Imperial College London
Department of Computing

Automatic Cell Tracking in Noisy Images for Microscopic Image Analysis

Pedro Damian Kostelec

September 2014

Supervised by Ben Glocker

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

Abstract

Acknowledgement

I offer my sincerest gratitude to life, ...

Contents

1 Introduction NEW

1.1 Motivation NEW

To develop methods for cell tracking

explain what is a the neukocyte
what is its function
why is it still important to track them (what we don't know)
section todo describes the imaging modality

1.2 Objectives NEW

Describe the different modules of the system - detector, tracker, stat computation

- create a robust pipeline for tracking neukocytes in image sequences of noisy images obtained throught that ventilator method

- the system should be able to accept an image sequence, detect the cell in each frame, and uset hem to compute the trajectories of the cells.

- A basic system to return metrics about the trajectories

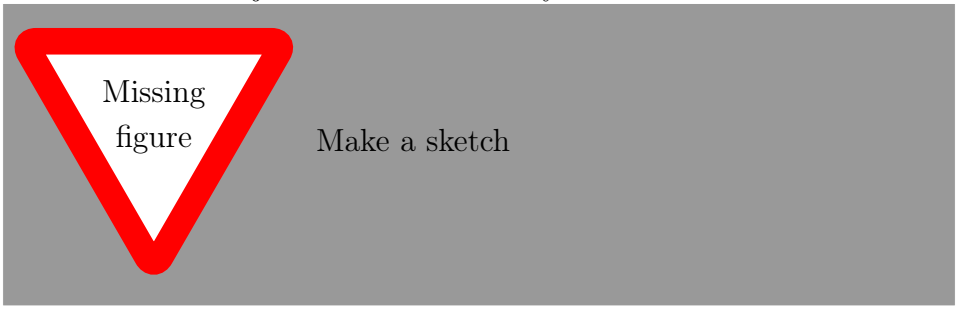
1.3 Contributions NEW

Machine learning based cell tracker system

that is able to work on highly noisy datasets, where several frames can come out of focus, cells dissapear and reapear, etc.

Dot Annotation GUI

to ease the annotation of datasets of point objects, as well as link them to create trajectories in a frame by frame basis



1.4 Thesis structure NEW

The remained of the thesis is structured as follows.

chapter 2 is a brief literature survey outlining the exesting methods for cell detection and tracking

chapter 3 describes the cell detection module and its implementation

chapter 4 describes the tracking module

chaapter 5 represents the biological statistics module, and we focus on listing the different types of metrics that it computes

chapter 6 represents the image annotation tool that was created to facilitate image annoations

chapter 7 evaluates the cell detector and cell tracker

chapter 8 is reserved for some concluding remarks and ideas that could be used to enhance this work.

3 Detection of cells NEW

3.1 Overview NEW

3.2 Learning to detect non-overlapping cells NEW

Check both Arteta's papers on the subject

Subsection with the changes that were made to Arteta's work

Rewrite: Several of these subsections are directly taken from Arteta's paper. They are mean as guidance, but should be adapted appropriately.

3.2.1 Extremal regions NEW

3.2.2 The non-overlap constraint NEW

3.2.3 Formulation of the model NEW

3.2.4 Learning the model NEW

Info about the binary/structural classifier

3.2.5 Implementation details NEW

Dynamic programming for inference NEW

3.2.6 Summary NEW

3.3 Speeding up the algorithm NEW

review this section

The main drawback of the algorithm presented by Arteta [1] is the poor performance. The original algorithm took about 30 seconds to detect cells in a 400x400 image. Because we will be processing hours of microscopy video it was important to reduce the detection time as much as possible. The major performance improvements where achieved by addressing three things.

The algorithms needs to extract a set of feature on every single detecetd MSER. First, we have first fine-tuned the MSER detector to detect less cells, more robustly.

Second, we have identified features that are slow to compute and improved their algorithmic behaviour. One such feature is the Contour Points Distribution Histogram implemented in *cpdh.m*. The function was performing excessive calls to slow functions to extract region characteristics, and was rewritten to call these function less often, while negligibely affecting its optimality.

