AUTOMATIC DETECTION OF CELL DIVISION INTENSITY IN BUDDING YEAST [1]

K. Nekrasov[2], D. Laptev[3], D. Vetrov[4]

[2] MSU CMC, knek@list.ru
[3] MSU CMC, laptev.d.a@gmail.com
[4] MSU CMC, vetrovd@yandex.ru

In different areas of biology the evaluation of cell division intensity plays an important role. The problem of automatic quantitative estimation of the intensity is presented in the paper. We process the images of cells by binariazation via LoG, and classification of objects to cells, buds and noise. We also formulate the final criteria for calculation of the budding index.

1. Introduction

In microbiology, molecular biology and yeast genetics research the important aspect is the evaluation of reproduction (budding) intensity under various factors, either in the cell cultures with contributed mutations, or certain modifications of gene expression. Budding intensity is defined by the number of observable dividing cells per total number of observable cells. Traditionally, budding index is evaluated by manual processing of images with cell culture.

Obviously, manual computation cause lower reliability of the results and requires a lot of effort from experts. In the paper we present an algorithm which processes the images of yeast cells automatically and estimates the budding intensity. This problem is also discussed in [1,2]. Fig. 1 illustrates the main steps of our method. First, we convert the source image to a binary form, in which each connected component represents easily recognized group of cells. For buinarization we use Laplacian of Gaussian [3,4] with subsequent post processing. Also we perform noise filtration and region filling [5,6]. In the second step, we produce the way to classify connected components of the binarized image to cells and buds. And in the third step, we evaluate the budding index according to information obtained on the second step. In the end of the paper we compare the results of our method with the results of manual processing which was done by biologists.

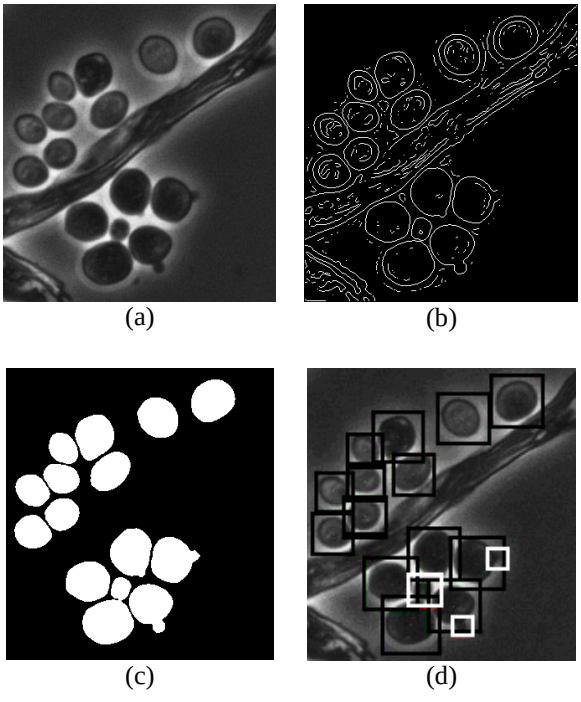

Fig. 1. Illustration of main steps of the algorithm: (a) a part of an original image; (b) the results of LoG method; (c) image after applying borders closure, filling and filtration; (d) the resulting image with identified cells (in black squares) and buds (in white squares).

2. Binarization

For image binarization at the first step we use the variation of one of the well-known methods for edge detection: Laplacian of Gaussian (LoG) [3,4] with subsequent closure of discontinuous contours. The LoG method finds edges by finding zeros after filtering the source image with a LoG filter. LoG filter is created from following formulas:

$$h_g(x,y) = e^{-(x^2+y^2)}/(2\sigma^2)$$

$$h(x,y) = \frac{(x^2+y^2-2\sigma^2)h_g(x,y)}{2\pi\sigma^6 \sum_i \sum_j h_g(i,j)}$$

