Imperial College London
Department of Computing

Automatic Cell Tracking in Noisy Images for Microscopic Image Analysis

by

Pedro Damian Kostelec

September 2014

Supervised by Ben Glocker

Submitted in part fulfilment of the requirements for the
MSc degree in Computer Science (Artificial Intelligence) of Imperial College London

Contents

1 Introduction DRAFT I

In this introductory chapter we describe the problem of tracking cells in hundreds of microscopy images manually and motivate the development of an automatic cell tracker. We also outline the objectives and contributions of the project and provide a report outline for the following chapters.

1.1 Motivation DRAFT I

Recent advances in intravital microscopy enable us to study the behaviour of different cells without excessively modifying the natural environment in which these cells are found within the observed organism.

Neutrophils are a type of leukocytes that have a crucial role in the clearance of infections [1]. A significant reduction in the number of neutrophils in the human body (or in mice) leads to severe immunodeficiency or death. They can be found in the bone marrow, liver, spleen and lung. Direct observation of neutrophils should help explain their presence in these organs, especially their large quantity in the lung.

A recent study [2] has shown that the lifespan of neutrophils is much longer than previously known (up to 12.5 hours for mice and 5.4 days for humans). Although this specific study is a source of doubtful criticism [3], the longevity of neutrophils increases during inflammation. This longer lifespan may permit them to perform a wider range of complex activities, beyond the clearance of infections. An analysis of their behaviour as observed through intravital microscopy could help reveal more of their roles.

Finally, there is accumulating evidence to support the existence of different lineages of neutrophils with discrete roles. It is unknown whether these are actually distinct lineages or if they instead all develop from the same neutrophil predecessor.

The observation of neutrophils requires the analysis of hundreds of frames of microscopy image sequences. The manual annotation of these image sequences to extract trajectories of movement of neutrophils is a time consuming and error prone process. Manual annotation severely limits the number of data that can be analysed and slows down the advancement of cell research. I would be a major advance to be able to reliably automatically identify and track cells over time in sometimes noisy complex images.

1.2 Objectives DRAFT I

Neutrophil image sequences obtained *in vivo* are sometimes of low contrast. *In vivo* observation of these cells in the lung is especially difficult due to the motion artefacts. The motion of the lung can also cause the images to be out-of-focus, which causes the cells to appear blurred and become difficult to identify and segment.

The aim of this research is to develop methods for cell detection and tracking. This system should be able to accept an image sequences of cells in tissue, identify the cells and track them over the entire sequence.

The identification of cells should take into account the nature of the imaging technique and the quality of obtained images. Simple methods such as thresholding would be too unreliable; more advanced methods need to be investigated.

Also the tracking of cells should take into account the nature of input data. Basic frame-by-frame tracking of cells is likely to have poor results because the cells frequently disappear into the depths of tissue or loose focus. Methods that take into account the temporal behaviour of cells are likely to result in more robust tracking.

1.3 Contributions DRAFT I

The work in this project led to a development of a robust pipeline for automatic cell tracking. The pipeline combines a cell detection algorithm from [4] and, to the best knowledge of the author, a novel tracking method that is directly based on the observed data.

The cell tracking module uses a global data association approach to reliably generate cell detection trajectories based on a global decision. This research has upgraded previous methods to rely on a large set of features obtained from observed data. A machine learning approach is used to compute likelihoods for linking cell detections into increasingly longer trajectories. Because the likelihoods are obtained directly from training examples, the method is able to produce good results also on noisy datasets, where several frames in a sequence can come out-of-focus, cells disappear and reappear over time, etc.

Finally, an image annotation tool is developed that allows simple dot annotation of cells and linking of cells among frames to represent trajectories.

1.4 Report structure DRAFT I

The rest of the thesis is structured as follows:

Chapter 2 is a brief literature survey outlining existing methods for cell detections and tracking. The chapter gives a reasonable background of how the methods for cell detection and tracking have evolved over time.

Chapter 3 describes in more depth the cell detector from Arteta et al [4] and outlines the modification that improved the detection speed.

Chapter 4 describes in detail the cell tracking module. It outlines the two step process of generating robust tracklets and then joining them into long trajectories using trained classifiers.

Chapter 5 is an overview of example datasets on which the tracking tool is tested and presents a cell annotation tool developed to ease the annotation of image sequences.

Chapter 6 evaluates the performance of the cell tracking module.

Add something more specific

Chapter 7 includes some concluding remarks and ideas that could be implemented to continue to advance the field of automatic cell detection and tracking.

2 Related work DRAFT I

This chapter is a survey of methods for cell detection and cell tracking. It summarizes the different techniques in a structured way, and discusses the advantages and disadvantages of each method. The chapter is divided into three sections. Section 2.1 describes methods to perform cell detection on images containing cells. Section 2.2 presents methods used to track cells in a sequence of images. Finally, in section 2.3 we describe which methods seem most promising for our application of cell tracking.

