32nd Annual International Conference of the IEEE EMBS
Buenos Aires, Argentina, August 31 - September 4, 2010

Learning Cellular Texture Features in Microscopic Cancer Cell Images for Automated Cell-Detection

Tomáš Kazmar,* Matěj Šmíd,* Margit Fuchs,† Birgit Luber,‡ and Julian Mattes*

Abstract— **In this paper we present a new approach for automated cell detection in single frames of 2D microscopic phase contrast images of cancer cells which is based on learning cellular texture features. The main challenge addressed in this paper is to deal with clusters of cells where each cell has a rather complex appearance composed of sub-regions with different texture features. Our approach works on two different levels of abstraction. First, we apply statistical learning to learn 6 different types of different local cellular texture features, classify each pixel according to them and we obtain an image partition composed of 6 different pixel categories. Based on this partitioned image we decide in a second step if pre-selected seeds belong to the same cell or not. Experimental results show the high accuracy of the proposed method and especially average precision above 95%.**

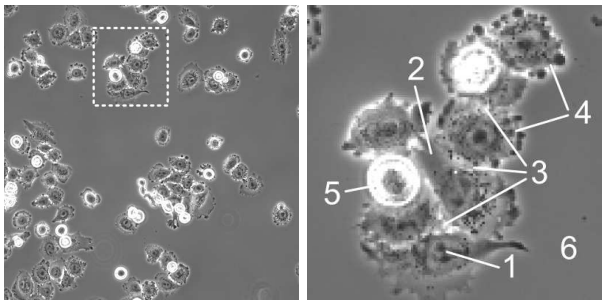
I. Introduction

The motion pattern of cancer cells provides insight into the functional state of the cells. In order to study the effect on the motile behavior of the cells, when treating them with an anti-cancer drug in the frame of an anti-migratory cancer therapy approach [1], the motion of a large number of cells has to be explored and quantified. For this purpose methods for automated cell detection and cell tracking contribute to systematic and reproducible measurements allowing to handle large amounts of data where a manual evaluation would be too time-consuming.

In the presence of single isolated cells, cell detection reduces to the calculation of a foreground/background mask [2]. However, in order to limit the effort of recording films of the moving cancer cells we record approximately 50 cells at a time (see Fig. 1(a)). The cells tend to agglomerate in this case and, with increasing number and size of the cell clusters, cell detection becomes more and more challenging. In contrast to other cell detection methods which do not deal with clusters at all [3], [4], or not with larger clusters [5], we address such situations as well. Moreover, our aim is to detect a cell by identifying its center of the nucleus which is used for tracking the cells afterwards.

A point which makes the detection of a single cell in a cell cluster difficult results from the specific appearance of the cells in our images which differs with respect to the images shown in other papers [6]. Fig. 1(b) illustrates that cell appearance can be described as being composed by the following parts: a nucleus containing fine grains, a halo around the nucleus, cytoplasm, membrane and the cell border again showing a halo but also some dark coarse grains. Clearly, this considerable number of different structures is not easy to deal with when aiming at detecting the different cells in a cluster. For instance, the halo around the nucleus might be taken for the halo at the cell border and two cells are detected instead of one.

Our approach works on two different levels of abstraction. First, we apply statistical learning to learn 6 different types of different local cellular texture features and partition the image. At the same time we generate initial seeds for cell centers where we use two different methods: radial symmetry decomposition (RSD) [7], [8] and blob-like keypoint detection [9]. RSD decomposes larger cell clusters into radially symmetric parts and it provides centers of radial symmetry. Blob-like keypoint detection is a method based on eigenvalue analysis of the Hessian of local neighborhoods in the image. Using the partitioned image we decide in a second step if these pre-selected seeds belong to cells or not. This is in contrast to Pan et al. [10], who also use statistical learning in order to draw conclusions about pre-selected points, but make a decision based directly on the input image.



(a) input frame (b) subregion (magnified 4x)

Fig. 1. Example input frame from one of the sequences, enlarged subregion shows nucleus (1), membrane (2), halo (3), border (4), float (5) and background (6).

II. Method

The proposed method has three major steps in order to accomplish cell detection: first, cell seeds are generated in the form of small regions, second, the image is partitioned based on cellular features, and last, seeds are filtered and joined in order to reduce the number of false positives (FP) as much as possible, see Figure 2.