2D LoG filter of size 5x5 with standard deviation equals to `0.5` can be approximated by the following matrix:

$$\begin{bmatrix} 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0564 & 0.7146 & -4.9048 & 0.7146 & 0.0564 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \end{bmatrix}$$

In our method, we use LoG with kernel (1) and threshold, which is estimated according to equations (2,3).

$$K - LoG\ kernel\ with\ \sigma = 2,\ size = 13 \times 13 \quad (1)$$

$$B = filter(I, K - \frac{\sum_{i,j} K_{i,j}}{size^2}) \quad (2)$$

$$thresh = \frac{0.75}{|B|} \sum_{i,j} |B_{i,j}| \quad (3)$$

Here K – LoG kernel; I – grayscale view of the source image, which is represented as matrix; function B=$filter(A,K)$ - applies filter K to image A and returns resulted image in B; $|B|$ – total amount of elements in matrix B; $|B_{i,j}|$ is absolute value of a number in matrix B placed by index (i,j);

Result of applying LoG method to an image on Fig. 1(a) is shown on Fig. 1(b).

LoG without the closure doesn't give us suitable results, because the borders of some cells are not continuous and that doesn't allow us to easily find the individual cells in the following steps of the algorithm.

To get rid of the borders discontinuity we look for the endpoints across the image [7].

Most of the images have areolas around each cell, due to the specifics of lightening. This leads to the effect of doubled borders (the first border is around the cell and the second one is around the areola) which can be clearly seen in Fig. 1(b).

That areolas leads us to the following problem. If we connect all the endpoints that are within a certain area, then the continuous contour can contain fragments of areola. For the next steps of the algorithm curves must denote only cell borders.

We noticed that the areolas are characterized by a high density of endpoints. So it makes sense to consider only those places on image that are characterized by low density of endpoints to identify the borders of the cells. If the region around some endpoint has small number of other endpoints then we connect this point to the others in this region.

Formally this implies the following
let B be the binary image, EP be the set of endpoints,
$E_{p,r}$ be the set of endpoints within radius r from the point p. Then $\forall p \in EP: |E_{p,r}| \leq \varepsilon, \forall q \in E_{p,r}$ connecting points p and q on the binary image B. Thresholds $r = 6,\ \varepsilon = 3$ were selected manually.

All the closed borders on the picture represent the contour of a group of cells. For the next step we fill all enclosed areas [6]. Next we use median filter with 3-by-3 neighborhood to delete all the remaining noise on the image [5]. Each output pixel of filtered image contains the median value in the 3-by-3 neighborhood around the corresponding pixel in the input binary image (see Fig. 1(c)).

3. Regions classification

In this step we separately classify each of the connected regions in the filled image. For the following processing, we need to set four thresholds - minimum and maximum radiuses of a bud and of a cell (r_1, r_2, R_1, R_2).

We consider a single connected component D with the boundary ∂D (see Fig. 2(a)) and area S_D (area of a cell with minimum radius). Components with areas less than $S_D \leq \pi R_1^2$ we consider as noise.

We inscribe a variety of circles within the component D, which touch contour ∂D at least in one point (see Fig.2(b)). Any circle that is completely covered by another can be thrown from further consideration. This optimization significantly reduces the number of circles which are treated as potential cells or buds.