2.1 Cell detection DRAFT I

Cell detection consists in identifying individual cells in microscopy images. Due to the different imaging techniques and the types of cells we may wish to detect, several algorithms have been developed each to handle a specific case. This section is an overview of these techniques.

2.1.1 Cell segmentation using the Watershed technique DRAFT I

A basic cell detection method relies in binarizing an image to separate the background from the cells, followed by a segmentation step to extract the cells. Chen, Biddell et al [5] use a flooding approach for segmentation. The entire method can be summarized as follows. First, a spatial adapter filter eliminates some noise in the image. Second, it locates the pixels with minimum intensities in a small sliding window. Third, for each minimum point it proceeds to the progressive flooding of its neighbouring points and a final post-processing step discards false regions.

A similar method by Chen et al [6] consists on using a Watershed algorithm [7] to separate cell nuclei after using Otsu's thresholding method to segment nuclei from the background. Morphological operations [8] can be used to fill holes and eliminate patches that are too small to correspond to cells. Additionally they developed a nuclei-fragment merging method based on the shapes and sizes of the nuclei to deal with the problem of over-segmentation caused by the Watershed algorithm.

The disadvantage of these methods is that the number of pixels belonging to either cells or background should be approximately the same, and that the signal-to-noise ratio should be high.

2.1.2 Cell segmentation using level sets DRAFT I

Another interesting technique to segment cells is a contour evolution method that makes use of the general appearance of cells to segment them using level sets. Mukherjee et al [9] make the observation that leukocyte shapes are nearly circular and that at least for a significant part of the border of the cells, the intensity profile is different from the cell cytoplasm and from the background. Using this observation, identification of a leukocyte is formulated as a minimization of an energy function incorporating image gradient and intensity homogeneity within the closed contour encompassing the cell. The benefit of this method is that it can be adjusted to perform well in images with significant clutter and poor contrast by increasing the importance of the homogeneity, or for images with good contrast, where the gradient magnitude term is given more importance. The disadvantage of this method is that cells cannot overlap. The energy function can be minimized with the gradient descent method.

To reduce the solution space for the energy function only the boundaries of connected components within the image-levels sets are evaluated with the energy function. Only the connected components satisfying the size and shape constraints of the cells are extracted. The remaining components are eliminated using area morphology operations. This level-set analysis provides a more efficient solution that is linear in the number of intensity levels in the image in contrast to the much higher complexity of a curve evolution method.

The level set method is contour evolution approach which has good results in segmentation. Tang et al [10] have successfully combined level-sets and local grey thresholding [11] for segmenting neuron stem cells in images which have been obtained by confocal microscopy.

2.1.3 Cell detection by model learning DRAFT I

This method can be shortened just a bit, since I describe the details later

The previous methods perform efficiently in cells with sufficiently good contrast. In images where the cell borders are unclear, images are of varying intensity, cell density is high, or cells can be of different shapes, these methods may not perform as well. In such cases machine learning methods can perform better by learning a model of a cell based on a training set of annotated examples.

Arteta et al [4] [12], propose an algorithm that uses a highly-efficient MSER region detector [13] to find a broad number of candidate regions. These regions are then scored depending on the similarity to the cell type of interest by a machine learning algorithm.

The authors organized the extremal regions obtained from the MSER detector into trees, so that each tree corresponds to a set of overlapping extremal regions. Each such region is given a value corresponding to the similarity of the region to a cell. The non-overlapping regions which achieve high scores can then be selected using dynamic programming of the trees. The learning is performed using a structured SVM [14] which is able to take into account the non-overlap constraint, and

achieves good performance. The learning is performed on weakly annotated images – a single dot is necessary on each cell.

The advantage of this approach is the tolerance to changes in image intensities, cell densities and sizes. The major downside is the non-overlap constraint. Fortunately the authors have also developed an algorithm to detect partially overlapping cells [12].

The idea is to learn to detect overlapping cells together with the number of cells in the region. The algorithm starts by generating a set of nested regions. Each region is then scored using a set of classifiers that evaluate the similarity of the region to each of the possible classes, where each class corresponds to the number of cells that the region contains. An inference procedure then selects the non-overlapping subset of regions, and assigns each a class label indicating the number of cells that the model believes lie in the region. This eliminates the non-overlap constraint, but requires a larger training set of annotated images to train the detector.

2.1.4 Cell detection by image restoration DRAFT I

Another approach is to use an image restoration technique followed by thresholding [15] [16]. Bise et al [15] apply this technique on phase-contrast microscopy images . The technique utilizes the optophysical principle of image formation by phase-contrast microscope to transform the image into an artefact free image by minimizing a regularized quadratic cost function. A simple image thresholding technique can then be used for segmentation.

