are processed further. The coordinates of all the points of the

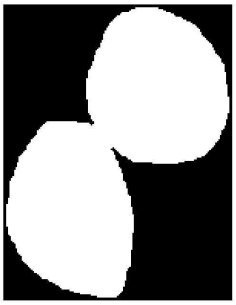

Figure 2.18: Cell

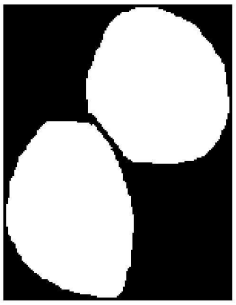

Figure 2.19: Composite cell separated

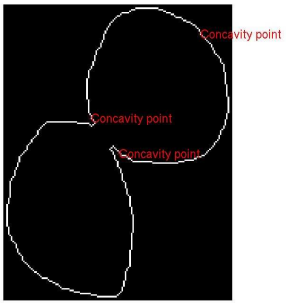

Figure 2.20: Cell contour and concavity points

contour are extracted in order, keeping the order is important because later in the process only points that are a certain distance apart in the contour can be chosen as a candidate pair for cutting. Candidate points are found by selecting a set of points from the contour with a concavity value higher than a threshold. If no points are found, the threshold is iteratively reduced. From this set, the point with highest concavity value is chosen. From the same set

another set is obtained containing only points that are not too close to this maximum and the point with highest concavity value is chosen. This process is repeated iteratively until there are no more points in the set. After this process a set of candidate points is obtained. All the distances between a pair of points are computed and a cut is carried out if the distance is smaller that $\frac{1}{4}$ of the perimeter of the contour. Good results are obtained when two or three cells were together but further work needs to be done for more complex structures. Figure shows a connected component to be analyzed. Figure shows the contour of the component and the concavity points found. Finally figure shows the cut performed and the two individual cells that result from it.

2.3 Experiments and Results

In this section, a series of experiments are devised in order to evaluate and draw comparisons between different approaches to the problem. To evaluate the segmentation performance of the system, the following values are used: **the accuracy**, which shows the percentage of correct pixels classified, **SP**(Sensitivity to parasite), **SpP**(specificity to parasite), **SU**(sensitivity to uninfected) and **SpU**(specificity to uninfected), **SB**(sensitivity to background) and **SpB**(specificity to background). The sensitivity and specificity study shows how well the system classifies pixels. Sensitivity is defined as the ratio $\frac{TP}{TP+FN}$, where TP are the true positives and FN are the false negatives. Similarly, specificity is defined as $\frac{TN}{TN+FP}$, where TN are the true negatives and FP are the false positive. Dif-

ferent systems require different sensitivity and specificity values depending on the costs of making mistakes. In this case the system should be very sensitive and very specific, because the elimination of as many FN and FP as possible is required.

The pixels of the images to be segmented are labeled by an expert by hand. A user interface was developed to allow users to assign different labels to pixels. The segmentation performance is therefore assessed comparing the labels assigned to each pixel by the system to the labels assigned by the human expert. Also cells and parasites needed to be counted by hand and results of the counting compared to those obtained by the system. Appendix A gives a description of the functions used to manually count parasites and cells and label pixels. The data set consists of 22 images obtained using a microscope under different lighting conditions. In order to train our models bits of images which are important for recognition like parasites, uninfected cells and background were obtained by hand and put together in a new big image. Figure 2.3 shows an example of a training image obtained this way before normalization. The training images are normalized using the methods described in 4.1.1 and 4.1.2. Figure 2.4 shows the training image normalized using method described in 4.1.1 and figure 2.5 shows the training image normalized using the method described in 4.1.2.

Experiment 1: Performance of the system when trained using normalized and unnormalized data.

