Touching Cells Splitting by Using Concave Points and Ellipse Fitting

Xiangzhi Bai[1,2], Changming Sun[2], Fugen Zhou[1]
[1]*Image Processing Center, Beihang University, Beijing 100083, China*
jackybxz163@163.com
[2]*CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde, NSW 1670, Australia*
changming.sun@csiro.au

Abstract

A novel touching cells splitting algorithm by using concave points and ellipse fitting is proposed to split circle-like or ellipse-like touching cells in this paper. The algorithm is divided into two parts. The first part is contour pre-processing, whose purpose is to find the concave points of the contour and separate the contour into different segments using the concave points. The second part is ellipse processing, whose purpose is to process the different segments of the contour as a possible single cell by using the properties of the fitted ellipse. The second part is the main contribution of this paper which demonstrates a new way to use ellipse fitting to split the binary contour of the touching cells. Experimental results show that the proposed algorithm is efficient.

1. Introduction

Touching objects splitting is important in image analysis. Various splitting algorithms are proposed to split different types of touching objects according to different applications [1-9]. Some of the algorithms are on touching cells splitting [1, 2, 6, 7, 9].

To efficiently split touching cells, many algorithms based on various prior knowledge or various mathematical tools are proposed [1-7, 9]. The shape based algorithms [3-5] split the touching cells through analysing the concavity of the contour of the touching cells. Because the binary image of the touching cells can not be ideally obtained, the contour of the touching cells usually has many false concave regions, which will affect the efficiency of these shape based algorithms. Watershed based algorithms [1, 10] are also efficient for some applications. But, when the cells heavily touch each other or there is no strong gradient existing in the touching cells, the watershed based algorithms usually have the problem of over-segmentation or under-segmentation. Some morphological operation based algorithms [2, 11] are also proposed to split the touching cells. These algorithms split the touching cells through carrying out different mathematical morphology operator sequences on the image to find the gradient of the touching cells. Obviously, when the gradient of the touching cells is not strong, these algorithms will not perform well. The global minimization based method [9] is also efficient for touching cells splitting in some cases. But, if there are many cells which heavily touch each other, the markers the method should have are usually not easy to get. To split the circle-like or ellipse-like touching cells, some ellipse fitting based algorithms are proposed [6-8]. These algorithms apply the ellipse fitting by using different information to fit the possible single cell in the touching cells as a single ellipse and achieve the purpose of touching cells splitting. Although these algorithms have good performance in some cases, they do not perform well on cell contours with fluctuation or having un-usual shape. Many cells have circle-like or ellipse-like shapes, and the touching cells in our application have circle-like or ellipse-like shapes. So, we propose a new and efficient ellipse fitting based algorithm in this paper to split the touching cells.

Our algorithm uses concave points and ellipse fitting to process the contour of the touching cells and split the contour into different single ellipses which represent different cells. The algorithm mainly focuses on the splitting of the touching cells, and the initial binary contour can be obtained by various algorithms such as thresholding method, watershed transformations and so on. The algorithm is divided into two parts, first is the contour pre-processing, second is the ellipse processing. The purpose of the contour pre-processing is to find the concave points of the contour and separate the contour into different segments by using the concave points. The purpose of ellipse processing is to process different segments of a contour for cell splitting by using the properties of the fitted ellipse. The second part is the main contribution of this paper. It demonstrates a new way to use the ellipse fitting to split the binary contour of the touching

978-0-7695-3456-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DICTA.2008.11

cells. Experimental results show that our proposed algorithm is efficient at splitting circle-like or ellipse-like touching cells.

