ZESZYTY NAUKOWE POLITECHNIKI BIAŁOSTOCKIEJ 2008
Informatyka – Zeszyt 3

Cezary Bołdak[1], Marcin Sadowski[1], Jerzy Jaroszewicz[2]

SEMI-AUTOMATIC COUNT OF HEPATIC STELLATE CELLS FROM MICROSCOPIC IN-VITRO IMAGES BY MODELING ELLIPTICAL NUCLEI

Abstract: A new method of semi-automatic liver cell counting is presented. Instead of segmenting the cells bodies (not regular and fragmented in some stages of the cells life) it localizes the cells nuclei which are bright, homogeneous and elliptical structures with darker body (body fragments) on their circumference. The nuclei are modeled by ellipses which can be found in two manners: by local region growing algorithm and by reconstruction of the ellipse equation from its contour points. The found ellipses set is then downsized (since all possible ellipses are initially considered) by eliminating the closest one to another and the worst ones by mean of a special fitness function. The method is implemented as a visual, multiplatform JAVA application, easy to use in the scientific every-day work. It is evaluated on real microscopic in-vitro images of the hepatic stellate cells.

Keywords: medical image processing, cell count, hepatic stellate cells

1. Introduction

In the modern medicine different techniques of imaging play very important role. Thanks to other than optical acquisition modalities (X-rays, ultrasounds, magnetic fields, infrared rays, etc.) they offer a non-invasive visualisation of internal body structures and even of body processes (e.g. fMRI). Images recording helps in the physicians' education, exchanging of knowledge between them and in temporal analysis of disease development or treatment efficacy.

On the other hand, the physician can get overwhelmed with the quantity, dimensions and complexity of such images. All this visual information (or its major part) would be useless without efficient and pertinent methods of image processing. Starting from sometimes trivial image quality enhancement (contrast, noise removing), through multimodal images registration, compression, tranmittion and visualization,

[1] Faculty of Computer Science, Balystok Technical University, Białystok, Poland
[2] Department of Infectious Diseases, Medical University of Biaşystok, Białystok, Poland

up to sofisticated segmenting of interesting structures from images and characterizing them[10], modern computer-aided tools can improve the human work or at least make it less tedious.

An example addressing the latter issue is a problem of object counting, where the goal is to give exact or approximative quantity of some (similar) objects present in the image. This work, quite common in the medicine, seems very easy to the human, but when the object number increases, it becomes tedious and error-prone, and this is why there have been many attemps to make it automatic. [5] and [12] count (segments) spermatozoides, [2,6,8,9] - blood cells, [1] - survived cells in photodynamic therapy, [11] - liver cells, [13] - neural stem cells, [7] - neutrophils and lymphocytes.

Chronic liver injury results in hepatocyte damage, which triggers activation of hepatic stellate cells (HSC) [4]. HSC play a central role in liver fibrosis development. Following the stimulation, those cells undergo phenotypic transdifferentiation to myofibrolasts. Along with morphological transformation they became major source of collagen type I in the liver and secrete pro-fibrogenic cytokines and inhibitors of matrix-degrading enzymes (tissue inhibitor of matrix metalloproteinase). This causes the increased extracellular matrix deposition over degradation, which eventually leads to liver cirrhosis development. Thus, HSC represent an appealing target for antifibrotic therapy.

This article is organized as follows. Section 2. gives a review of a few automatic counting methods from the bibliography. Section 3. describes 2 proposed algorithms for liver cells counting. Section 4. contains a short description of an application implementing our method. Experiments with real microscopic images of in-vitro liver cells using our 2 algorithms compared to a simple region growing technique and to an expert counting are presented in section 5.

2. Background

Numerous works start the counting with a preprocessing of the treated image. The noise can be removed by a low-pass filtering [1,9]. Very common technique used here is the thresholding [12], sometimes multilevel [8], adaptive [11] or local [1]. The objects boundaries can be also discovered here by an edge detector [1,9]. All this is very often followed by an extra operation improving the resulting binary image, like the mathematical morphology opening/closing [8,9,11].

The objects to count can then be identified as groups of connected binary pixels [8,9,11]. The watershed segmentation can also isolate the objects from the background [1]. If the image is not converted to the binary one, other sophisticated techniques can be applied to segment the object contours [6]. [13] uses Voronoi diagrams to separate regions occupied by the stem cells. [7] combines 2 techniques: region-based (using basic texture features) and contour -based (cells boundaries detected as high-gradient structures) into one hybrid method to segment different cells.

At the end, these results can undergo a post-treatment phase. Among its goals one can mention: elimination of groups not satisfying some conditions (too small size [1,11], geometrical center located outside the object [1,11], shape contour not accepted by a neural network trained on desired shapes [6], presence of background pixels inside the object [8]), dividing groups into several distinct objects, touching one another (basing on too big size [11] or radius variation [8]), incorporating close, distinct fragments to the main object (e.g. basing on the Hough transform [8]).

Very exhaustive review of application of different techniques to the count problem is given in [5].