The aim of this experiment is to compare the performances of the system when

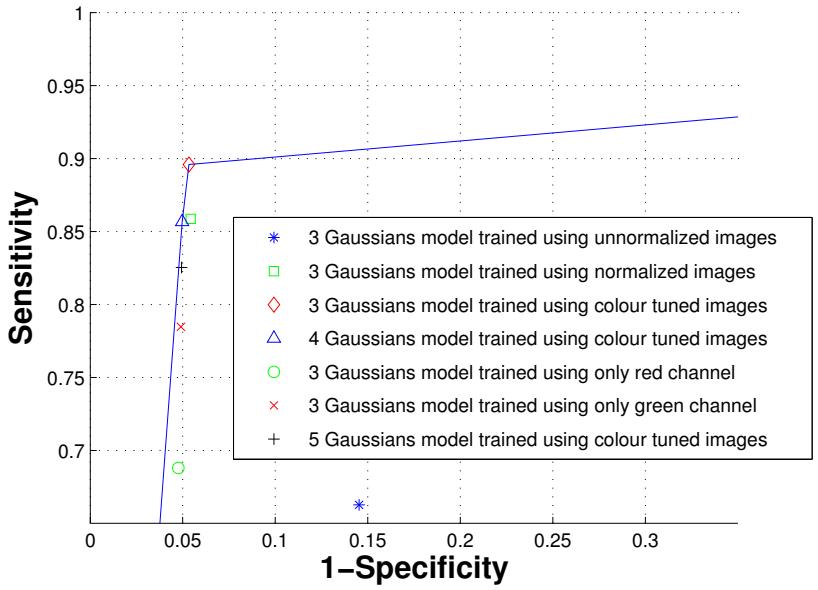

Figure 2.21: Segmentation performance of the models

it is trained using unnormalized versus normalized data. The model is composed of 3 Gaussians. Pixels in the image are assigned to one of three different classes: background, cell and infection. Images in the training set present high variability due to a number of factors that might affect the quality of the images when taken with the microscope. Some of these factors are: Light source, quality of the microscope, experience of the expert. Gray world normalization is applied on the training images in order to reduce the effect of some of these factors. Two models are trained using the normalized and the unnormalized data and their performances are compared. The following tables show the results of the segmentation.

Results of pixel classification when unnormalized images are used

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	27639	11179	9758
	Background	544034	10057435	547018
	Uninfected	256339	2108893	3844705

accuracy = 80.0240 %

SP = 56.9 % SU = 47.1 % SB = 94.87 %

SpP = 95.39 % SpU = 95.03 % SpB = 66.12 %

Results of pixel classification when normalized images are used

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	35505	715	12356
	Background	23912	10112033	1012542
	Uninfected	399923	283069	5526945

accuracy = 90.04 %

SP = 68.9 % SU = 89.0 % SB= 99.7 %

SpP = 98.39 % SpU = 90.85 % SpB = 95.4 %

As seen from the tables above, the overall performance of the system is considerably improved when images are preprocessed. This result agrees with intu-

ition since the normalization process should make the images more uniform and therefore make the clusterization easier. The background in recognized very accurately and also there is an improvement in recognizing parasites.

Experiment 3: Performance of the system using color tuned training set

After the images are normalized using grayworld the color of the cells is still not uniform among the images. In this experiment the color of the cells is tuned the performance of the system is compared when images are only normalized to the performance when training images are normalized and cells are color tuned. The training images used are shown in figures 5 and 6 and the methods to normalize are described in 4.3.1 and 4.3.2.

Results of pixel classification using a set of normalized and color-tuned images

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	38611	727	9238
	Background	79897	10111639	956951
	Uninfected	306759	289587	5613591

accuracy = 90.56 %

SP = 79.49 % SU = 90.4 % SB = 99.4 %

SpP = 97.77 % SpU = 91.37 % SpB = 95.3 %

The sensitivity to parasite increases, therefore false negatives have been reduced

although there is a decrease in the specificity which indicates that false positive are present therefore more uninfected areas or background are being labeled as parasites than before. In the next experiment an attempt is made to increase this specificity and also increase the sensitivities and specificities of background and uninfected.

Experiment 4: Performance of the system using four Gaussians and trained using normalized and color tuned images

After images are normalized and the color is made uniform it is noticeable that the color of the uninfected areas of cells present two different tones. In order to capture these tones one more gaussian is added to the model. By doing this it is hoped that one Gaussian will capture the infected area, one gaussian will capture the background and the other two will recognize the two different tones of the uninfected cells.

Results of pixel classification using a set of normalized and color-tuned images and 4 Gaussians

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	30245	727	17604
	Background	954	10111771	1035762
	Uninfected	9226	289853	5910858

accuracy = 92.22 %

SP = 62.26 % SU = 95.18 % SB = 99.8 %

SpP= 99.8 % SpU = 90.59 % SpB = 95.35 %

It can be seen from the results in the table that there is a dramatic decrease in the number of errors made by the system when recognizing parasites versus cells and parasites versus background. In previous models pixels that should have been labeled as background or cell were mistakenly labeled as parasites. With this new model this error is reduced significantly. The drawback is that some areas of the infection are considered to be part of the uninfected cell.

