Wojciech Bieniecki*, Victor Manuel Martínez González**, Szymon Grabowski*

Fine-Tuned Spot Detection on ELISPOT Images

1. Introduction

Enzyme linked immunospot assay (ELISPOT) [10, 11] is a powerful technique used for detection and quantification of antigen specific immunological responses at the single cell level. A typical ELISPOT experiment is performed in membrane bottomed 96-well plates with each well bottom (6 mm diameter) coated with antibodies specific against interferon γ. The microscope head is moving over the plate and the photo of each well is taken (Fig. 1).

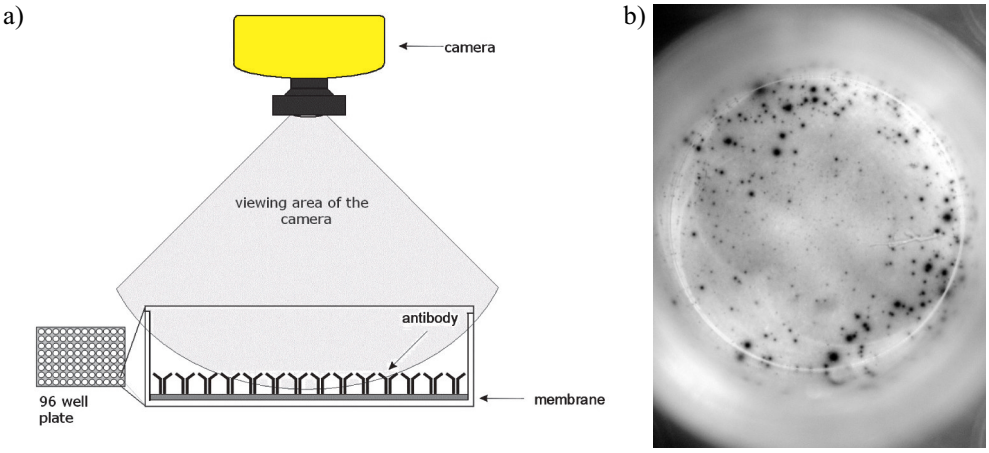

Fig. 1. The process of ELISPOT images acquisition (a) and the example of taken picture (b)

In our research project performed in collaboration with the Department of Nephrology and Transplantation Medicine of the Medical University in Wroclaw, we dealt with image processing methods, where the images were acquired from an Nikon Eclipse E-200

* Computer Engineering Department, Technical University of Lodz
** International Faculty of Engineering, Technical University of Lodz, Poland, ETSIT Universidad Politécnica de Valencia, Spain

microscope equipped with AF micro Nikkor 60 mm lenses. Our task was to automate analysis of those microscopic images containing (approximately) round spots of various colors and sizes. The output data should comprise the counts of spots according to their color and the spot area distribution (histogram).

Each spot indicates an immunological response of recipient lymphocyte with donor lymphocytes and its size and intensity reflects the strength of response. Monitoring of cellular immunological responses after renal transplantation could have a prognostic value to diagnose the long-term graft outcome (i.e., kidney) and to evaluate the level of immunosupression therapy needed.

The project was supported by a government grant (2005–2007). We had examined the following approaches:

- Bernsen's and peaks-and-valleys thresholding on grayscale converted images [1, 5],
- color space clustering [3],
- Hit-Miss segmentation [4].

The routines for detecting rough location and size of the spots and well as recognizing their color were quite successful, but the main problem was how to precisely measure the amount of the staining, which translates to the area of individual spots. The reason why it is hard are blurry edges of the spots. All the approaches listed above use thresholding at least in their initial stage and thus are very sensitive to choosing the threshold; in other words, changing the threshold slightly may severely change the measured area of a given object. In the current work we present an algorithm based on edge detection which appears to yield more stable results and the returned spot contours quite closely correspond to real ones.

2. Our algorithm

The wells on a plate have circular shapes which should be regions of interest (ROI) on the acquired images. More precisely, the ROI should enclose only a diaphragm with the spots, and the boundaries of the well should be marked as accurately as possible, because the radius of the well enables calculation of the image scale and affects further processing. We solved this problem in [2], providing a circle-finding procedure which works with sets of points detected with edge tracing.