3. Method

As shown at Figures 1 the cells in different stage of life can have their bodies fragmented what makes segmenting them using algorithms based on a simple pixel grouping (e.g. region growing) impossible. Finding a model can also be a hard task because of their varied shapes. However, we have noticed there is one stable feature in these cells - their nuclei. They have in the most of cases an elliptical form, homogeneous inside and bordered by the cell body or their fragments. This is why we have focused on them and proposed a method counting regions with the given above characteristics.

Our method consists of three steps. In the first one the image is preprocessed to get a binary representation of pixels belonging to the background and to the segmented cells. In the second step we construct ellipses - candidates for the cells nuclei. Two approaches are considered - one based on the region growing and one approximating the ellipse equation from a set of border points. These candidate ellipses are far too numerous and in the final, third step the best local ones are chosen following a special fitness function.

Images treated in this work should be gray level ones - color information is not exploited here.

3.1 Preprocessing

The main goal of this step is to produce a binary image with one pixel set representing the background and the cells nuclei and the second set representing the cells bodies. The latter structures are darker in the processed images so a simple thresholding could do this work. The first problem are variations in the cells bodies gray level because of their grain structure (Figure 1(a)). To smooth the levels we blur the images by convolving them with the Gaussian filter mask [10] (Figure 1(b)). The second problem is probably with the image acquisition process. The scanned images are sometimes not properly exposed and some their regions can be brighter than others (Figure 1(c)). One global threshold can not be applied here. To properly differentiate the cells pixels from the background we used an adaptive threshold technique where each pixel has its own threshold value calculated locally. It increases the calculation time but allows appropriate handling of real images. Each local threshold $T(x,y)$ is computed in a local square neighborhood of the point (x,y) as a mean gray level in it, decreased by a positive constant C - parameter of this technique. This constant can be interactively adjusted by the operator but in all the cases here its value was equal 5.

$$T(x,y) = \left(\frac{1}{N^2} \sum_{i=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} \sum_{j=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} I(x+i,y+j) \right) - C \qquad (1)$$

where N is a size of square local neighborhood (equal 21 in all our experiments), $\lfloor . \rfloor$ - floor operation, and $I(x,y)$ gray level of the image in location (x,y). To improve the effectiveness of this stage, the above adaptive threshold is calculated only for a limited set of pixels and linearly interpolated for the rest of them. Experiments have shown that calculating every $\lfloor N/2 \rfloor$ pixel in x and y dimensions preserves the thresholding accuracy and significantly reduces the calculation time. So the binary image BI is constructed following the rule below:

$$BI(x,y) = \begin{cases} 'cell' & \text{if} \quad I(x,y) < T(x,y), \\ 'background' & \text{if} \quad I(x,y) \geq T(x,y). \end{cases} \qquad (2)$$

One can observe that the binary contours (Figure 1(d)) not always closely follow the real ones (Figure 1(c)) - they can be narrower, but this slight change does not influence the elliptical form of the nuclei. It can be stated very clearly here - the main interest of this work is to count the cells, not to segment their real shape.

At the end of this stage all connected groups which size is below some interactively chosen value are removed from the BI - they the most likely are artifacts.

3.2 Construction of ellipses

We propose 2 different techniques to construct ellipses representing the cells nuclei. The first one, based on the region growing algorithm, should be faster when applied to smoother images, with low noise level, but its performance can be disturbed by