Experiment 5: Performance of the system when only one color channel is used to train and classify.

In this experiment only one of the color channels of the RGB image is used to train and classify. The images are first normalized and color tuned.

Results of pixel classification using the red channel

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	10713	820	37043
	Background	18	10377122	771347
	Uninfected	3906	469456	5736575

accuracy = 92.6318 %

SP = 22 % SU = 92.38 % SB = 92.4 %

SpP= 99.9 % SpU = 92.78 % SpB = 93.8 %

Results of pixel classification using the green channel

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	24652	824	23100
	Background	92	10310136	838259
	Uninfected	27461	400985	5781491

accuracy = 92.5 %

SP = 50.7 % SU = 93.10 % SB = 92.4 %

SpP = 99.8 % SpU = 92.31 % SpB = 93.5 %

Results of pixel classification using the blue channel

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	0	12325	36251
	Background	0	11107946	40541
	Uninfected	0	3661495	2548442

accuracy = 78.4 %

SP = 0 % SU = 41.04 % SB = 99.6 %

SpP = 1 % SpU = 99.31 % SpB = 41.3 %

The results obtained using the blue channel are a bit misleading. Although the accuracy of the system seems to have increased, if the specificity and sensitivity of the system are computed it can be seen that its capacity to identify

parasites is reduced drastically. Images usually contain a small proportion of parasites compared to the proportion of uninfected cells and background and this accounts for the high accuracy obtained. Using the other channels the performance of the system is very poor.

Experiment 6: Performance of the system using 5 Gaussians and training set normalized and color tuned.

In order to improve the sensitivity of the system the aim is to achieve a better modeling of the tones of the parasite. By using five Gaussian we hope two Gaussian will recognize parasites, two Gaussians will recognize uninfected areas and finally one Gaussian will identify the background

Results of pixel classification using 5 Gaussians

		PREDICTED LABELS		
		Infected	Background	Uninfected
TRUE LABELS	Infected	30123	725	17728
	Background	996	10109604	1037887
	Uninfected	10824	289260	5909853

accuracy = 92.2 %

SP = 62.01 % SU = 95.17 % SU = 90.6 %

SpP = 99.93 % SpU = 90.57 % SpU = 95.3 %

A similar result is obtained when using only four Gaussians. The performance does not improve significantly.

Figure 2.21 shows the sensitivity and specificity values of each of our models. These values were obtained by averaging the sensitivities and specificities to infected, uninfected and background. As seen from the graphic three models achieve very good performance: the two models that were trained and tested on normalized images using either of the two methods described earlier and composed of three Gaussians, and the model trained and tested on images color tuned and composed of four Gaussians. The model composed of five Gaussians achieved also good results, similar to those obtained by using four Gaussians. The model trained and tested on unnormalized images achieves poor results.

Experiment 7: Cell counting.

In this experiment the overall performance of the system is assessed. The total number of cells and the number of infected cells are obtained and compared with a count made by a human. The best segmentation performance is obtained using four Gaussians and normalizing the training set using both methods described in 4.3.1 and 4.3.2 before training the model. This model is used to segment the image and then count the total number of cells and the number of infected cells using the method described in 4.5. The table below shows the results obtained.

TH - Total number of cells counted by hand

TA - Total number of cells counted by system

TIH - Total number of infected cells counted by hand

TIA - Total number of infected cells counted by system

Images	TH	TA	TIH	TIA
I1	15	19	1	1
I2	76	81	3	2
I3	76	84	3	3
I4	76	79	2	2
I5	50	51	2	2
I6	17	20	0	1
I7	17	20	0	1
I8	59	61	2	2
I9	46	50	2	2
I10	41	42	0	1
I11	44	47	3	3
I12	36	36	2	2
I13	50	50	0	0
I14	67	67	3	3
I15	58	54	3	3
I16	49	49	2	1
I17	52	55	1	2
I18	50	49	2	2
I19	44	45	1	2
I20	41	49	1	1
I21	63	60	2	2
I22	51	56	1	2

