

# Segmentation of Individual Cells in Phase Contrast Microscopy Images

Jindřich Soukup<sup>1,2,3</sup>  
soukupj@utia.cas.cz

Michal Lašan<sup>2</sup>

Filip Šroubek<sup>1</sup>

<sup>1</sup> Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic

<sup>2</sup> Faculty of Mathematics and Physics, Charles University in Prague

<sup>3</sup> Institute of complex systems, FFPS, University of South Bohemia in Ceske Budejovice

---

## Abstract

Phase contrast microscopy is widely used for live cell imaging. Automatic computer processing of phase-contrast images is complicated, particularly because of the halos that are present around the cells. We introduce a new algorithm able to segment individual cells by using the halos. The halos between cells are detected by thresholding and skeletonization, and gaps in the edges are filled in using Dijkstra's algorithm. The result of the algorithm was compared with the results of manual segmentation performed by a human expert and evaluated on level of precision of individual cells' edges. The source code and test data, including the manual segmentation are made available to the public.

## 1 Introduction

Nowadays, computer processing of microscopic images is becoming more common as a part of research in the field of cell science, replacing slow and time-consuming manual processing by human experts. A challenging task is for example to process images of live mammalian cancer cells from a phase contrast microscope. The cells have irregular shape, appear semi-transparent in the images (their inner structure is visible), and the colour of their inner parts is quite similar to the background colour. A characteristic feature of this microscopy method are light areas around the cells, so called halos (see Fig. 1). All this complicates the automatic computer processing of such images.

When processing the time-lapse images of live cells, the main goal is to describe the movement and behaviour of individual cells. From the image processing point of view, this means segmenting individual cells.

We distinguish two categories of segmentation methods that process phase contrast microscopy images: Methods [1,2] that separate background from contiguous cells, and methods [3,4,5,6] that segment individual cells from colonies (densely packed cells touching each other) or even from cell monolayers (cells create a contiguous layer) which is much more difficult.

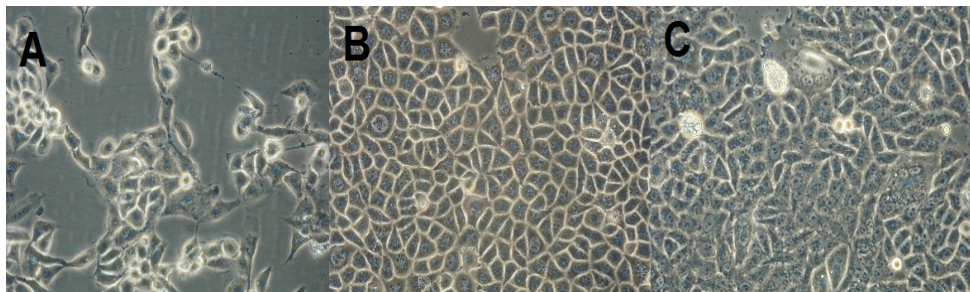


Figure 1: Phase contrast microscopy images of A) vero cells E6, B) L929 mouse fibroblast, C) HeLa cervix epitheloid carcino cells.

Methods [3,4] were applied to contiguous cells, methods [5,6] were tested only on images containing separated cells with small spaces in between.

All the above-mentioned methods have their advantages, as well as disadvantages (require labeled database, high computational demands, lengthy parameter tuning, ...) and may fail when processing certain types of data (noise, blurring, presence of impurities).

Our work offers an alternative way to segment individual cells. It is based on the presence of halos around individual cells and it can process images of many contiguous cells or images of cell monolayers. To evaluate the algorithm's performance we developed a numerical evaluation on the level of individual cells based on precision, recall and F1 measures.

To provide a baseline for testing performance of previously mentioned methods or any future methods we provide the method's source code, test data and manual segmentation by human experts on our website <http://zoi.utia.cas.cz/IndividualCells>.