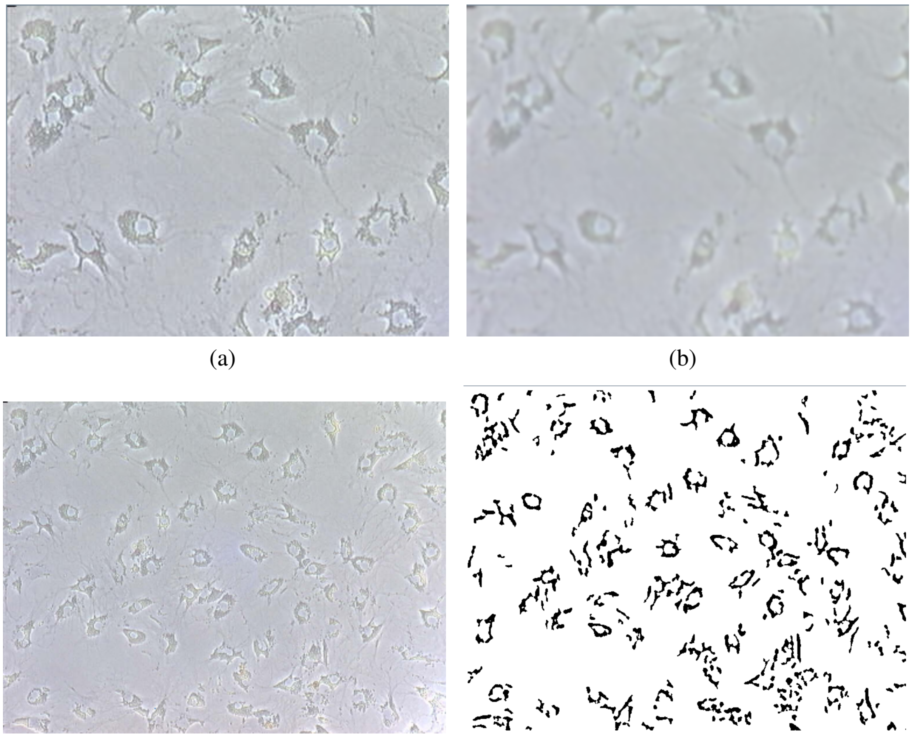


Fig. 1. Analysed image (a) A fragment showing the grained structure of the cells bodies (b) Result of the Gaussian blurring removing the grained structure (c) Entire image showing the mean gray level changing smoothly (d) Result of the adaptive thresholding (after removing the smallest marked groups)

even one noise pixel inside the cell nucleus. The second one can restore an ellipse equation even in presence of several noise pixels but is more complex.

In two mentioned above solutions the entire image image is traversed horizontally and vertically to examine all potential locations of the cells nuclei centers. It does not necessarily demand visiting all pixels in the image - we have observed that using both vertical and horizontal step of traversing equal 2 does not influence at all the performance of the count and significantly accelerate the process. Further increasing of this step can be possible but requires deeper analysis in function of the cells dimensions.

Both technique use one parameter R_{max} - the maximal radius of the cell nucleus, given visually by the operator according the the treated image.

Region-growing based In this variant the ellipse centered in the point $S(S_x, S_y)$, with its two axes equal a and b and rotated by angle ω around the OX axis is given by the following equations (3).

$$\begin{aligned} x &= S_x + a \cdot \cos\alpha \cdot \cos\omega - b \cdot \sin\alpha \cdot \sin\omega \\ y &= S_y + a \cdot \cos\alpha \cdot \sin\omega + b \cdot \sin\alpha \cdot \cos\omega \end{aligned} \tag{3}$$

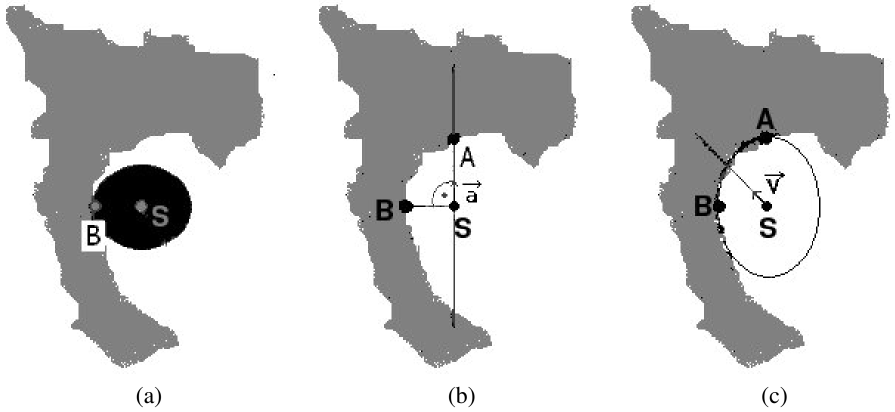
where $\alpha = 0..2\pi$.

In fact it is enough to know locations of 3 points: center point S and 2 extremal points, farer A and closer B, as shown at Figure 2, to compute all values from (4):

$$a = |A - S|, \qquad b = |B - S|, \qquad \omega = \arctan\frac{a_x}{a_y}, \tag{4}$$

where $\overrightarrow{a} = \overrightarrow{S - B} = [a_x, a_y]$.

As it has been stated above, we examine all possible locations of the seed point S. In order to localize the point B we look for the closest pixel marked $'cell'$ around S. It is found by the region growing algorithm [10] with a queue (FIFO - First In First Out) instead of a stack (FILO - First In Last Out) as a structure holding pixels locally marked as 'to be visited' (Figure 2(a)). This modification makes the algorithm to include the closest pixels in the first order. This search is of course spatially limited by the maximum nucleus radius R_{max}. Once the point B located, we place the point A at the closest pixel marked $'cell'$ along two vectors $\overrightarrow{a}$ and $-\overrightarrow{a}$ perpendicular to the vector $\overrightarrow{b} = \overrightarrow{S - B}$ (Figure 2(b)). Once again we limit the search to the radius R_{max}. If any of the points A and B is absent, we reject the ellipse centered at S.



(a) (b) (c)

Fig. 2. Finding of ellipses - method 1 (a) Region growing from the seed point S finding point B (b) Finding of point A (c) Found ellipse

Approximating of the ellipse equation The previous technique is quite sensitive to irregularities in the elliptical form of the cell nucleus and to not removed noise. This is why we have proposed the second one, which is to limit the influence of local variations of the nucleus form from the ideal ellipse.