In this work, we make use of the past experience and adapt the procedure to precisely locate and measure the actual spots. Basically, the methods using thresholding perform complete segmentation, i.e., their output are connected components corresponding to individual spots. Still, the found spot contours are inaccurate (Fig. 2).

Another possible approach is based on tracing edges on the image. This has a potential of being more accurate in returning local object boundaries, but there are two problems concerning it: an easy one is to fill the interior of the contour of the spot, and a harder one is to obtain a closed contour instead of several non-connected arcs. We follow this approach in the current work.

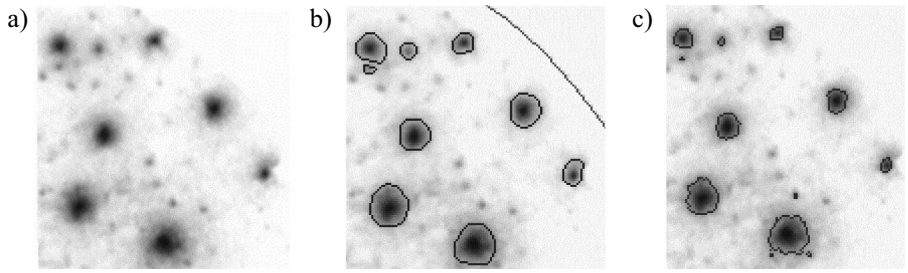

Fig. 2. The result of spot detection basing on various thresholding methods

The algorithm that we propose has the following stages:

1. resolution normalization;
2. noise reduction using low-pass filtering;
3. boundary emphasize using sharpening filtering;
4. Canny curve detection;
5. circle-fitting procedure applied to arcs;
6. reconstruction of circular spot boundaries.

We claim that the new algorithm is less vulnerable to variable image properties: image resolution, out-of-focus zones and improper exposition (non-uniform lighting). Even if the detected spot borders are severely fragmented and incomplete, still the fitting procedure may evaluate the spot circles.

2.1. Image preprocessing

In this phase we attempt to prepare the image for the Canny procedure, which performs best if the border to be traced is about 3–5 pixels wide. In examined images obtained using Nikon Camera (1182 × 1280) the ROI has about 750 pixels of diameter, and the spots diameter varies from 10 to 50 pixels. To obtain best performance the dimensions of the picture must be doubled (using bilinear interpolation).

On the resized image two procedures are imposed (Fig. 3):

– Gaussian smoothing filter with a mask diameter 5 pixels.
– Laplacian like sharpening filter with a mask diameter 3 pixels.
– linear adjustment of the image intensity histogram (for contrast equalization).

1	1	2	1	1
1	2	4	2	1
2	4	8	4	2
1	2	4	2	1
1	1	2	1	1

0	-1	0
-1	5	-1
0	-1	0

Fig. 3. Gaussian smoothing and Laplace sharpening masks

2.2. Boundaries detection

There are many approaches to edge and boundary detection. For this purpose the Canny procedure [6, 8] has been chosen, because of its benefits:

- after the proper preprocessing it generates stable results,
- is invulnerable to image rotation,
- thanks to "threshold hysteresis", it can detect and trace weak or fading lines.

The algorithm is carried out in three phases:

1. The image is filtered with the vertical and horizontal Sobel mask in order to estimate the gradient magnitude and angle (Fig. 4).
2. If the intensity gradient magnitude is greater than the threshold value t_1, the procedure of line tracing is started, and continued until the line direction changes or its strength falls below the t_2 threshold ($t_2 < t_1$),
3. The algorithm usually finds line zones (few perpendicular neighboring lines). The proper lines are extracted with aid of non-maximum suppression procedure.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Fig. 4. Sobel gradient masks

The Canny procedure generates a binary image, which is subsequently segmented into connected components (individual lines), which are then candidates for arcs of the reconstructed circles (Fig. 5).