* Tomáš Kazmar, Matěj Šmíd, and Julian Mates are with the Biomedical Data Analysis Group, Software Competence Center Hagenberg GmbH, Softwarepark 21, A-4232, Austria, tomas.kazmar@scch.at

† Margit Fuchs is with Centre de recherche en cancérologie, Université Laval, HDQ, CRCHUQ - Québec, Qc, G1R 2J6, Canada.

‡ Birgit Luber is with Institut für Allgemeine Pathologie und Pathologische Anatomie, Technische universität München, Klinikum rechts der Isar, D-81675 München, Germany

978-1-4244-4124-2/10/$25.00 ©2010 IEEE

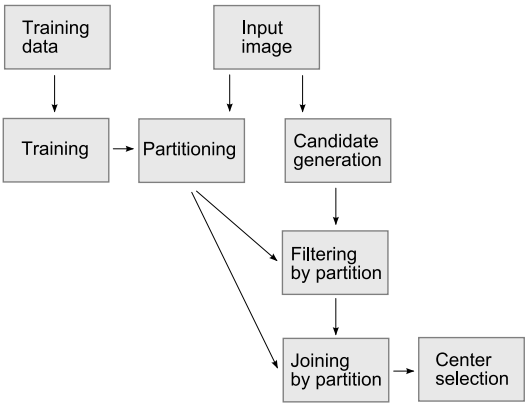


Fig. 2. Method scheme.

A. Preprocessing

Background mask distinguishing between the background and cells greatly reduces the space we need to explore in further steps. In order to extract such a mask M, thresholding of statistical variation of input image F measured by: intensity range, variance, median of absolute deviations from median, sample-to-noise ratio, or entropy etc. is a usual step [11], [12]. In our case measuring entropy in small neighborhoods proved to work the best as it gives a bimodal histogram where the two modes are easily separated, eg. by expectation-maximization fitting of two Gaussians. One has to keep in mind that this relies on the fact there is a sufficient background area as well as a cell area.

B. Initial seed generation

The first step of our method after preprocessing is seed generation, its goal is to produce probable locations of cell centers based on local image intensity information without missing any of the cells. Thanks to the general setup of our method seed generation can be accomplished in any meaningful way, here we propose two possibilities: first, radial symmetry decomposition [7], [8], and second, blob-like keypoint detection [9].

1) Radial symmetry decomposition (RSD): In RSD, each image pixel with high gradient votes for centers of symmetry (another pixels). Votes from all the pixels are accumulated. First the pixels vote for a larger area in the direction of gradient but with each iteration the voting area shrinks and the voting direction is re-oriented towards local maximum in the accumulator. Final local maxima serve as the generated seeds. For further details please refer to [8], we apply the RSD algorithm defined therein with small changes in the weighing of votes where we take into account just the gradient magnitude as opposed to a linear combination of gradient magnitude and the actual intensity.

2) Blob-like keypoints (BK): Application of the Hessian to analyze local structure comes from the context of interest point detection [9]. We use it in the following way, a matrix of partial derivatives of the image function F:

$$H = \begin{bmatrix} \frac{\partial F}{\partial^2 x} & \frac{\partial F}{\partial x \partial y} \\ \frac{\partial F}{\partial x \partial y} & \frac{\partial F}{\partial^2 y} \end{bmatrix}$$

is constructed and eigenvalue analysis of H performed. Depending on the maximum, minimum eigenvalue λ_1, λ_2 one can argue about local structure, if both are large there is a maximum or minimum according to the sign, if both are small the function is flat, other cases are not interesting for us. To generate the seeds we propose to take connected regions with $L(\vec{x}) > \sigma_L$ in:

$$L = \lambda_1(\vec{x})\lambda_2(\vec{x}) \text{ over } \vec{x} \in \left\{\mathbb{R}^2 \mid \lambda_1(\vec{x}) > 0, \lambda_2(\vec{x}) > 0\right\}$$

To approximate the derivatives we use a convolution with a Gaussian derivative filter along with smoothing by a Gaussian. Both have the same scale which is smaller than that of nuclei so that we do not miss any cell. If the cell scales would differ too much, it is possible to generate the seeds using multiple filters at different scales and feed them to the following steps of the method.