We start again from the every possible location of the the nucleus center - the seed point S. Starting from it we launch lines equally dividing the angle 2π around this point and collect the closest pixels marked $'cell'$ along this lines - $\mathbf{x}_i = (x_y, y_i)$ (Figure 3). This search is also spatially limited by the maximum nucleus radius R_{max}. If these points number is below some threshold, the potential ellipse centered on S is rejected. To get an ellipse from this set of points we employ a solution proposed in [3]. A general conic is represented by a set of points $\mathbf{x}$ satisfying the following (5):

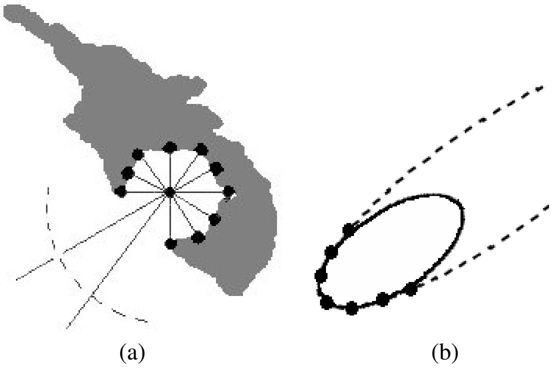


Fig. 3. Finding of ellipses - method 2 (a) Finding border points $\mathbf{x}_i$ from the seed point S (b) Reconstructing an ellipse from points

$$\mathbf{A} \cdot \mathbf{XY} = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \tag{5}$$

where $\mathbf{A} = [ABCDEF]$ is a coefficients vector, initially uknown and $\mathbf{XY} = [x^2\ xy\ y^2\ x\ y\ 1]^T$. In [3] it was shown that a solution minimizing the least square distance between the points $\mathbf{x}_i$ and the conic in presence of some quadratic constraints on $\mathbf{A}$ (in form $A^T\mathbf{CA} = 1$, to eliminate the trivial solution $\mathbf{A} = 0$) can be found by considering rank-deficient generalized eigenvalue system:

$$\mathbf{D}^T\mathbf{DA} = \lambda\mathbf{CA}, \tag{6}$$

where $\mathbf{D} = [\mathbf{XY}_1, \mathbf{XY}_2 \cdots \mathbf{XY}_n]$ is called the *design matrix*, and $\mathbf{C}$ is a 6x6 *contraint matrix*. [3] proposed a contraint $4AC - B^2 = 1$ what gave:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{7}$$

From six pairs $(\lambda_j, \mathbf{A}_j)$ the only one, with $\lambda_j < 0$, is retained (it was proved there would be exactly one such pair), giving the coefficient vector $\mathbf{A}$ from (5).

However, to calculate contour points of the ellipse, similarly to (3), we need the parameters: center point S, axes a and b and rotation angle ω. To find them we use the following strategy: the coordinate system, turned by angle θ, changes the coordinates accordingly to:

$$x' = x\cos\theta + y\sin\theta \qquad y' = -x\sin\theta + y\cos\theta. \tag{8}$$

Inserting the inverse tranformation:

$$x = x'\cos\theta - y'\sin\theta \qquad y = x'\sin\theta + y'\cos\theta. \tag{9}$$

to (5) we obtain a new formulation of the ellipse equation in the new coordinate system:

$$A'x'^2 + B'xy + C'y'^2 + D'x' + E'y' + F' = 0. \tag{10}$$

where

$$\begin{array}{l} A' = A\cos\theta^2 + B\sin\theta\cos\theta + C\sin\theta^2 \\ B' = -2A\cos\theta\sin\theta + B\cos\theta^2 - B\sin\theta^2 + 2C\sin\theta\cos\theta \\ C' = A\sin\theta^2 - B\sin\theta\cos\theta + C\cos\theta^2 \\ D' = D\cos\theta + E\sin\theta \\ E' = -D\sin\theta + E\cos\theta \\ F' = F \end{array}. \tag{11}$$

Since the ellipse oriented in parallel to the coordinate system has its B coefficient equal 0, from (11): $B' = 0$ we can compute $\omega = \theta$ satisfying:

$$\cot 2\theta = (A - C)/B \tag{12}$$

and reformulating (10) to the canonical ellipse equation:

$$\frac{(x' - S'_x)^2}{a^2} + \frac{(y' - S'_y)^2}{b^2} = 1 \tag{13}$$

we obtain:

$$a = a' = \sqrt{\frac{\frac{D'^2}{4A'} + \frac{E'^2}{4(C'-F')}}{A'}} \qquad b = b' = \sqrt{\frac{\frac{D'^2}{4A'} + \frac{E'^2}{4(C'-F')}}{C'}} \tag{14}$$

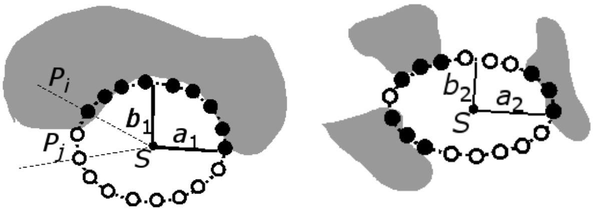
and

$$S'_x = -\frac{D'}{2A'} \qquad S'_y = \frac{-E'}{2C'} \tag{15}$$

from which, using the relations (9), we can calculate original S_x and S_y and compute all needed ellpse points from (3).