2.2 Cell tracking DRAFT I

After detecting each cell in each image of the sequence we need to determine which cells in an image correspond to which in the next image. This way we can generate tracks of the cells across the sequence. This is the problem of tracking. We have identified four main approaches to tracking:

Tracking by model evolution Active contours or level sets are used to detect a cell in the first frame of the sequence, and the cell models are then propagated to the next frame.

Tracking by data association Cells in consecutive frames are associated according to a similarity measure, which compares features such as the cell intensity and the relative spatial location.

Tracking with a dynamics filter This method uses a dynamics filter (e.g. Kalman or Particle filter) to predict the spatial location of the cell in the following frames.

Tracking by global data association This method approaches tracking as a global optimization problem and all trajectories are generated at once. These methods aim to maximize the probabilities that the generated trajectories are correct.

The remaining of this chapter briefly describes how these methods have been implemented in different papers.

2.2.1 Tracking by model evolution DRAFT I

This approach consists of identifying a cell in the first frame of a sequence by means of active contours or level sets and then propagating the models to the next frame. Model evolution tracking methods handle cell deformation well, but can often get stuck in local minima. This method can be computationally expensive because of the need to evolve the cell models for each frame. It is possible to improve the tracking accuracy at the expense of computation time.

Mukherjee et al [9] model the problem of cell tracking as maximizing a similarity measure between level sets in consecutive frames. The method minimizes the energy associated with leukocyte boundary detection and then matches the boundary with that of the previous frame. To maintain spatial coherence, the authors assume that the displacement of cells between frames is marginal. Such an automatic tracker is able to handle slight rotations or deformations of the leukocytes well.

2.2.2 Tracking by frame-by-frame data association DRAFT I

When tracking by frame-by-frame data association, cells in consecutive frames are associated according to a similarity measure, which compares features such as the cell intensity or relative spatial location. This method is very efficient, but only effective when cells are accurately detected. It is possible to improve the association by analysing several frames in a sequence.

Chen et al [6] perform the matching process based on the distances between the cells. A match is found if a feature distance is below a threshold. The authors have also developed strategies to overcome the problem of false matched and ambiguous correspondence when nuclei are touching or partially overlapping. In this case information about these nuclei is stored (nuclei size and their relative location – up, down, left, right –). When they separate, the algorithm can resolve the ambiguities by comparing their current status with the previously stored information.

To reduce the vulnerability of frame-by-frame data association it is possible to analyse several frames of the sequence. This is the approach by Huh [16]. For each new frame, a set of hypothesis is computed for each detected blob, corresponding to migration, exit, entrance, clustering and mitosis. After all hypothesis are obtained, a best combination is selected by formulating and solving an integer programming problem. If, after the best association is found, there are remaining cells, these are considered again in the following frames.

The similarity function used to perform data association between consecutive frames is of crucial importance. House et al [17] propose a novel cost function that encodes the response of cells to conditions of their environment. The tracking problem is formulated as a standard Bayesian filtering problem. The model allows the state of the cells in the images to evolve in time. To solve the complete assignment they have used a probabilistic data association algorithm which produces an

optimal solution.

2.2.3 Tracking with a dynamics filter DRAFT I

Shorten further

Dynamics filters are a powerful addition to tracking systems because they predict the motion of a cell, and reduce the search space in which we look for matches. These algorithms not only perform frame-by-frame tracking, but also take temporal information into account, resulting in models that are robust to long-term occlusion and cell detection error.

Xu et al [18] developed an ant stochastic searching behaviour based tracking system to track multiple cells in fluorescent image sequences. The system is similar to particle filters, but the motion behaviour is modelled to be similar to how ants behave when searching for food. When ants move around, they deposit small quantities of chemical pheromone. Once they find food, they retrace the steps deposing larger amounts of pheromone. Other ants can then use this guide to find food faster, but are allowed to deviate from the path and pursue their own paths. The author propose a system where the food is not static, so that it can be used for cell tracking. Compared to particle filters, this algorithm is more computationally expensive, but achieves better accuracy. The algorithm is able to deal with cells entering or leaving the scene, and occlusion.

Kalman filters are another popular method for state prediction. Tang et al [10] use a Kalman filter to guide the tracking of active cells. Active cells are those that move a distance larger than a threshold (supposedly their radius) between frames. Inactive cells are tracked separately by observing their overlap region.