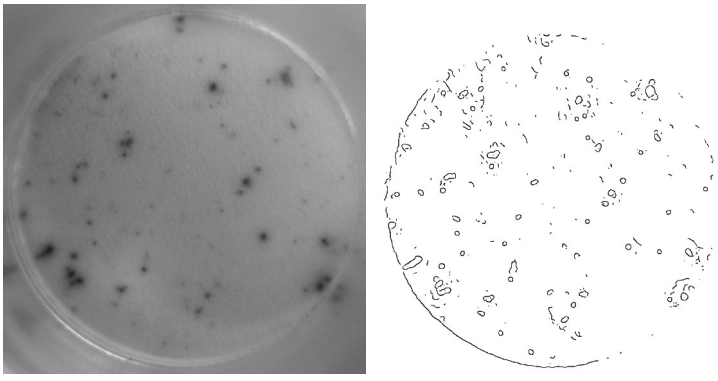

Fig. 5. The input image and after Canny operator

2.3. Arcs approximation

In this stage each connected component is considered as array given function and the analytical representation of this function has to be evaluated. The circle-finding procedure introduced in [7] runs in two steps:

1. fitting procedure using linear least squares method (the implementation from [9]),
2. iterative fine-fitting using geometric distance (the implementation form [9]).

Consider x^{T} and y^{T} as m-length vectors of coordinates of the component points. Let us denote:

$$[A]=\begin{bmatrix} x_1^2+y_1^2 & x_1 & y_1 & 1 \\ x_2^2+y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_m^2+y_m^2 & x_m & y_m & 1 \end{bmatrix} \tag{1}$$

Assuming, that m, the number of points in each connected component, is more than 4, the matrix $[A]$ is rectangular and singular. To deal with $[A]\cdot x=b$ linear equation in this case the singular value decomposition method must be applied.

The idea of decomposition of the matrix $[A]$ (M×N) is creation of column orthogonal matrix $[U]$ (M×N), diagonal matrix $[W]$ (N×N) with zero or non-zero elements and square, orthogonal matrix $[V]$ (N×N).

$$[A]=[U]\cdot\begin{bmatrix} w_{11} & & 0 \\ & \ddots & \\ 0 & & w_{nn} \end{bmatrix}\cdot[V]^T \quad \text{where} \quad [U]^T\cdot[U]=[V]^T\cdot[V]=[1] \tag{2}$$

In further processing of the fitting algorithm, the square matrix $[V]$ (4×4) is used and the circle parameters are calculated as follows:

$$x_0=\frac{-v_{24}}{2\cdot v_{14}};\quad y_0=\frac{-v_{34}}{2\cdot v_{14}};\quad r_0=\sqrt{\frac{v_{24}^2+v_{34}^2}{4\cdot v_{14}^2}-\frac{v_{44}}{v_{14}}} \tag{3}$$

When initial (x_0, y_0, r_0) values are calculated, the iterative procedure is launched.

The aim of the procedure is to minimize the square error

$$\varepsilon=\sum\left(\left\|(x,y)-(x_0,y_0)\right\|-r\right)^2 \tag{4}$$

Let us denote:

$$f = \begin{bmatrix} \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} - r_0 \\ \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} - r_0 \\ \vdots \\ \sqrt{(x_m - x_0)^2 + (y_m - y_0)^2} - r_0 \end{bmatrix} \tag{5}$$

and

$$[J] = \begin{bmatrix} \frac{x_1 - x_0}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}} & \frac{y_1 - y_0}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}} & 1 \\ \frac{x_2 - x_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} & \frac{y_2 - y_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} & 1 \\ \vdots & \vdots & \vdots \\ \frac{x_m - x_0}{\sqrt{(x_m - x_0)^2 + (y_m - y_0)^2}} & \frac{y_m - y_0}{\sqrt{(x_m - x_0)^2 + (y_m - y_0)^2}} & 1 \end{bmatrix} \tag{6}$$

The algorithm solves the equation:

$$[J] \cdot h = f \tag{7}$$

with help of QR decomposition method.

$$[J] = [Q] \cdot [R] \quad \Rightarrow \quad [R] \cdot h = [Q]^T \cdot f \tag{8}$$

where **[*Q*]** is orthogonal ($[Q]^T \cdot [Q] = [1]$) and **[*R*]** is triangular (upper). In each iteration, the circle is updated

$$(x_0, y_0, r_0) = (x_0, y_0, r_0) + h \tag{9}$$

and the error

$$\delta = \frac{\|h\|_\infty}{\|(x_0, y_0, r_0)\|_\infty} \tag{10}$$

is estimated. The iteration repeats until $\delta < \delta_0 = 10^{-5}$ or the number of iterations exceeds 100 (then the algorithm is not convergent – the line is not an arc).