2.4 Discussion

This work attempted an unsupervised learning approach to image segmentation of malaria infected red blood cells combined with mathematical morphology for cell counting. The image was segmented using a mixture of Gaussians model and then granulometry was used to obtain the average size of cells in order to count the total number of cells. The two main difficulties encountered during the project were the variability of the images, which made the segmentation process more problematic, and the fact that cells sometimes overlap in the image making the counting of the cells a very difficult task. The segmentation of the image using a mixture of Gaussians gave very good results when images were normalized. The counting of cells gave satisfactory results taking into account that sometimes it is impossible for two experts to agree on the number of cells present in an image. Although the overall performance of the system is satisfactory, further testing should be done using more images taken under a range of different conditions in order to assess the performance of the preprocessing and segmentation methods. This work could be taken further by attempting to combine the segmentation results obtained using a Mixture of Gaussians with supervised learning methods and morphology. Morphology could be used to describe the parasites and help to discriminate between uninfected areas and parasites, which is one of the problems encountered when using only the Mixture of Gaussians. Supervised learning methods could be used to learn what parasites are and then analyze the parasites areas found by the Mixture of Gaussians to decide their class.

Bibliography

[Bai92] D. G. Bailey. Segmentation of touching objects. In *Proceedings of the 7h New Zealand Image Processing Workshop*, pages 195–200, August 1992.

[Bil98] Jeff A. Bilmes. A gentle tutorial of the em algorithm and its applications to parameter estimation for gaussian mixture and hidden markov models. Technical report, International computer science institute, 1998.

[CBGM02] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld:image segmentation using expectation-maximization and its applications to image querying. *Transactions on pattern analysis and machine intelligence*, 24, August 2002.

[Del02] Frank Dellaert. The expectation maximization algorithm. Technical report, College of Computing, Georgia Institute of Technology, 2002.

[FKZ95] G. Fernandez, M. Kunt, and J.-P. Zryd. A new plant cell image segmentation algorithm. *j-LECT-NOTES-COMP-SCI*, 974, 1995.

[FSC98] Graham D. Finlayson, Bernt Schiele, and James L. Crowley. Comprehensive colour image normalization. In *Proceedings of the 5th European Conference on Computer Vision*, volume 1, pages 475–490, July 1998.

[GJ94] Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via em approach. In *Advances in Neural Information Processing Systems 6*. Kaufmann Publishers, 1994.

[RDKJ01] Cecilia Di Ruberto, Andrew Dempster, Shahid Khan, and Bill Jarra. Analysis of infected blood cell images using morphological operators. *Image and Vision Computing*, pages 133–146, 2001.

[RG99] Sam Rowels and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[RI02] A. Ridolfi and J. Idier. Penalized maximum likelihood estimation for univariate normal mixture distributions. Technical report, cole Polytechnique Fdrale de Lausanne, 2002.

[SHB98] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, Analysis, and machine vision*. International Thomson Publishing, 1998.

[SK05] H.S. Sheshadri and A. Kandaswamy. Detection of breast cancer tumor based on morphological watershed algorithm. *ICGST International Journal on Graphics, Vision and Image Processing*, V5:17–21, 2005.

[TDK05a] F.B Tek, A. G. Dempster, and I. Kale. Blood cell segmentation using minimum area watershed and circle radon transformations. In *Proc. Int. Symp. on Mathematical Morphology*, volume 1, pages 739–742, April 2005.

[TDK05b] F.Boray Tek, Andrew G. Dempster, and Izzet Kale. Parasite detection in images of stained blood slides using bayesian classifier. 2005.

[vOGD+02] P. van Osta, J.M. Geusebroek, K. Ver Donck, L. Bols, J. Geysen, and B. M. ter Haar Romeny. The principles of scale space applied to structure and colour in light microscopy. *Proc. R. Microsc. Soc.*, 37(3):161–166, 2002.

Appendix A

User manual

A.1 System description

The system is composed of a set of functions that are used to perform different tasks. This tasks can be classified into three groups: functions that perform preprocessing, system functions which are the functions that perform the main tasks of the system: segmentation and counting and finally auxiliary functions which are used for different tasks like labeling of images and creation of training images.

Basically the system counts the total number of cells in an image and the total number of infected cells. The images follow a logical path that consists of the following steps: first the image are preprocessed, after this images are segmented, in order to segment the images a model of the objects present in the images needs to be created. Finally cells are counted. In the following section

a description of the functions involved in each of the steps is given.