Cells do not always exhibit just one type of motion and a single Kalman filter cannot take this into account. Li et al [19] propose an automated tracking system which runs several Kalman filters in parallel, each corresponding to a different modality: random walk, first-order and second-order linear extrapolations which correspond to Brownian motion, migration with constant speed and migration with constant acceleration. The Kalman filters are run by the interacting multiple models (IMM) filter. The transitions between models is determined by a finite state Markov chain. Their system is composed of five modules: cell detector, cell tracker, dynamics filter, track compiler and track linker. The cell detector separates the background from cell pixels. The cell tracker propagates the cell labelling to the next frame in the sequence. The track compiler compares the results of the cell tracker, cell detector and dynamics filter to create a new track for new or daughter cells, and updates or terminates existing tracks. The track linker connects two or more track segments if they belong to the same cell.

2.2.4 Cell tracking by global data association DRAFT I

The benefit of global data association approaches is that they aggregate the results of all frames and make a global decision rather than propagating the results of each frame to the next. This makes

them less dependent on errors from the cell detection module.

Massoudi et al [20] propose a tracking method that does not rely on perfect segmentation and can deal with uncertainties by exploiting temporal information and aggregating the results of many frames. The system only requires probabilities or potentials that represent cell positions. Tracking is modelled as a network flow problem and is formulated as a linear program based on the occupancy likelihood of the edges of the network. The system can detect false positives or false negatives and correct itself. The method handles division events and cells entering or exiting the screen at the boundaries.

Zhang et al [21] also researched tracking of pedestrians using network flows and achieved superior performance to frame-by-frame methods. In their approach, data association is defined as a maximum-a-posteriori (MAP) estimation problem given a set of object detections results as input observations. The flow paths in the network correspond to non-overlapping trajectory hypothesis, and the flow costs to the observation likelihoods and transition probabilities. The global association is found by a min-cost flow algorithm.

An equivalent MAP problem is defined by Huang et al [22] and Bise et al [23]. First, they both generate reliable tracklets by linking cell detections responses in consecutive frames. Second, the short tracklets are associated into longer and longer tracklets. This second, global data association approach is solved by Huang et al iteratively using the Hungarian association algorithm [24] and by Bise at al as an integer optimization problem. This approach can achieve state-of-the-art performance and surpass the accuracy of previously discussed methods.

2.3 Conclusion DRAFT I

This chapter presented an overview of methods used for cell detection and tracking. Some of these methods are designed to perform better under specific imaging and cell conditions. The combination of thresholding and the Watershed method for segmentation is likely to perform well in high quality images where the cells are clearly discernible. However, in the case of noisy images with varying contrast, the machine learning approach presented by Arteta et al is likely to perform much better, at the cost of computation time to compute the larger number of features for each candidate cell region. Both these methods have been briefly evaluated on the noisy image sequences that we study in this research. The first method required a lot of manual adjustments to perform acceptably well. The second method was able to reliable detect cells in the low signal-to-noise images. For this reason, the machine learning method by Arteta et al has been chosen for this application. A more in depth explanation of the method is presented in chapter 3.

Although the detection method is able to reliable detect most of the cells in the image sequences, the cells sometimes cannot be detected due to several factors. First, the cells can submerge into the depth of the tissue and reappear a few frames later. Second, due to the motion artefacts caused by the moving (lung) tissue when the images are obtained, the images can be out-of-focus for several frames. These and other artefacts make the cell tracking problem much harder. Therefore, we needed

a method that would be able to handle false detections and missing observations over a few frames. A global data association method heavily inspired by [23] has been developed and is presented in chapter 4.

There are several reasons why automatic cell tracking systems are not as popular as we would expect. The main reason is probably that the level of accuracy falls short of manual annotation. Furthermore, these systems are often not robust to changes in cell type, cell culture condition and microscopy image quality. It is our hope that the state-of-the-art methods we have chosen and will develop for our tracking application will enable us to track cells in noisy images with a performance comparable to that of a human.

3 Detection of cells DRAFT I

3.1 Method overview DRAFT I

In order to track cells across frames we need to first be able to identify the cells within the images. As described in section 2.1, there is a broad range of automatic methods for cell detection and segmentation.

The requirements for detecting cells in our application are:

1. Accurate detection in sometimes noisy images. The detector should be able to robustly identify cells in datasets that include noise, motion artefacts (induced by the camera and the moving tissue) and variable lens focus. The detector does not need to detect cells that are out of focus, but should reliable detect those are in focus. Out of focus frames are dealt with in the tracking module.

2. There is no need for accurate segmentation. We are primarily interested in the accurate tracking of cells, and analysis of the cells appearance is of secondary interest. For the purpose of tracking, it is sufficient to be able to reliable localize cells, and extract some features that identify these cells.

3. The algorithm should adapt well to different microscopy modalities. The studied datasets are obtained from various organs within the body of mice that have distinct textures. The detector should be able to detect cells in any of these datasets with minimal manual adjustments.

Much of the previous work described in section 2.1 was focused on accurate segmentations of cells. Many of the described methods would return very good segmentations, but they would fail when presented with a very noisy dataset of varying contrast, or need major manual tuning of parameters when presented with a new dataset with a different kind of cells.