C. Image partitioning by learning cellular features

In order to abstract from the image data we want to interpret the image in terms of cellular features. We define the following six different features: *nucleus*, *membrane*, *halo*, structured *border*, *float*,[1] and the *background*. The selected cellular features have differing appearance and biological meaning, see Figure 3. *Nucleus* and the structured *border* are both textured and slightly darker, the texture of *nucleus* coarser, structured *border* is formed by ruffling and other membrane extensions at the cell-background interface. Both *membrane* and *background* are flat, *membrane* usually having wider histogram but smoother texture. *Halo* and *float* are prominent features typical for phase-contrast microscopy which have high intensities, with *halo* there is a texture and thin linear shape whereas *float* has nearly no texture and covers larger areas.

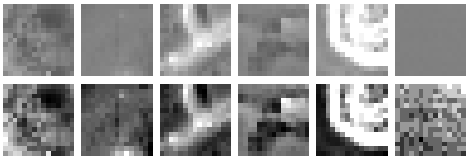


Fig. 3. 15x15 patches of the six cellular features, first row original, second row normalized for output. Only *nucleus*, *membrane*, and *background* cover the whole patch, *halo* and *border* are mixed with each other and *float* appears with a part of *background*.

Machine learning is used to learn these six cellular features. Specifically our learner of choice is an ensemble of randomized decision trees, aka random forest [13], for a discussion of this choice see Section III-A.1. The input space to the classifier is based on analyzing local 5x5 neighborhoods

[1]This is no more than our label, definitely no well-established term, but we think it describes well the fact that this cell feature is produced by (mitotic) cells which are not settled on the substrate and float in the medium.

of each pixel in terms of texture using 14 texture features [14]. Before the analysis we correct vignetting artifacts in the image [15].

D. Seed selection

Inputs to seed selection are both the initial cell seeds and the interpretation of the image in the form of cellular-feature partitioning. Using the partioning we select the seeds that are placed correctly over cells — we call this step filtering, and merge the seeds that happen to lie inside a single cell — we call this joining. The goal is to lower the number of false positives as much as possible.

Filtering takes each cell seed and checks the underlying cellular features computing the ratio r_{class} which expresses seed area covered by each of them. If the seed is too small it is enlarged in order to obtain reliable statistic. As the right seeds should be placed over nucleus and never over background, or membrane, and as sometimes the centers of small cells are composed rather by *float* than by *nucleus*, the decision is taken based on:

$$\begin{gathered}(r_{nucleus} > t_{nucleus} \vee r_{float} > t_{float}) \wedge \\ r_{membrane} < t_{membrane} \wedge \\ r_{background} < t_{background}\end{gathered}$$

where t_{class} are parameters of the method. This rule is complemented by a rule checking that the distance d to the closest *background* region is above a minimum D.

In the joining step, we construct a matrix of mutual interconnection costs which is achieved once more by looking at the partitioning, the r_{class} ratios are computed for the region separating the two seeds and when

$$r_{halo} + (r_{background} + r_{membrane}) - (r_{nucleus} + r_{border})$$

is under a specified threshold J the two seeds are replaced by their convex hull. Finally, seeds covering areas smaller than S are discarded. To speed up the computation gating is applied — costs are evaluated only for pairs of cells which are close enough (twice the average cell radius was taken).

III. Experimental Results

We evaluated our algorithm on 7 films (512x512 px 8-bit grayscale) showing moving MDA-MB-435S breast cancer cells. The cells have been imaged in average during 7 hours, 1 frame per 3 minutes, using an Axiovert laser scanning microscope LSM 510 (Zeiss) with PNF 20_0.4 PH2 lens and a HeNe laser at 543 nm in transmission scanning mode [16].

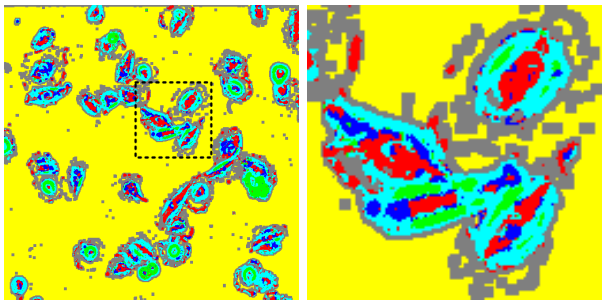
A. Training, testing and validation data

A single frame from one sequence was used for training the partitioning which models the final setup in a biological laboratory environment where a skilled biologist is to be asked to mark different cellular features in one representative frame. Three films were used for testing — to tune the method parameters, another three films served as a means for validation when all the parameters had already been fixed.