Together with the algorithm itself we have developed a software tool for editing the segmentation results (merging or splitting the cell regions), so that any mistakes of the algorithm can be easily and quickly corrected, making the algorithm more usable.

The rest of the paper is organized as follows: Sec. Methods describes our algorithm, and Sec. Results shows the precision of our algorithm when compared to human experts.

## 2 Methods

The goal of our method is to find individual cells in the images, or more precisely their borders. The algorithm consists of several stages. In the first stage, an incomplete set of edges between cells is created by thresholding and subsequent skeletonization. The aim of the second stage is to fill in the missing parts of the edges by connecting the loose ends of the skeleton. The third stage adjusts the resulting edge map by merging too small areas to the neighbouring cell areas. In the following text we first explain the main

principles of the method and then describe modifications that further improve the method.

The main characteristic of phase contrast microscopy images is that cells are often surrounded by a white area – the halo see Fig. 2A. Halos are also present in areas where two cells touch each other. This is used by the first stage of our algorithm. First we slightly blur the image (by gaussian with standard deviation 10) and divide it into lighter and darker areas (see Fig. 2B) using Otsu's automatic thresholding method [7]. The blurring ensures that the thresholding does not select the occasional light points inside the cells. The lighter area therefore consists mostly of the halos. The result of the thresholding is subsequently subjected to skeletonization (see Fig. 2C). We use a modified version of skeletonization [8], which reduces the number of short branches of the main skeleton.

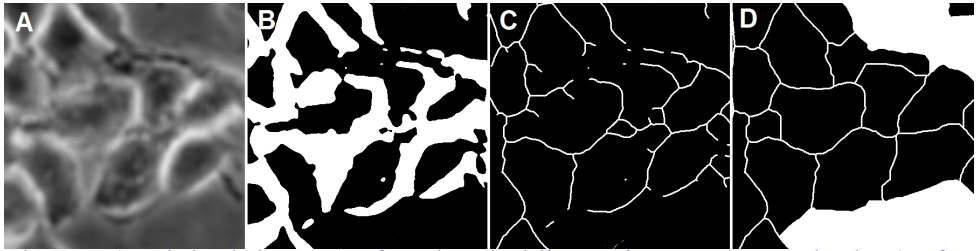


Fig. 2: A) original data, B) after thresholding using Otsu's method, C) after skeletonization, D) result of the method - white region is background, white lines are found cell borders.

We presume that most of the loose ends of the skeletonization results from the halo being less distinctive in the given area, and so in the second stage of the algorithm we connect these loose ends to other skeleton fragments. In order to make these connections closer to reality, we try to make them as short as possible and keep them in the brightest areas. For this we use Dijkstra's algorithm for finding the shortest path in a graph [9]. We represent every pixel as a vertex  $V$  of a graph and edges between the vertices connect neighbouring pixels using 8-neighbourhood connectivity. The edge weight is calculated based on the pixel intensity  $I$  as  $f(I(V_1) + I(V_2))$ , with  $f$  being a function that assigns lower value to the lighter pixels, e.g.  $f(x) = -x$  or  $f(x) = -\log(x+1)$ . For our experiments we used the latter one. For the diagonal edges we multiply the weight by  $\sqrt{2}$ . Dijkstra's algorithm runs from each loose end and finds the nearest part of the skeleton. At the end of this phase we have an outline of individual cell edges.

At the end we check whether the results correspond with our knowledge of the studied cells. For this we use several criteria which we do not discuss due to lack of space but the main is the cell size. If the final border segmentation shows a smaller area than this criterion allows, we fill in this area completely and skeletonize it once more. In most of the cases the smallest possible size of a cell corresponds to the size of a cell in the mitotic phase (cells shrink during this phase). Final result can be seen at Fig. 2D.