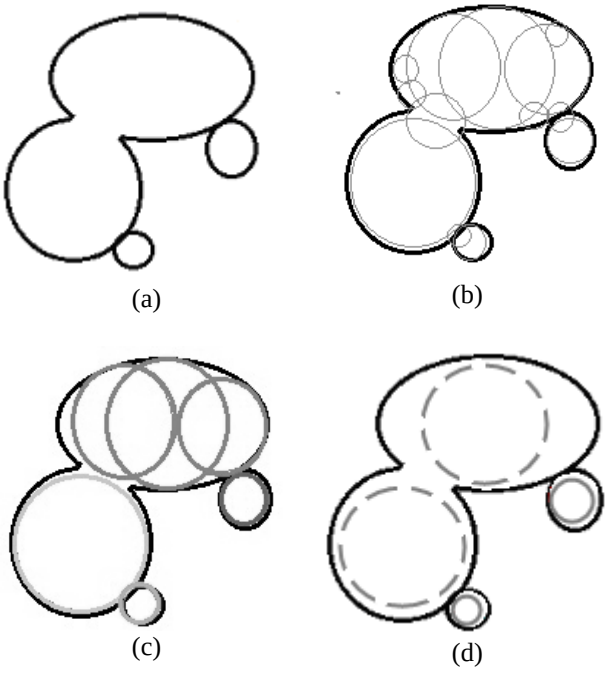

Fig. 2. Illustration of main steps of the classification procedure: (a) sketch of original contour ∂D; (b) contour with inscribed circles inside; (c) sketch of contour with grouped circles – different groups shown in different shades of gray; (d) sketch of result image with classified domains: gray line – buds, gray dashed line – cells.

Let us define the function $\rho(p, \partial D), p \in D$, which is equal to the Euclidean distance from the point p to contour ∂D. It can be easily calculated for each point of D [8]. Also it is obvious that every point p is actually the center of the inscribed circle with a radius $r(p) = \rho(p, \partial D)$. We also define $r(p) = 0$ $\forall p \in R^2 \setminus D$.

To use the optimization mentioned above, we define the function $f(p) = r(\rho) \forall p \in D$: $\forall q : | p - q | = 1 \Rightarrow f(p) \geq f(q)$, and $f(p) = 0$ for other points. The definition of function *f(p)* allows us to reduce the number of zeros at least by 50%.

Then we consider the circles, defined by function f. We split all circles to classes according to their radius R: cells ($R_1 \leq R \leq R_2$), buds ($r_1 \leq R \leq r_2$), small noise ($R \leq r_1$), average noise ($r_2 \leq R \leq R_1$) and big noise ($R \geq R_2$). After that we exclude from consideration all the circles of the small noise class.

Suppose we have big circle R and small circle r ($R > r$), l - distance between their centers. Let us say that the big circle overlaps the small one with degree C if $\frac{r-l+R}{2r} \geq C$.

Then we remove all circles from buds class, which are overlapped by cells circles, small noise and large noise with degree C_{buds} part. We take $C_{buds} = 0.5$.

Next, we remove all circles from cells class, which are overlapped by large noise on C_{cells} part. We take $C_{cells} = 0.25$.

After that we exclude all the noise from further consideration, so only buds and cells classes remain. For more accurate separation, we unite circles from one class in groups by its distance from each other and difference in radiuses.

We unite all the circles to groups using the following procedure: two circles of one class belongs to one group if the biggest circle overlaps the smallest one with degree $P_{cells} = 0.25$ (for circles of cells class) or $P_{buds} = 0.5$ (for circles of budds class). The result of the procedure is shown in Fig. 2(c). This procedure can be done effectively by using disjoint sets described in [9]. Parameters were selected manually.

When groups are formed we characterize each group by its average circle:

$$\forall group\ G, average\ circle_G = \sum_{circles \in G} \frac{circles}{|G|}$$

Let us call average circles with radius $r \in [r_1, r_2]$ the potential buds, and the ones with radius $R \in [R_1, R_2]$ – the potential cells. With that, all circuit regions in component D are classified (Fig. 2(d)).

4. Final calculation

For the final calculation of number of separate cells and budding cells on original image (Fig. 1(a)) we use following formulas:

$$\forall \textit{connected region } D \in B, \textit{where } B - \textit{binary image},$$

$$buds_D = \min(potential\ cells_D, potential\ buds_D) \quad (4)$$

$$cells_D = \max(potential\ cells_D - potential\ buds, 0) \quad (5)$$

$$mitosis = \sum_{D \in B} buds_D \quad (6)$$

$$simples = \sum_{D \in B} cells_D \quad (7)$$

$$index = \frac{mitosis}{mitosis + simples} \quad (8)$$

Formulas (4,5) express the number of simple cells and budding cells on single connected component D. Formulas (6,7) count the number of simple and dividing cells on whole binary image B. And equation (8) counts the percentage of budding cells from all the cells on image.