For the purpose of our application we have chosen the cell detector presented in [4], because it conforms to all three requirements. The machine learning method is able to handle noisy datasets, and the model can easily be retrained when a new, previously unseen, type of images needs to be analysed. The method does not focus on accurate segmentation of cells, but on the robust localization. The major drawback of this method was performance. The authors reported run-times of 30 seconds to detect cells on 400-by-400 pixel images on an i7 CPU. This slow performance would make it unusable for tracking large image sequences. However, we were able to optimize the MATLAB code for feature computation so that the detection process takes only a fraction of the

original run-time.

The detection process runs in three steps. First, robust candidate regions are extracted from an image. The method uses an efficient MSER region detector [25] to find a large number of candidate regions. Second, each of these regions is assigned a value which is produced by a classifier, which indicates the appropriateness score that this this region is a cell. Finally, the non-overlapping subset of these candidate regions with high similarity to the annotated cells in the training data is selected via dynamic programming.

The detector can be trained on a small number of training images, where each cell is annotated with a single dot.

The code used in our method is based on the original code from [4]. Some of our modifications, especially in the feature computations, allow the detector to run significantly faster. Additional work was required to combine the cell detector with the tracking module described in chapter 4.

The remained of this chapter is divided as follows. In section 3.2 we describe how the candidate regions are obtained. Section 3.3 shows the inference procedures that selects the optimal subsets of candidate regions. In section 3.4 we formulate the learning method and in we describe the choice of features in section 3.5. Finally, in section 3.6 we briefly outline the main changes that improved the performance of the original algorithm.

3.2 Detection of candidate regions DRAFT I

region

y stable region

The first stage of the cell detection process is the extraction of a large number of cell candidate regions. A region is extremal if the image intensity everywhere inside of it is highter than the image intensity at its outer boundary. For a grayscale image I it is the set of connected components of the thresholded image $\mathcal{I}_{>\mathrm{t}} = \{\mathcal{I} > \mathrm{t}\}$ for some threshold t. In practice we consider only regions that are maximally stable, as defined in [25], obtained using an efficient maximally stable region detector (MSER).

An important property of extremal regions is *nestedness*, which means that two extremal regions detected on the same image can be either nested or non-overlapping.

The number of obtained candidate regions is high, and an inference step determines which belong to cells.

Extremal regions are invariable to affine transformations of image intensities. They are stable, which means that their support is virtually unchanged over a range of thresholds, they can be detected at multiple scales and can be enumerated very efficiently. This makes them ideal for detecting candidates regions that correspond to cells in microscopy imaging. Figure 3.1 shows an example result of running the MSER detector on a sample set of microscopy cell images.

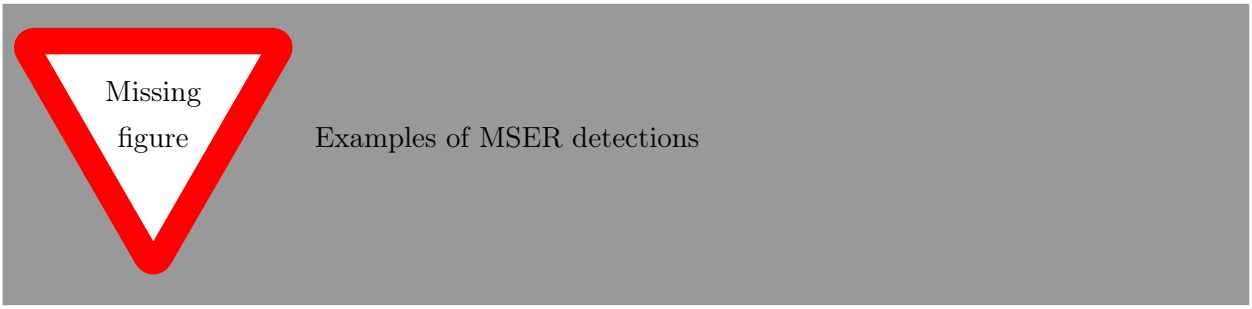

Figure 3.1: Mser detection examples

3.3 Inference under the non-overlap constraint DRAFT I

The inference procedure selects a subset of candidate extremal regions that corresponds to cells. Let V_i be a value assigned to each candidate region R_i indicating the appropriateness of the region to correspond to a cell. The inference procedures selects a subset of the non-overlapping candidate regions such that the sum of the values of selected regions is maximized. To make the problem computationally feasible, the regions are organized into tress according to the nestedness property. Each tree represents a set of overlapping extremal regions. The problem can then be formalized as followes. Let $\mathbf{y} = \{y_1, y_2, \dots y_N\}$ be binary variable such that $y_i = 1$ if the region R_i is selected. Let $mathcalY$ be the set of non-overlapping regions. The maximization problem can be defined as:

$$F(\mathbf{y}) = \max_{\mathbf{y} \in \mathcal{Y}} \sum_{i=1}^{N} y_i V_i.$$

Further details on how the exact solution is obtained can be found in [4].