Second, several MATLAB builtin function were 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.

These opimizations resulted in a significant performance boost. Instead of 30 seconds, the algorithm can now detect cells on the same images in about .

Measure the number of seconds it takes to process one of these 400x400 images

3.4 Feature selection NEW

The effictiveness of the machine learning method to detect cells depends on the quality of features. Good features have a lot of discriminative power between cells and non cells. The used approach classifies extremal regions as cell/non-cell. The regions are extracted using the MSER detector. Then each region is processed and features are extracted from it. We have evaluated 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-dimensioal 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 function that helps select the most appropriate set of features given specific constraints, for example a maximum

computation time, minimal precisiona and recall values, etc. A graph generated by the function is shown in figure 3.1.

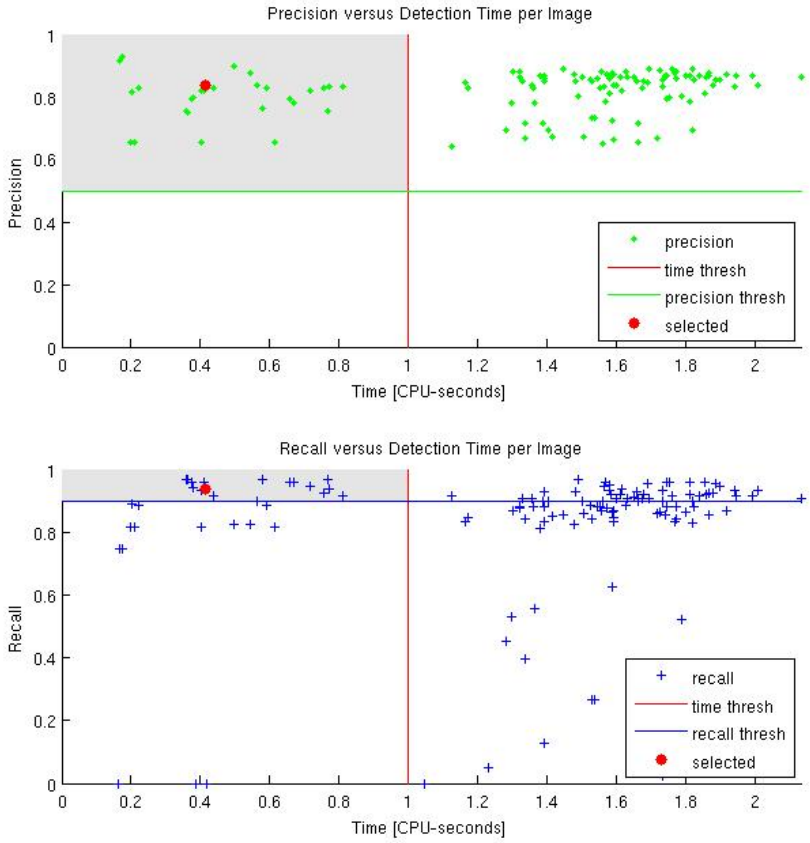


Figure 3.1: The plots helps the user to select the most appropriate feature, given a combination of computation time per image, mean precision and recall. This example, which was obtained by selecting only within feature sets that compute in less than 1 CPU-second, have at least 0.5 precision and 0.9 recall have resultsed in a feature set containing features 1, 3 and 4. Most importance was given to precision followed by computation time and recall. The selected feature set computes in 0.414 cpu-seconds (Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz) per image, with mean precision of 0.836 and mean recall of 0.9363.

Bibliography

[1] 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. 14, 20

[2] 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. 18

[3] R. Bise, Z. Yin, and T. Kanade, "Reliable cell tracking by global data association.," in *ISBI*, pp. 1004–1010, IEEE, 2011. 21, 25

[4] 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. 21

[5] 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. 21

[6] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak,

and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9(7), pp. 676–682, 2012. 20

[7] S. F. I. o. T. L. Philippe Thévenaz, Biomedical Imaging Group, "Point picker: An interactive imagej plugin that allows storage and retrieval of a collection of landmarks," May 2014. 20