3.3 Final selection

So far, we have found a lot of ellipses representing all potential locations of the cells nuclei. Of course, many of them are redundant, located very close one to another and describing the same nucleus, others being "glued" to different cavities in the cells bodies. Such redundant, close ellipses, except one representing the entire group as the nucleus approximation, must be removed from the solution, as well as the false approximations outside the cells bodies.



(a) (b)

Fig. 4. Fitness function of 2 ellipses built from 18 points e_j, one half of them have binary value 1 (black circles) and the second half value 0 (white circles): (a) 1 sequense of 9 points, $a_1 \approx b_1$; (b) 3 sequences of 3 points each one, $a_2 \ll b_2$. The first ellipse has the fitness function value ($9^P \cdot \frac{a_1}{b_1} \approx 9^P$) higher than the second one ($(3^P + 3^P + 3^P) \cdot \frac{a_2}{b_2} \ll 9^P$).

To complete this task we need a mean to evaluate the quality of every ellipse in order to eliminate the worst ones. We propose a fitness function measuring the "completeness" of the ellipse shape. If the ellipse **E** is represented by a list of its contour points $(e_1, e_2, \cdots, e_m)$ calculated from (3), each such point can be given a binary value $v(e_j) = 0$ or 1 in function of its location on the border 'cell nucleus/cell body'. The value '1' is assigned if on the profile P_j, going from the ellipse center S to the point e_j, a pixel combination $('background','background','cell,'cell')$ is found in a local neighborhood of the theoretical location of this border (point e_j). The local neighborhood means this part of the profile P_j which is defined by the point e_j moved by length $\frac{b}{2}$ outside and inside the ellipse **E**, b being the minor ellipse axis. In the opposite case the value '0' is assigned. Among these contour points all sequences seq_k of values '1' are located:

$$Seq_k = \begin{cases} (1,2,\cdots,m) & \text{if} \quad \forall_{j=1,\cdots,m} v(e_j) = 1, \\ (f,\cdots,l) : v(e_{f-1}) = v(e_{l+1}) = 0, \forall_{j=f,\cdots,l} v(e_j) = 1 & \text{otherwise.} \end{cases} \tag{16}$$

where m is the ellipse points e_j number and all indices j are cyclic, i.e. $e_{m+1} = e_1$ and $e_0 = e_m$. The first formulation is for the case where all points e_j have value '1'. The fitness of the ellipse **E** is defined as a sum of the length of all the sequences Seq_k raised to the the power p:

$$fitness(\mathbf{E}) = \frac{\min(a,b)}{\max(a,b)} \cdot \sin\left(\frac{\max(a,b)}{R_{max}} \cdot \frac{\pi}{2}\right) \cdot \sum_k (card(Seq_k))^P. \tag{17}$$

The power $P > 1$ promotes ellipses having grouped the pixels with value '1' over those having them scattered (we use $P = 1.3$). The first multiplying factor favors symmetric ellipses (natural nuclei with circular shape, where shorter and longer ellipse axes - $min(a,b)$ and $max(a,b)$ respectively - are equal) and the second one favors ellipses of size close to $2R_{max}$.

Finally, we compare every two ellipses $\mathbf{E}_i$ and $\mathbf{E}_i$. If they are to close each to other we remove one of them with lower fitness function:

$$\begin{array}{ll} \text{if} \quad ((|S_i - S_j| < 2R_{max}) \,\text{and}\, (fitness(\mathbf{E}_i) \geqslant fitness(\mathbf{E}_j)) & \Rightarrow \quad \text{remove}(\mathbf{E}_j) \\ \text{if} \quad ((|S_i - S_j| < 2R_{max}) \,\text{and}\, (fitness(\mathbf{E}_i) < fitness(\mathbf{E}_j))) & \Rightarrow \quad \text{remove}(\mathbf{E}_i). \end{array} \tag{18}$$

We also remove all ellipses with the fitness function below some threshold, interactively and visually adjusted by the operator:

$$\text{if} \quad (fitness(\mathbf{E}) < fitness_{min}) \quad \Rightarrow \quad \text{remove}(\mathbf{E}). \tag{19}$$

4. Application

A visual application called "Smart Cell Counter" implementing 3 algorithms of counting (one based on the simple region growing and two variants of the method proposed in this work) was developed using the Java language. The language choice was done in order to allow developing and using the application under various operating system even without recompiling and to simplify the development. The program was tested in the Linux and MS Windows environments. The cost of this choice is probably slower performance in comparison to other languages like C/C++.