2.4. Spot border reconstruction

The Canny algorithm finds the following lines:

- fragments of spots' borders (sometimes more than 1 for each spot),
- fragments of spots' interior (caused by non homogenous spot staining),
- well's boundaries.

Initially we leave only the circles for which $r \in (0.05 \cdot D; \quad 0.1 \cdot D)$ where D is the diameter of ROI. Than we delete the circles that cross the ROI circle.

It is very likely that the remaining connected components are parts of the spot contours (Fig. 6).

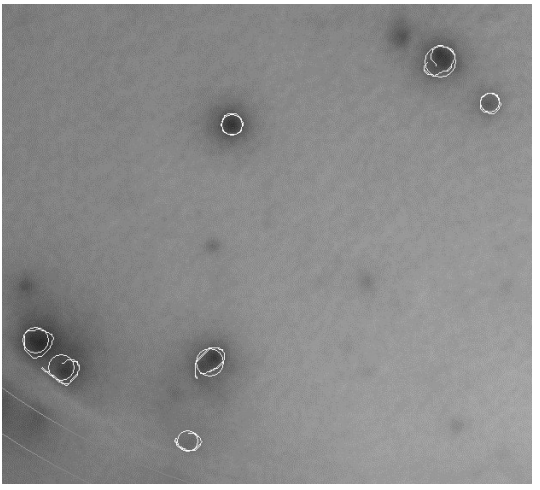

Fig. 6. The circles approximated for found spot contours

Sometimes it happens, that the spot contour is fragmented. In this case the same number of circles is produced for each spot. Circles, that overlap, i.e. the condition (11) is satisfied:

$$\rho\big((x_1, y_1), (x_2, y_2)\big) \le \min(r_1, r_2) \tag{11}$$

approximate the same contour and the arcs should be merged.

The merging algorithm works in three phases.

Phase 1: For each arc find the neighboring arcs basing on (11). Let us look at Figure 7 (the artificial spots).

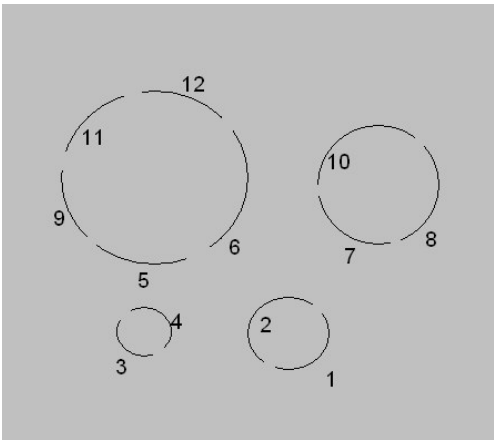

Fig. 7. The exemplary image: enumeration of found arcs

For this example, the following neighbors are found (Tab. 1).

Table 1
Object parameters for the exemplary image

Object number	Circle (x, y, r)	Found neighbors
1	(360.1, 256.1, 30.9)	2
2	(367.2, 254.2, 43.0)	1
3	(224.8, 260.7, 22.7)	4
4	(228.3, 259.8, 21.2)	3
5	(235.2, 406.5, 85.6)	6; 9; 11; 12
6	(248.5, 397.8, 71.1)	5; 9; 11; 12
7	(435.9, 391.3, 52.3)	8; 10
8	(444.0, 389.1, 47.1)	7; 10
9	(224.1, 398.4, 71.9)	5; 6; 11; 12
10	(438.8, 390.3, 55.8)	7; 8
11	(238.1, 390.4, 87.8)	5; 6; 9; 12
12	(236.4, 389.1, 87.7)	5; 6; 9; 11

Phase 2: After the process of traversing the adjacency graph we build the list of clusters (Tab. 2).