The algorithm can be substantially improved when prior to connecting the loose ends of skeleton we first label the image pixels as belonging to cells or background. There are several algorithms available for this task—we use the SegTool programme [2] which is fast and quite accurate. This modification prevents drawing false edges between clusters of cells that do not touch each other. It also makes searching for optimal connections in Dijkstra’s algorithm faster.

Other modifications adjust the process of searching for optimal connections using Dijkstra’s algorithm. The algorithm has a natural tendency to use connections displayed in Fig 3A (the red or yellow connections). To achieve the green connection, we use two ad hoc rules. The first one states that the connecting line must leave the contour. By the contour we mean the result of the Otsu thresholding as seen at Fig. 2B. This rule prevents the red connection, see Fig. 3A. The second rule defines the direction in which a connection line is searched, see Fig. 3B.

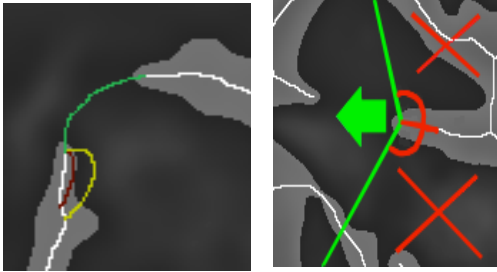


Figure 3: A) Illustration of the problems with connecting skeleton ends. The green connection line is the preferred one. Without additional modifications the algorithm creates the red or yellow connection line. B) Illustration of the rule for direction of connection line search.

The algorithm has several inner parameters. These are ad hoc rules that enhance the performance of the algorithm, such as the amount of initial image blurring and two rules (in Sec. Alg. Modifications) for searching connections between contour ends. These parameters need to be tuned only once to achieve the most successful segmentation and they are fixed in subsequent runs of the algorithm. The minimal size of cells depends on the type of scanned cells and is defined by the user.

### 3 Results

When comparing results of individual cell segmentation with ground truth (in our case manual segmentation done by human experts), we can distinguish three types of errors in the segmentation results: A) misaligned cell borders, B) two or more cells merged into one region C) one cell split into two or more regions. Boundaries between these three error categories are not well defined, combination of merging and splitting could sometimes results in cell border misalignment and vice versa. Therefore our evaluation method measures these three types of errors together.

For each cell from the manual segmentation we found an area from our algorithm that had the largest overlap. For all these pairs we measured the size of the overlapping area – true positive (TP), the size of the area in the

manual segmentation apart from the overlap – false negative (FN), and the size of the area in our segmentation apart from the overlap – false positive (FP). We added up the sizes of TP, FP and FN of all the cells pairs and converted them to precision ( $P = TP/(TP+FP)$ ), recall ( $R = TP/(TP+FN)$ ) and F1 ( $F1 = 2PR/(P+R)$ ) measures.

The algorithm was implemented in Java and Matlab. The CPU time (using Dual core 2.3 Ghz) for segmentation of 4 MPixel image was about 30 seconds. The algorithm was tested using three different types of cells: HeLa cervix epitheloid carcino cells (4 images), L929 mouse fibroblast (1 image), and vero cells E6 (3 images). The images were taken using Olympus X51S8F-3 microscope with magnification 20x and resolution 2288x1712. In total 8 images were considered on which were almost 2000 cells. We compared the algorithm results with results of manual segmentation done by an expert. The mean precision values for all of the test images are  $P = 0.65$ ,  $R = 0.73$ , and  $F1 = 0.68$  (cell-background labelling was provided by SegmentationTool) and  $P = 0.66$ ,  $R = 0.74$ , and  $F1 = 0.69$  (cell-background labelling was taken from manual one by merging all the cells). Number of the cells that our algorithm found is about 13% higher than real number.

As we have already mentioned, the results of the automatic segmentation algorithm can be improved (connecting or dividing areas and shifting edges) by our software tool. The quality of segmentation therefore influences only the amount of work needed to correct the mistakes.