Although cells in the studied datasets occasionally overlap, detecting only non-overlapping cells is a good balance between manual work required to annotate datasets, training time and accuracy of the detections. The described framework has been expanded to learn to detect partially overlapping regions [12]. However, that model requires a longer training time, and a larger set of manually annotated images to accurately detect blobs of overlapping cells. Finally, the global data association technique used in the cell tracker module should be able to correctly associate cells if the overlap time is short, such as in trajectories that are crossing and then continuing in opposite directions.

3.4 Learning the classifier DRAFT I

The value V_i assigned to each region R_i is a classifier score that indicates the similarity of the extremal region to a cell. The classifier is trained using dot-annotated training images $\mathcal{I}^1, \mathcal{I}^2, \mathcal{I}^3, \dots$ as follows. Let $R_1, R_2, \dots, R^j_{N^j}$ indicate the set of N^j candidate regions extracted from the training image $\mathcal{I}^j$. The value V_i^j of each region is computed as the dot product between a feature vector $\mathbf{f}_i^j$ computed for each region and a weight vector w: $V_i^j = \mathbf{f}_i^j \cdot w$.

The learning procedure has to learn the weight vector w so that the inference procedure tends to pick regions with a single annotated cell $n_i^j = 1$. In order to consider the non-overlap constraint of regions a structured SVM [26] is used that directly optimizes the performance of the inference on the training set. The loss function that the learning framework tries to optimize counts the number of deviations from the one-to-one correspondence between the annotated dots and the selected regions:

$$L(\mathbf{y}^j) = \sum_{i=1}^{N^j} y_i^j |n_i^j - 1| + U^j(\mathbf{y}^j),$$

where n_i^j indicates the number of annotated dots in region R_i^j and $U^j(\mathbf{y}^j)$ counts the number of annotated dots that do not have a corresponding selected region R_i^j. Further details about the convex objective we try to minimize and how it can be optimized using a cutting-plain algorithm can be found in [4].

3.5 Feature selection DRAFT I

The effictiveness of the machine learning method to detect cells depends on the quality of features computed for the regions. Good features have a lot of discriminative power, so that they can effectively distinguish regions corresponding to cells to regions that don't correspond to cells. In order to find the best feature vector we have evaluated the performance of the classifier with several combinations of these features:

1. The area A of the region represented by a 10-dimensional binary vector with the entry $\lceil logA \rceil$ set to 1.

2. 10-dimensional histogram of intensities within the region.

3. The position of the descriptor in the image in terms of x-y coordinates of a centroid fitted to the descriptor.

4. Two 6-dimensional histograms of differences in intensities between the region border and a dilation of it for two different dilation radii.

5. A shape descriptor represented by a 60-dimensional histogram of the distribution of the boundary of the region on a size-normalized polar coordinate system.

6. The orientation of the descriptor after attempting to normalize its orientation.

7. The proportion of edge-pixels in the region.

Each of these features has different discriminative power, and takes some time to compute. The application of the cell detector requires that images are processed within a time limit. For this reason, we have trained and tested the algorithm with all the $2^7 - 1$ possible combinations of these

features.

We have also developed a tool that helps select the most appropriate set of features given specific constraints, for example a maximum computation time, minimal precision and recall values, etc. A graph generated by the function is shown in figure 3.2.