One of the goal of this project was creating a tool to use in the scientific work of physicians from the Department of Infectious Diseases of Medical University of Białystok. The user interface was consulted with them and it allow visual and interactive counting of the liver cells. As the main features one can mention:

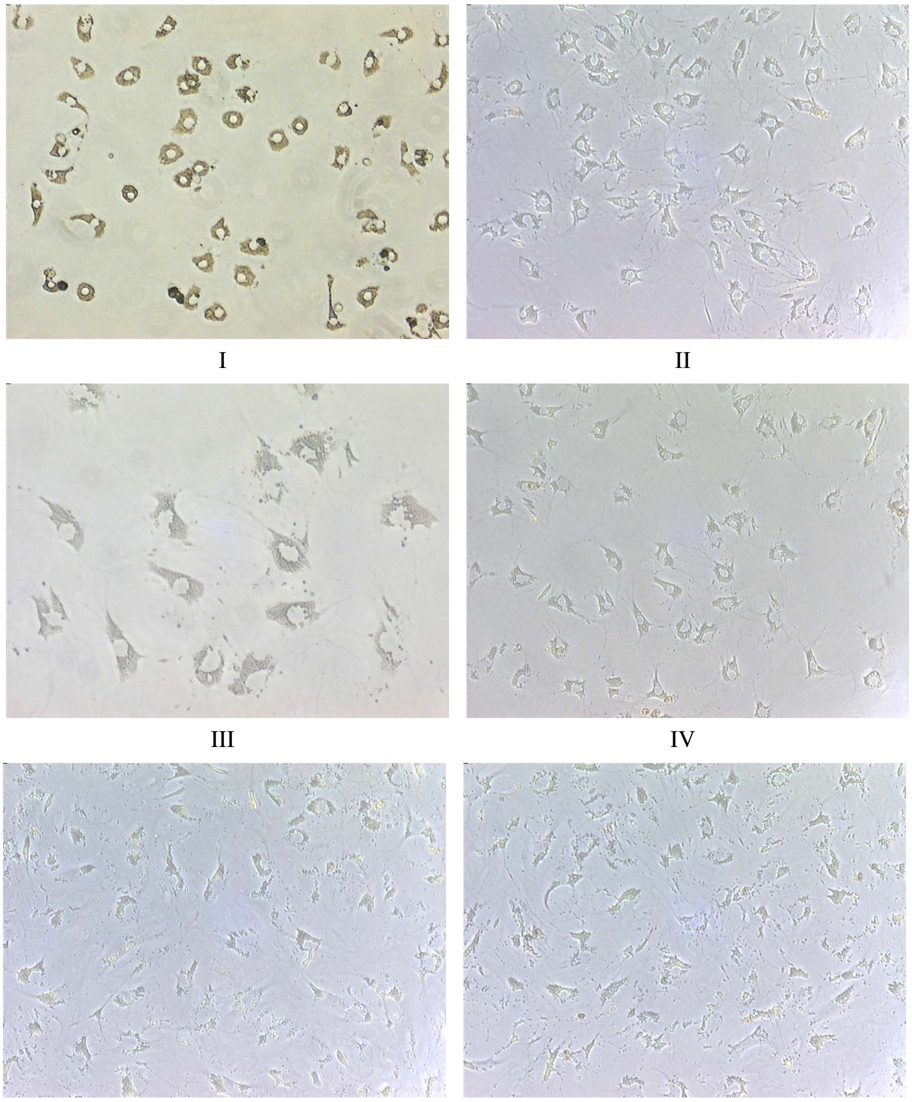
- visual and interactive setting of the threshold C (Equation (1)) during preparations of the binary image (Figure 1(d)) - the user can adjust a slider control and observe in the real time the influence of it on the image;
- visual setting of the R_{max} parameter describing the maximum nucleus radius - the user do it by defining a line with 2 mouse clicks on an example cell image;
- visual and interactive setting of the threshold $fitness_{min}$ (Equation (19)) to remove the worst ellipses - the user adjusts a slider control ad immediately sees which ellipses disappear;
- finally, the user can manually, with one mouse click, remove single ellipse judged as invalid and place a new one in a place considered the cell nucleus; however this facility is probably not to use in everyday practice since tedious when the cell number is elevated.

5. Experiments

The method was evaluated on 6 real microscopic images of the hepatic stellate cells in different stages of life (Figure 5), on which a specialist marked cells as the reference. We compare the results to the simple region growing technique where groups of connected pixels marked $'cell'$, which size is above some threshold, are counted as the liver cells. All 3 algorithms use the same binary image. For every image and every algotihm ALG ($SCC1$, $SCC2$) we computed the following statistics (Table 1):

- Ref - reference number of the ellipses (marked by the physician) $= TP + FN$;
- ALG_{TP} - cells of reference detected by the algorithm ALG, even with misplaced nuclei , since the main goal is to count the cells, no to give their exact locations (True Positive);
- ALG_{FP} - false cells detected by the algorithm ALG (False Positive);
- ALG_{FN} - cells of reference not detected by the algorithm ALG (False Negative).