Image preprocessing functions

grayworld. This function receives as input an image and returns the normalized image. After grayworld normalization the mean image color maps to (1,1,1) that is to grey. The purpose of this normalization is to remove the variability due to illuminant color.

normalizeImages. This function receives as input a set of RGB images in a N X M X 3 X K matrix and the parameters of a Gaussian model with 2 Gaussians where one Gaussian has been trained to recognize background and the other anything different to background. Before being able to use this function we need to create a 2-Gaussian model and pass the parameters of this model to the function. The function returns another matrix with the images color tuned. Basically this function uses the model to obtain from each image a new image where background and non-background pixels have been identified. Once we know where the non-background pixels are we compute their mean value for each image in the set. As we process the images of the set we compute the grand mean which is the mean of all the non-background pixels values of all the images. The function subtract to each non-background pixel its mean value and adds the grand mean. This results in a more uniform color of all the non-background pixels of the set.

comprehensive.. This function normalizes an image using the comprehensive method. Variation due to lighting geometry (eg: light intensity, light direction) and illuminant source are removed. It receives as input the image to be normalized and returns the image normalized.

System functions

EM. This is one of the core functions of the system. It receives as input parameters the number of Gaussians we want our model to have, the Data and the number of iterations. This function finds the parameters of the mixture of Gaussians model given some training data and uses the EM algorithm to find them. A set RGB images is passed to the function and it will find the best parameters for the Gaussians. The number of Gaussians should be chosen accordingly to the number of different classes of objects we expect to find in the image. The parameters of the model can be saved onto a file and used to identify classes in a new set of images.

analyzeCells. This function can be used after segmentation in order to separate composite cells into their constituents. The function receives as input parameters a binary image obtained by setting, in the image obtained after segmentation, each non-background pixel to 1 and each background pixel to 0. The function returns another binary image where some of the cells considered composite by the function are separated. This function works well when the composite cell has two or three clear cells together. Further work needs to be

done when cells appear too cluttered.

testModel. Once a model has been created using a set of training images, new images can be segmented using that model. This function accepts as input parameters the parameters of the mixture and the image and returns a matrix containing the labels assigned to each pixel.

countCells. This function is used to count the number of uninfected and infected cells in an image. The image should be an image of labels obtained after segmentation where each pixels is assigned to a class. The function receives as input parameters the image labeled in the segmentation step and the average size of cells. This size can be obtained by hand or using the function cellSize which performs a granulometric analysis on an image. In order to count the infected cells the pixels belonging to each of the cells detected are analyzed, their labels are stored in the image result of the segmentation step. The function returns two values: the total number of cells in the image and the number of infected cells.

Auxiliary functions

makeImage. This function is used to create a training image. Training images are created selecting bits from a set of images. These bits are selected by the

user. The function opens up a set of images and the user selects from each image the bits he considers important for the recognition task. The function internally uses the Matlab function roipoly. The function returns a new image which is the concatenation of all the bits selected by the user from the images.

labelImage. In order to assess the segmentation performance of the system we need to compare the labels that our model assigns to each pixel to the labels that a human expert assigns to each pixel. This manual labeling can be done using this function. labelImage uses internally the function roipoly and therefore all the pixels inside the polygon are set to 1 and the rest to zero. Using this function the user can label the pixels of all the cells of an image. The user can keep track of the cells already labeled because all the pixels belonging to that cell in the image are reset to black. When the user closes the window the matrix with the labels is saved onto a file. The paths can be reseted and are pointing to the directories I used during my experiments.

GUI. I developed this user interface myself in order to label the images. To run this user interface, type GUI(n) from the command line where n is the number of different labels you will need. The image you select appears and over it an adjustable grid. When you double click in a square all the pixels in that square will be labeled with the label selected in the combo box. You can click once in a square and click once in a different square and this will result in all the pixels inside the polygon being set to the corresponding label. This method of

labeling the images is quicker than the previous one but not as accurate. The resulting image with all the pixels labeled is saved onto a file.

countCellsByHand. This function helps the user to count manually the number of cells in an image. This count is used afterwards to compare with the count obtained by the system. The function receives as input parameter a binary image (with ones for non-background and 0 for background). Each time the user clicks on a cell the counter adds 1 to its value and the point is set to red in order to allow the user to keep track of the cells already counted

cellSize. In order to count cells we need to know their average size. This average size can be obtained automatically using the granulometric analysis performed by this function. Alternatively this size can be estimated by hand selecting a few representative cells and averaging the total number of pixels. The cells can be selected using the matlab function roipoly. As we saw earlier the function countCells uses this average size to estimate the number of cells present in the image.

createTable. This function receives as input two matrices containing the true labels of the pixels of an image, the labels predicted by the model and the number of classes. It returns the classification performance of the model in the shape of a table with as many rows and columns as classes. From the values of the table an analysis of sensitivity and specificity can be made for each classifier.

Appendix B

MATLAB code