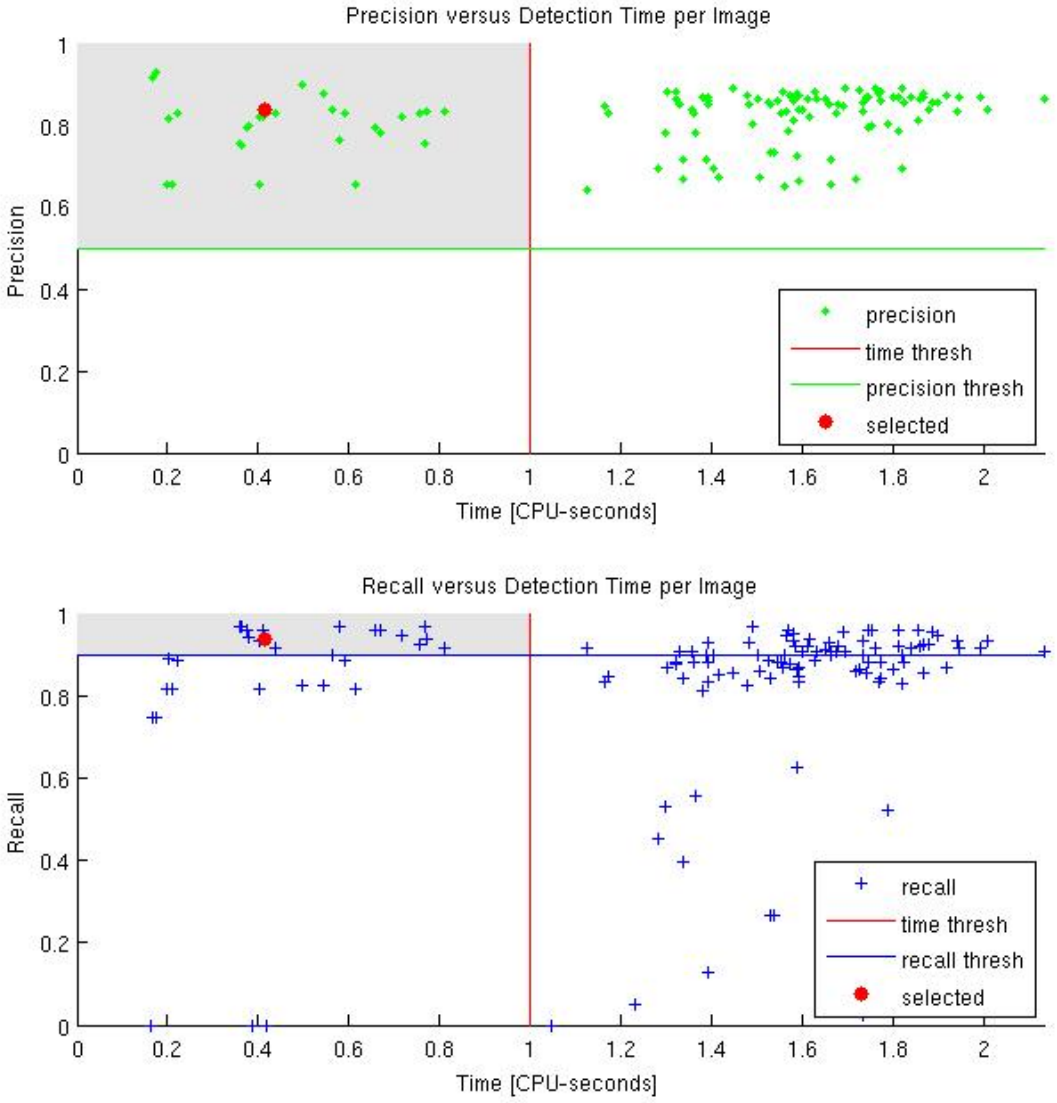

Figure 3.2: The plots helps the user to select an appropriate feature set according to the computation time per image, mean precision and recall. This example filters features sets that compute within 1 CPU-second per image (on average), have at least 0.5 precision and 0.9 recall. The feature set corresponding to the red dots contains features 1, 3 and 4. Most importance was given to high precision followed by low computation time and high recall. The selected feature set computes in about 0.4 CPU-seconds per 512-by-512 pixel image with mean precision of 0.836 and mean recall of 0.9363 (as measured on an i7-2600 CPU wish a clock frequency of 3.40 GHz).

3.6 Performance improvements DRAFT I

The main drawback of the algorithm presented by Arteta [4] is the slow feature computation. The original algorithm took about 30 seconds to detect cells in a 400-by-400 pixel image. Because the aim of this project is tracking cells, we expect to be processing hundreds or thousands of frames of microscopy image sequences. It was important to reduce the detection time as much as possible. The major performance improvements where achieved by addressing three things.

The algorithms computes the features for the classifier on every candidate maximally extremal region. The original method was configured such that thousands of MSERs were computed for each frame. While this provides more robust learning and detection, it was slowing the algorithm down excessively. First, we have first fine-tuned the MSER detector to detect less regions, but still return a large number of features such that the detection rate wasn't penalized.

Second, we have identified the slowest computed features and improved the algorithmic behaviour of the original MATLAB implementation. One such feature is the Contour Points Distribution Histogram. The function was performing many calls to slow built in functions to extract region characteristics. This, and other functions, were rewritten to call these expensive sub-procedures less frequently, without affecting their behaviour.

Third, we noticed that the MATLAB built-in function were performing a lot of parameter checking, to make sure that the input parameters were valid. Many of these parameters were strings, and string manipulations are very slow in MATLAB. Several MATLAB built-in function were rewritten or modified to remove excessive parameter checking, which in several cases represented an overhead of over 30%. These parameter checks are welcome when developing the algorithm, but once the algorithm is complete, several of these checks can be safely removed.

It is also worth mentioning that the updated original code for the detector is now stabler and better tested. A few rare bugs have been fixed, after being found when running the detector over large number of datasets.