We weren't able to give the True Negative statistics - in this formulation of the cell detection problem it is difficult to precise what it could mean (every pixel correctly



I

II

III

IV

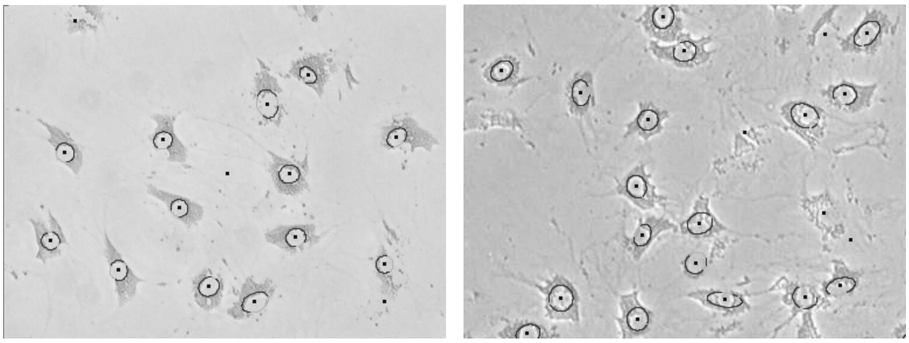
V

VI

Fig. 5. Analyzed images

rejected as a cell nucleus, every non-cell shape correctly not classified as the cell body ?).

Using these statistics the following benchmarks were computed (Table 2):



(a) (b)

Fig. 6. Example of images with found ellipses marking the cells nuclei. Darker parts of the ellipses mean detected cells bodies.

Table 1. Statistics counted for each algorithm

Image	**Ref**	**RG$_{\mathbf{TP+FP}}$**	**SCC1$_{\mathbf{FN}}$**	**SCC1$_{\mathbf{TP}}$**	**SCC1$_{\mathbf{FP}}$**	**SCC2$_{\mathbf{FN}}$**	**SC2$_{\mathbf{TP}}$**	**SCC2$_{\mathbf{FP}}$**
I	47	48	3	44	14	2	45	5
II	55	60	10	45	5	9	46	3
III	13	17	0	13	1	0	13	0
IV	46	48	3	43	4	2	44	4
V	29	53	2	27	12	7	22	10
VI	33	55	6	27	16	11	22	11

- *sensitivity* - ratio of the correctly detected cells to the to the reference cells number ($=\frac{TP}{TP+FN}$);
- $\frac{detected}{Ref}$ - how close the results of the algorithm are to the reality, in term of the cell number only - the detected cell number (ceoorely or not) will be the main result used by cphysicians in every-day work ($=\frac{TP+FP}{TP+FN}$).

Table 2. Benchmarks of the algoritmhs

Image	**RG**	**SCC1**		**SCC2**	
	$\frac{\mathbf{det}}{\mathbf{Ref}}$	$\frac{\mathbf{det}}{\mathbf{Ref}}$	**sensitivity**	$\frac{\mathbf{det}}{\mathbf{Ref}}$	**sensitivity**
I	**102.1%**	**123.4%**	93.6%	**106.4%**	95.7%
II	**109.1%**	**90.9%**	81.8%	**89.1%**	83,6%
III	**130.8%**	**107.7%**	100.0%	**100.0%**	100.0%
IV	**104.3%**	**102.2%**	93.5%	**104.3%**	95.6%
V	**182.6%**	**134.5%**	93.1%	**110.4%**	75.9%
VI	**166.7%**	**130.3%**	81.8%	**100.0%**	66.7%

The region growing algorithm was not evaluated in details. In general, if the image contains the distinct, solid cells, it performs well. However, when the cells get fragmented (e.g. images V and VI), it start detecting every cell fragment as a distinct cell and heavily overestimate their number. Also when the cells are connected, they are detected as a single entity.

Two variants of the proposed method have similar score, but the second version ($SCC2$ - ellipses reconstructed by their equations) seems closer to the reality. The main ratio $\frac{det}{Ref}$ is more often closer to 100% for $SCC2$, especially for harder to treat images (V and VI). The sensitivity is also slightly better for the second version, except for those hard V and VI images (such images, in the late stage of the cell evolution, are rarely treated by physicians, and even manual counting in this case is very imprecise). If one looked at the graphical results (ellipses drawn on the images) it could also notice better modeling of the nuclei by $SCC2$, but this is irrelevant for the simple counting task.

Calculation time The experiments were run on the AMD Athlon XP 2400+ 2.0 Ghz machine with 512 MB RAM installed. All images were treated in their original resolution 760x574 (except for the image III that had been downsized two times in their x and y dimmensions to set the cell sizes comparable between all scans) and the local neighborhood was $N = 21$ (1) . For the preprocessing stage: the Gaussian smoothing took about 1 second, local thresholds calculation - also about 1 second (thanks to their interpolation); the final binarization (2) and the small groups removal were done in the real time.

For the rest of counting in two variants, depending on the complexity of the image: it took from 8/13 seconds ($SCC1/SCC2$, the simplest image I) up to 14/17s (image II) for images of the size 760x574 and 4/3 s for the half size image III. In general the $SCC1$ is a bit faster, but the difference is not very large.