Table 2
Clusters for the exemplary image

Cluster number	Arcs numbers
1	1; 2
2	3; 4
3	5; 12; 11; 9; 6
4	7; 10; 8

In the last phase all the arcs' points for each cluster are submitted to Find-Circle procedure to obtain fine-tuned circles.

3. Results

The experiment enclose the comparison of three segmentation methods: Bernsen thresholding, Background equalization + adaptive thresholding, and Canny edge detector + arc merging. In Figure 8 we can observe the borders of segmented objects with use of three methods.

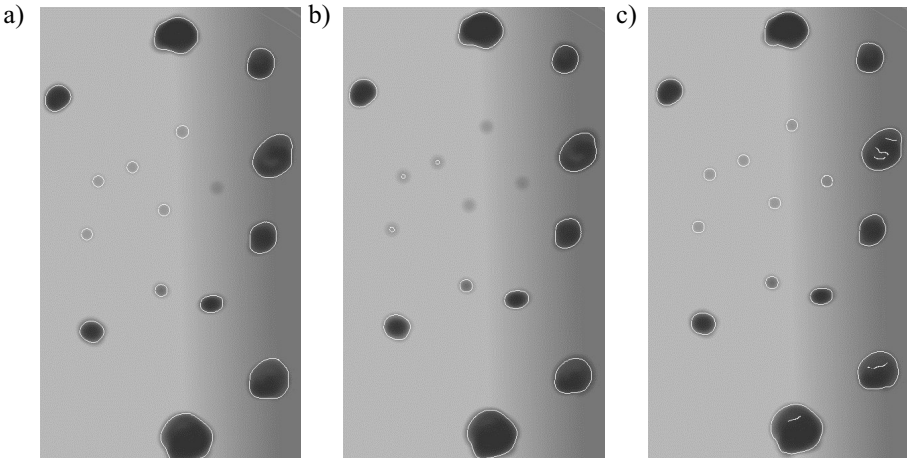

Fig. 8. The results of image segmentation

For this image the basic geometry properties: area, perimeter, Aspect Ratio and Gamma coefficient have been calculated. The results for individual spots have been printed in Table 3. From Figure 8 and Table 3 we can see that the adaptive thresholding missed some

small spots. It occurs, because small spots have weak intensity. Both Bernsen's algorithm and Canny's operator were able to detect the spots. Comparing the area value obtained for the spots for different methods, we easily notice that the edge detecting gives larger spots than thresholding (in this case). Using another thresholding algorithm or changing their parameters will give other areas. Changing the thresholds in Canny's method may introduce some irrelevant lines (edges) or suppress some important fragments of contours, but the contours will appear always in the same place. That guarantees that the objects' sizes are more stable. Proper contour detection enables accurate measurement of the object shape descriptors, which is also visible in Table 3.

Table 3
Object parameters for different segmentation types:
A – adaptive, B – Bernsen, C – Canny

Obj.	Area			Perimeter			AR			Gamma		
meth	A	B	C	117	B	C	A	B	C	A	B	C
1	1290	1127	1283	62	123	132	0.78	0.77	0.78	0.93	0.93	0.92
2	1085	874	1021	61	108	117	0.84	0.81	0.79	0.91	0.93	0.93
3	2684	222	303	60	53	62	0.68	0.86	0.86	0.88	0.97	0.98
4	–	226	300	205	54	61	–	0.88	0.92	–	0.96	0.98
5	–	214	292	64	50	60	–	0.90	0.88	–	1.00	1.00
6	2684	2503	3000	62	189	205	0.68	0.67	0.66	0.88	0.88	0.89
7	230	257	323	254	56	64	0.87	0.88	0.88	1.00	1.00	0.98
8	–	196	295	61	49	62	–	0.83	0.85	–	0.99	0.96
9	3892	3862	4572	99	237	254	0.82	0.82	0.84	0.88	0.86	0.89
10	–	194	297	62	49	61	–	0.84	0.88	–	1.00	0.98
11	733	627	720	147	93	99	0.69	0.65	0.68	0.91	0.91	0.91
12	–	216	294	203	52	62	–	0.84	0.85	–	0.98	0.95
13	1122	1313	1593	152	135	147	0.85	0.78	0.80	0.91	0.90	0.92
14	1211	1373	2969	209	139	203	0.79	0.73	0.76	0.91	0.88	0.90
15	1926	2486	1642	172	188	152	0.74	0.78	0.68	0.81	0.88	0.89
16	2027	2655	3101	173	196	209	0.69	0.67	0.70	0.84	0.86	0.89