TABLE I
Recall/precision of partitioning for different machine learning methods.

	DT	*SVM*	*RF*
bg	.98 / .96	.98 / .96	.98 / .96
border	.57 / .62	.58 / .53	.67 / .79
float	.87 / .93	.80 / .91	.93 / .97
halo	.86 / .72	.61 / .60	.93 / .86
membrane	.53 / .65	.55 / .70	.56 / .74
nucleus	.75 / .76	.78 / .76	.88 / .81
summary	.76 / .77	.72 / .74	.82 / .85

1) Test 1: partitioning space and the machine learning algorithm: Apart from random forest (RF, implementation from [17]), we tested support vector machines (SVM), specifically C-SVM as implemented in [18], and decision trees (DT) [19], always applying hyper-parameter tuning and keeping the data set balanced. Differences between DT and SVM were not significant while both were outperformed by RF in the sense of recall and precision, see Table I. Furthermore, an advantage of RF was its easy hyper-parameter setup. A typical output of partitioning is shown in Figure 4. The cell features which are hard to be distinguished are *membrane* vs. *background* and *border* vs. *nucleus*, most probably higher-level information would be needed. As long as in joining we treat these features identical and in filtering nearly identical, current scale was retained.



(a) partitioning (b) subregion (magnified 4x)

Fig. 4. Typical partitioning output, different cell features are color coded: nucleus (blue), membrane (gray), halo (light blue), border structures (dark blue), float (green), and background (yellow).

Apart from the Haralick texture features [14] finally applied, we further experimented with the input space fed into the classifier. The simplest space formed only by the 25 original patch intensities did not yield good results. Variable importance output by RF confirmed that the central intensity I_{00} is by far the most important which led us to the assumption that a space formed by a combination of I_{00} and the texture features would perform better. However the improvement obtained (recall increased from 82 to 84%, precision from 85 to 86%) did not outweigh more common overfitting. We also experimented with different texture spaces formed by statistical measures such as mean, variance, second normalized image moments, covariances etc. without getting significantly better results.

TABLE II
FINAL RSD/BK RESULTS AFTER THE JOINING STEP.

method		*average*	*minimum*	*maximum*
RSD	*FP*	1.0	0.	5.
	FN	33.	16.	48.
	Recall	0.50	0.28	0.63
	Precision	0.98	0.91	1.0
BK	*FP*	2.1	0.	6.
	FN	18.	6.	35.
	Recall	0.73	0.58	0.85
	Precision	0.96	0.92	1.0

2) Test 2: comparison of seed generators: We tested the quality of seed generation for further processing by our method, therefore we looked at FN/recall, not caring much about FP/precision. As the methods generate a lot of seeds close to the background in completely incorrect positions, we removed such seeds by dilating the background mask to obtain better comparison.

The evaluation of the obtained seeds was done in the following way, from each of the films we took 5 random frames, for each of these frames ground truth was defined interactively and the results of the method were compared to it. Evaluation was done automatically assigning each ground truth cell location to the closest detected seed using the Gale/Shapley algorithm [20]. In situations where the closest seed did not lie within 20 pixels[2] the cell was considered undetected. On the three test sequences, RSD gave 90% recall and BK 94%. The average number of false negatives with RSD was 4.9 (max. 8), with BK 3.1 (max. 6).

The runtimes for the two different seed generation methods differ greatly, BK runtime being in the order of seconds and RSD up to several minutes.

3) Test 3: method evaluation: Finally, in the last experiment the whole method was tested first fixing the parameters on the three testing sequences, $t_{nucleus} = 0.6$, $t_{membrane} = 0.2$, and $t_{background} = 0.1$, $t_{float} = 0.8$, $D = 7$ px for filtering, and $J = 0.3$, $S = 20$ px^2 for joining. Afterwards the method was evaluated with the validation sequences.