6. Conclusion

The presented method seems to give good results even when the cells to count are fragmented. Especially is such cases it shows its superiority over the simple region growing algorithm. Two variants of ellipse founding techniques perform well, with a slight advantage of the one based on the direct ellipse equation reconstruction. However, it requires more profound validation, with more images, ideally during the every-day physician work. Is should be achieved more easily thanks to the simple-to-use, graphical, multiplatform and interactive application implementing the method.

The method has been designed to count the hepaitc stellate cells, but it could be also used to work with other cells/object with the same characteristic - homogeneous, elliptical, bright central region surrounded by darker fragments.

Bibliography

[1] Chen, Y., Biddell, K., Sun, A., Relue, P.A., Johnson, J.D.: An Automatic Cell Counting Method for Optical Images; Proc. BMES/EMBS 1999 Conf., vol. 2, pp. 819, 1999.

[2] Dorini, L.B., Minetto, R., Leite, N.J.: White blood cell segmentation using morphological operators and scale-space analysis; Proc. SIBGRAPI 2007 Conf., pp. 294-304, 2007.

[3] Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct Least Square Fitting of Ellipses; IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 476-480, 1999.

[4] Friedman, S.L., Bansal M.B.: Reversal of hepatic fibrosis - fact or fantasy ?; Hepatology, 43(2 Suppl 1), pp. S82-88, February 2006.

[5] Guillon, P.: Contribution à l'évaluation de causes d'infertilité humaine d'origine masculine - Recherche de solutions optimales en segmentation, reconnaisance de form et classification appliquées à des images microscopique de spermatozoïdes; PhD Thesis, Université de Rennes I, France, 2000.

[6] He, W., Wilder, J.: Nucleus shape recognition for an automated hematology analyzing system; Proc. EMBS/BMES 2002 Conf., vol. 2, pp. 1043-1044, 2002.

[7] Korzyńska, A., Strojny, W., Hoppe, A., Wertheim, D., Hosner, P.: Segmentation of microscope images of living cells; Pattern Analysis and Applications, vol. 10, no. 4, pp. 301-319, 2007.

[8] Liao, Q., Deng, Y.: An accurate segmentation method for white blood cell images; Proc. IEEE BMI 2002 Int. Symp.., pp. 245-248, 2002.

[9] Ma, Y.D., Dai, R.L., Lian, L., Zhang, Z.F.: An Counting and Segmentation method of Blood Cell Image with Logical and Morphological Feature of Cell; Proc. ICONIP 2001 Conf., IDNumber115, `http://www.cse.cuhk.edu.hk/apnna/proceedings/iconip2001/` Shanghai, China, 2001.

[10] Pratt, W.K.: Digital Image Processing; Wiley, 2001.

[11] Refai, H., Li, L., Teague, T.K., Naukam, R.: Automatic count of hepatocytes in microscopic images; Proc. ICIP 2003 Conf., vol.2, pp. II-1101-4 vol.3, 2003.

[12] Witkowski, Ł.: Komputerowy system do śledzenia ruchu plemników, wyznaczania parametrów ruchu oraz wyliczenia gęstości nasienia; Proc. KBIB 2007 Conf., P50, 2007.

[13] Zduńczuk, M., Korzyńska, A.: Modyfikacja metod segmentacji hybrydowej pod kątem analizy obrazów komórek w hodowlach; Proc. KBIB 2007 Conf., P99, 2007.

PÓŁAUTOMATYCZNA METODA ZLICZANIA KOMÓREK GWIAŹDZISTYCH WĄTROBY NA MIKROSKOPOWYCH OBRAZACH IN-VITRO PRZEZ MODELOWANIE ICH ELIPTYCZNYCH JĄDER

Streszczenie: W artykule tym zaprezentowana jest nowa, półautomatyczna metoda zliczania komórek wątroby. Zamiast skupiać się na ciałach komórek (w niektórych fazach ich życia nieregularnych i pofragmentowanych) lokalizuje ona ich jądra, które są jasnymi, jednolitymi i eliptycznymi strukturami otoczonymi przez ciemniejsze ciało (lub jego fragmenty). Jądra komórek są modelowane na 2 sposoby: przez lokalny algorytm rozrostu obszaru i przez odtworzenie równania elipsy z jej punktów na obwodzie. Wśród wszystkich znalezionych elips (początkowa każda jej potencjalna lokalizacja jest brana pod uwagę) wybierane są lokalnie najlepsze oraz te, których dopasowanie, mierzone specjalną funkcją, jest powyżej pewnego progu. Metoda została zaimplementowana jako graficzna, wieloplatformowa, przyjazna dla użytkownika aplikacja w języku JAVA. Jej działanie zostało ocenione na rzeczywistych mikroskopowych obraz in-vitro komórek wątroby.

Słowa kluczowe: przetwarzanie obrazów medycznych, zliczanie komórek, komórki gwiaździste wątroby