5. Results

We analyse a set of images of cell colonies given us by biologists (Fig. 3). As we can see on the chart, the results given by our algorithm shows very good correposndence with the results of manual processing.

The analysis of three cell cultures, measured in different biological conditions, had shown that the proposed method results are different from expert partitioning for less than 6%. In 22 of 23 images the deviation was less than 4%.

The realization of the algorithm in MathLab® consumes about 15 seconds per photo with resolution 2560x1520 and high density of a cells. The manual processing requires about 5 minutes per image since typical image contains several hundreds of cells.

6. Conclusion

We have described the algorithm, which permits to automatically evaluate cell division intensity and to locate cells and buds on photos of yeast culture. Our method is shown to be fast and effective and competitive to the results of manual analysis which is very time-consuming and has high reuirements the the skill level of the personnel. The algorithm is currently being used in scientific research in the biological department of Moscow State University.

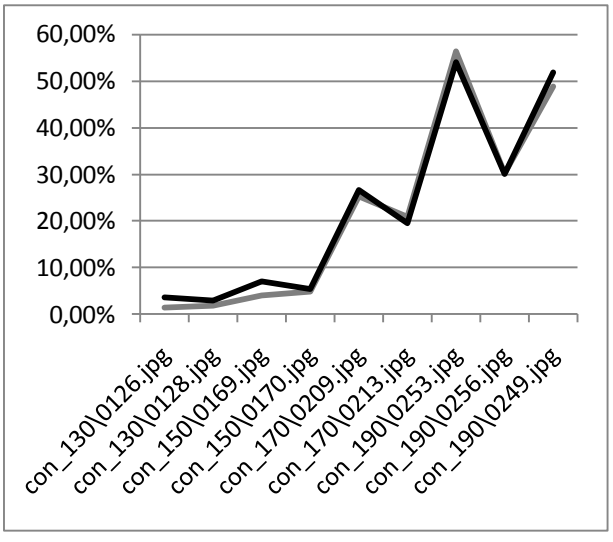

Fig. 3. The results of automatic and manual processing of cell images. Black line – automatic analysis, gray line – the results obtained by manual estimation of the budding intensity Vertical axis represents the percentage of budding cells from all the cells on photo.

References

1. D. Laptev, "Automatic counting of cells division intensity via microscope image analysis", Graphicon 2009
2. M. Kvarnstrom, K. Logg, A. Diez, K. Bodvard, M. Kall, "Image analysis algorithms for cell contour recognition in budding yeast". OPTICS EXPRESS. – 2008. - Vol. 16, No. 17
3. Om Pavithra Bonam and Sridhar Godavarthy, "Edge Detection", University of South Florida, Computer Vision (CAP 6415 : Project 2)
4. Tony Lindeberg, "Edge detection and ridge detection with automatic scale selection", Technical report ISRN KTH NA/P–96/06–SE, 1996.
5. Ben Weiss, "Fast median and bilateral filtering" Technical report Shell & Slate Software Corp., 2006.
6. Soille, P., "Morphological Image Analysis: Principles and Applications", Springer-Verlag, 1999.
7. Kong, T. Yung and Azriel Rosenfeld, "Topological Algorithms for Digital Image Processing", Elsevier Science, Inc., 1996.
8. Shuqun Zhang and Mohammad A. Karim, "Euclidean Distance Transform by Stack Filters", IEEE SIGNAL PROCESSING LETTERS, VOL. 6, NO. 10
9. Cormen Thomas, "Introduction to algorithms", 2001

[1] The work was supported by the Russian Foundation for Basic Research (project 08-01-00405).