Filtering as well as joining steps are crucial for the good performance of the method. Regardless of the seed generation method we reached over 95% precision (avg. 1.0 FP for RSD and 2.1 for BK), starting as low as 50% precision for RSD and 27% for BK. Nonetheless, we consider the final results of BK after joining to be much better as there is a great difference in the runtime of both methods as well as in recall (73% over 50% in favor of BK). Although recall is not crucial, when we consider post-processing by user which can add cells with more ease than spot and delete the wrong ones, the value of RSD is probably too low as in the interactive setup it would mean adding half of the cells to the ones detected by the algorithm. For the results after the joining steps see Table II.

IV. CONCLUSION

The proposed method gives high precision with a reasonably high recall, if we consider our challenging input sequences it is unparalleled among the currently available state-of-the-art methods for cell detection from single image in phase-contrast microscopy. High precision (with both seed generators over 95% on the validation data) is especially important to limit necessary user interaction (adding cells is easier than deleting). Further, we plan to investigate other seed generators apt for special types of cells, as well as, to learn the decision criteria for filtering and joining. Finally we plan to integrate the method in our cell tracking software so that we are able to exploit temporal information extracted by tracking to improve detection.

[2]Approx. half the size of a typical cell.

V. ACKNOWLEDGEMENTS

This work was carried out inside the CANCERMOTISYS project `http://www.cancermotisys.eu/`, supported by BMWF, Austria, and BMBF, Germany.

REFERENCES

[1] C. Decaestecker, O. Debeir, P. Van Ham, and R. Kiss, "Can anti-migratory drugs be screened in vitro? a review of 2D and 3D assays for the quantitative analysis of cell migration." *Med Res Rev*, vol. 27, no. 2, pp. 149–76, Mar 2007.

[2] K. Wu, D. Gauthier, and M. D. Levine, "Live cell image segmentation," *IEEE Transactions on Biomedical Engineering*, vol. 42, pp. 1–12, 1995.

[3] I. Ersoy, F. Bunyak, M. A. Mackey, and K. Palaniappan, "Cell segmentation using hessian-based detection and contour evolution with directional derivatives," in *ICIP*, 2008, pp. 1804–1807.

[4] C. Bradhurst, W. Boles, and Y. Xiao, "Segmentation of bone marrow stromal cells in phase contrast microscopy images," in *IVCNZ08*, 2008, pp. 1–6.

[5] O. Al-Kofahi, R. Radke, S. Goderie, Q. Shen, S. Temple, and B. Roysam, "Automated cell lineage construction," *Cell Cycle*, vol. 5, no. 3, pp. 327–35, 2006.

[6] K. Li, M. Chen, and T. Kanade, "Cell population tracking and lineage construction with spatiotemporal context," in *Proceedings of the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2007, pp. 295 – 302.

[7] Q. Yang, B. Parvin, and M. Barcellos Hoff, "Localization of saliency through iterative voting," in *ICPR'04*, 2004, pp. I: 63–66.

[8] O. Schmitt and M. Hasse, "Radial symmetries based decomposition of cell clusters in binary and gray level images," *Pattern Recognition*, vol. 41, no. 6, pp. 1905–1923, 2008.

[9] T. Lindeberg, "Scale-space theory in computer vision," 1994.

[10] J. Pan, T. Kanade, and M. Chen, "Learning to detect different types of cells under phase contrast microscopy," in *MIAAB*, 2009.

[11] F. Ambriz-Colin *et al.*, "Detection of biological cells in phase-contrast microscopy images." in *MICAI*, 2006.

[12] C. Wahlby *et al.*, "Finding cells, finding molecules, finding patterns." ser. LNCS, vol. 4826. Springer, 2007, pp. 104–114.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[15] B. Likar, J. B. A. Maintz, M. A. Viergever, and F. Pernus, "Retrospective shading correction based on entropy minimization," *Journal of Microscopy*, vol. 197, no. 3, pp. 285–295, 2000.

[16] M. Fuchs *et al.*, "Motility enhancement by tumor-derived mutant e-cadherin is sensitive to treatment with epidermal growth factor receptor and phosphatidylinositol 3-kinase inhibitors," *Experimental Cell Research*, vol. 276, pp. 129–141, 2000.

[17] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[18] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.

[19] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[20] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.