The obtained results are graphically represented in Fig 3. Yellow signifies under-segmented areas (where our algorithm connected two real cells into a larger one), light blue is for over-segmented areas (containing extra borders). Green signifies areas that were segmented accurately and dark blue is used for background without any cells on it.

For comparison, we tested ordinary watershed method on our dataset. Precision of this method is much worse ( $P = 0.87$ ,  $R = 0.26$ ,  $F1 = 0.40$ ) mainly due to oversegmentation (algorithm found 22-times more cells than present in the image).

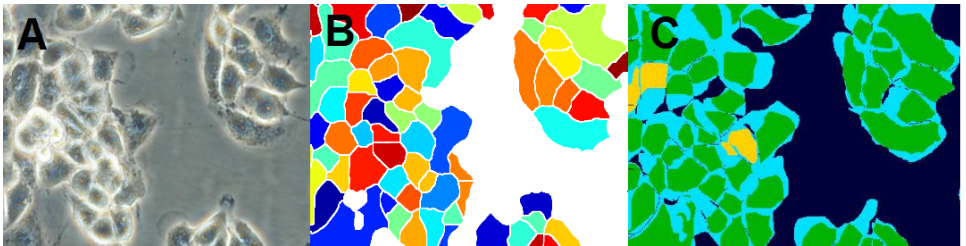


Figure 3: Results of our algorithm. A and D are the original images. B and E are examples of segmentation by our algorithm. C and F show correspondence with manual segmentation (green areas are correct, light blue ones are under-segmented and yellow are over-segmented. Dark blue is used for background).

## 4 Discussion and conclusions

We introduced a new method for segmentation of individual cells in images acquired by phase contrast microscopes. Unlike previous segmentation methods, it uses the presence of halos around individual cells.

One of the weaknesses of this algorithm is that it processes the individual images from time series separately. The algorithm could be improved by connecting tracking and segmentation. Further improvement could be achieved by using methods, such as active contours or level-sets, since the results of our algorithm could provide a good initialization for them.

Our method is currently being tested at Working place of tissue culture - certified laboratory, which researches on new potential materials for implants in vitro.

## Acknowledgment

The results of the project LO1205 were obtained with a financial support from the MEYS under the NPU I program, CENAKVA CZ.1.05/2.1.00/01.0024. The authors acknowledge the support of the GA JU 134/2013/Z, GAUK, grant No. 914813/2013 and grant GACR No. 13-29225S.

## References

- [1] Ersoy, I. et al.: Cell segmentation using Hessian-based detection and contour evolution with directional derivatives, *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, **2008**, 2008, 1804-1807
- [2] Soukup, J. et al.: Segmentation of Time-Lapse Images with focus on Microscopy Images of Cells, *17<sup>th</sup> International Conference on Image Analysis and Processing - ICIAP 2013, Springer-Verlag*, 2013, 8157, Part II, 71-80
- [3] Wang, X. et al.: Cell segmentation for division rate estimation in computerized video time-lapse microscopy, *Proceedings of SPIE, Spie*, **2007**, 6431, 643109-643109-8
- [4] Pan, J. et al.: Heterogeneous Conditional Random Field: Realizing joint detection and segmentation of cell regions in microscopic images, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE*, **2010**, 2940-2947
- [5] Li, K. et al.: Cell population tracking and lineage construction with spatiotemporal context, *Medical image analysis*, **2008**, 12, 546-66
- [6] Yin, Z. & Kanade, T.: Restoring artifact-free microscopy image sequences, *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, **2011**, 909-913
- [7] Otsu, N.: A threshold selection method from gray-level histograms, *Automatica*, **1975**, 20, 62-66
- [8] Hilditch, C.: Linear Skeleton from square cupboards, *Machine intelligence*, **1969**, 6, 403-409
- [9] Dijkstra, E.: A note on two problems in connexion with graphs, *Numerische Mathematik, Springer-Verlag*, **1959**, 1, 269-271