4. Conclusions

We proposed a new approach to spot detection and measurement on ELISPOT microscope images, combining the Canny operator and a circle-fitting routine that we previously

used (in somewhat different context) for that class of images. Good edge detection algorithms, like the Canny one, yield locally accurate spot contour arcs but their drawback is that those arcs are not connected, i.e., do not constitute full contours. We solved this problem, fitting arcs belonging to the same spot contour. For each arc, a circle to each it belongs was approximated and then circle clustering was performed not to produce too many (false) contours.

References

[1] Bieniecki W., Grabowski S., Sankowski D., Kościelska-Kasprzak K., Bernat B., Klinger M., *An Efficient Processing and Analysis Algorithm for Images Obtained from Immunoenzymatic Visualization of Secretory Activity*. Proceedings of the 8th. International IEEE Conference CADSM 2005, Lviv-Polyana, Ukraine, 2005, 458–460.

[2] Bieniecki W., Grabowski Sz., Kościelska-Kasprzak K., Drulis-Fajdasz D., Mazanowska O., Klinger M., *An Algorithm for Smart ROI Detection in ELISPOT Examination Images*. Proc. of 15th National Conf. „Sieci i Systemy Informatyczne", Łódź, Oct. 2007, 133–136.

[3] Bieniecki W., Krupiński M., Grabowski S., Kościelska-Kasprzak K., Drulis-Fajdasz D., Mazanowska O., Klinger M., *Unsupervised two-color ELISPOT image segmentation based on k-means clustering*. Proceedings of International Conference TCSET'2008, Lviv-Slavsko, Ukraine, 2008, 368–372.

[4] Bieniecki W., Gródecki A., Grabowski S., Kościelska-Kasprzak K., Drulis-Fajdasz D., Mazanowska O., Klinger M., *Zastosowanie algorytmu Hit-Miss do segmentacji barwnych obrazów mikroskopowych w badaniu metodą ELISPOT*. Automatyka (półrocznik AGH), t. 12, z. 3, 2008, 585–597, ISSN: 1429-3447.

[5] Bieniecki W., Węgrzyn M., Grabowski S., Kościelska-Kasprzak K., Drulis-Fajdasz D., Mazanowska O., Klinger M., *Zastosowanie algorytmów progowania adaptacyjnego do segmentacji barwnych obrazów mikroskopowych w badaniu metodą ELISPOT*. Automatyka (półrocznik AGH), t. 12, z. 3, 2008, 599–608, ISSN: 1429-3447.

[6] Canny J.F., *A computational approach to edge detection*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, no. 6, 1986, 679–698.

[7] Gander W., Golub G.H, Strebel R., *Least-Squares Fitting of Circles and Ellipses*. BIT Numerical Mathematics, Springer, 1994.

[8] Grigorescu C., Petkov N., Westenberg M.A., *Contour and boundary detection improved by surround suppression of texture edges, Image and Vision Computing*, vol. 22, no 8, 609–622, 2004.

[9] Press W.H., Teukolsky S.A., Vetterling W. T., Flannery B.P., *Numerical recipes in C*. Cambridge University Press, Cambridge, 2002.

[10] Tary-Lehmann M., Hricik D.E., Justice A.C., Potter N.S., Heeger P.S., *Enzyme-linked immunosorbent assay spot detection of interferon-gamma and interleukin 5-producing cells as a predictive marker for renal allograft failure*. Transplantation, vol. 66, 219–224, 1998.

[11] Versteegen J.M., Logtenberg T., Ballieux R.E., *Enumeration of IFN-gamma-producing human lymphocytes by spot-ELISA. A method to detect lymphokine-producing lymphocytes at the single-cell level*. J. Immunol. Methods, vol. 111, 25–29, 1998.