These optimizations resulted in a significant performance boost. The updated algorithm can perform the same computations in a fraction of the time taken by the original implementation provided by the authors of [4].

Appendices

Bibliography

[1] P. K. Elzbieta Kolaczkowska, "Neutrophil recruitment and function in health and inflammation," 2013. 7

[2] J. Pillay, I. den Braber, N. Vrisekoop, L. M. Kwast, R. J. de Boer, J. A. M. Borghans, K. Tesselaar, and L. Koenderman, "In vivo labeling with 2h2o reveals a human neutrophil lifespan of 5.4 days," *Blood*, vol. 116, no. 4, pp. 625–627, 2010. 7

[3] P. S. Tofts, T. Chevassut, M. Cutajar, N. G. Dowell, and A. M. Peters, "Doubts concerning the recently reported human neutrophil lifespan of 5.4 days," *Blood*, vol. 117, no. 22, pp. 6050–6052, 2011. 7

[4] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect cells using non-extremal regions," in *Proceedings of the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I*, MICCAI'12, (Berlin, Heidelberg), pp. 348–356, Springer-Verlag, 2012. 8, 9, 11, 17, 18, 19, 20, 22, 47

[5] Y. Chen, K. Biddell, A. Sun, P. Relue, and J. Johnson, "An automatic cell counting method for optical images," in *[Engineering in Medicine and Biology, 1999. 21st Annual Conference and the 1999 Annual Fall Meetring of the Biomedical Engineering Society] BMES/EMBS Conference, 1999. Proceedings of the First Joint*, vol. 2, pp. 819 vol.2–, Oct 1999. 10

[6] X. Chen, X. Zhou, and S.-C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *Biomedical Engineering, IEEE Transactions on*, vol. 53, pp. 762–766, April 2006. 10, 13

[7] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *Image Processing, IEEE Transactions on*, vol. 2, pp. 176–201, Apr 1993. 10

[8] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983. 10

[9] D. Mukherjee, N. Ray, and S. Acton, "Level set analysis for leukocyte detection and tracking," *Image Processing, IEEE Transactions on*, vol. 13, pp. 562–572, April 2004. 11, 13

[10] C. Tang, Y. Wang, and Y. Cui, "Tracking of active cells based on kalman filter in time lapse of image sequences of neuron stem cells." 11, 14

[11] D. Xu and L. Ma., "Segmentation of image sequences of neuron stem cells based on level-set

algorithm combined with local gray threshold.," Master's thesis, Harbin Engineering University, 2010. 11

[12] C. Arteta, V. S. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect partially overlapping instances.," in *CVPR*, pp. 3230–3237, IEEE, 2013. 11, 12, 19

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference*, pp. 36.1–36.10, BMVA Press, 2002. doi:10.5244/C.16.36. 11

[14] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Mach. Learn.*, vol. 77, pp. 27–59, Oct. 2009. 11

[15] R. Bise, T. Kanade, Z. Yin, and S. il Huh, "Automatic cell tracking applied to analysis of cell migration in wound healing assay," in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pp. 6174–6179, Aug 2011. 12, 24

[16] S. Huh, *Toward an Automated System for the Analysis of Cell Behavior: Cellular Event Detection and Cell Tracking in Time-lapse Live Cell Microscopy.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2013. 12, 13

[17] D. House, M. Walker, Z. Wu, J. Wong, and M. Betke, "Tracking of cell populations to understand their spatio-temporal behavior in response to physical stimuli," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 186–193, June 2009. 13

[18] B. Xu, M. Lu, P. Zhu, Q. Chen, and X. Wang, "Multiple cell tracking using ant estimator," in *Control, Automation and Information Sciences (ICCAIS), 2012 International Conference on*, pp. 13–17, Nov 2012. 14

[19] K. Li and T. Kanade, "Cell population tracking and lineage construction using multiple-model dynamics filters and spatiotemporal optimization," in *Proceedings of the 2nd International Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB)*, September 2007. 14

[20] A. Massoudi, D. Semenovich, and A. Sowmya, "Cell tracking and mitosis detection using splitting flow networks in phase-contrast imaging," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 5310–5313, Aug 2012. 15

[21] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008. 15, 26

[22] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Computer Vision - ECCV 2008* (D. Forsyth, P. Torr, and A. Zisserman,

eds.), vol. 5303 of *Lecture Notes in Computer Science*, pp. 788–801, Springer Berlin Heidelberg, 2008. 15, 26

[23] R. Bise, Z. Yin, and T. Kanade, "Reliable cell tracking by global data association.," in *ISBI*, pp. 1004–1010, IEEE, 2011. 15, 16, 26, 29, 47

[24] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955. 15

[25] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004. British Machine Vision Computing 2002. 18

[26] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, (New York, NY, USA), pp. 104–, ACM, 2004. 20