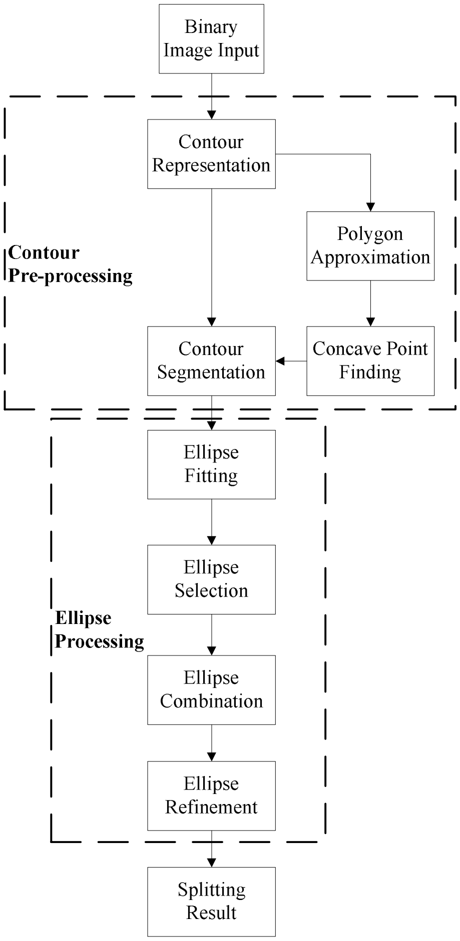


Fig. 1 Flow chart of the proposed algorithm

2. Algorithm

The whole algorithm is demonstrated in Fig. 1. As Fig. 1 shows, the algorithm is divided into two parts: contour pre-processing and ellipse processing. Details of the proposed algorithm are explained below.

2.1 Contour pre-processing

The main purpose of contour pre-processing is to segment the cell contour into segments by using the concave points so that the ellipse processing procedure may be carried out to find multiple cells within the contour.

The contour of the cell C is firstly represented by a sequence of points, so that the calculation of the concavity of each contour will be easy.

Then, the contour (C) of the touching cells is processed through polygon approximation [3] to smooth the small-scale fluctuations before finding the concave points. These small-scale fluctuations may affect the detection of the concavity and may cause false concave points. So, they should be smoothed. Let PAC represent the resulting contour after polygon approximation.

To find the real concave points, the concavity and the property of the region are used in this paper.

Let $p_c(x_c, y_c)$ represent the current point on PAC, $p_{pre}(x_{pre}, y_{pre})$ and $p_{next}(x_{next}, y_{next})$ represent the previous and the next points of p_c on PAC, respectively. Then, the concavity of the contour at point p_c, $concavity(p_c)$, can be calculated as follows:

$$a(p_{pre}, p_c)=\tan^{-1}((y_{pre}-y_c)/(x_{pre}-x_c)) \qquad (1)$$

$$a(p_{next}, p_c)=\tan^{-1}((y_{next}-y_c)/(x_{next}-x_c)) \qquad (2)$$

$$concavity(p_c)=$$

$$\begin{cases} \left|a(p_{pre},p_c)-a(p_{next},p_c)\right|, if \left|a(p_{pre},p_c)-a(p_{next},p_c)\right| < \pi \\ \pi-\left|a(p_{pre},p_c)-a(p_{next},p_c)\right|, else \end{cases} \qquad (3)$$

The real concave point p_c should satisfy the following two rules:

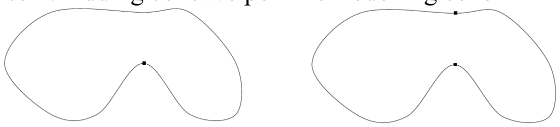
(1) $a_1 < concavity(p_c) < a_2$; (4)

(2) Line $\overline{p_{next}p_{pre}}$ should not pass through the inside of the cell regions.

The first rule indicates that if point p_c has a value of $concavity(p_c)$ in the range (a_1, a_2), p_c should be a possible concave point. a_1 and a_2 are pre-set thresholds which can be given through prior knowledge or some training programs. But, because of the fluctuations in the contour of the cell, some false concave points may also have their $concavity(p_c)$ in (a_1, a_2). To avoid the effect of the fluctuations, the second rule is used. The second rule indicates that only the region of the cell which is really concave may possess a real possible concave points. So, the two rules indicate real possible concave points.

After finding concave points, some special cases are considered to add more concave points.

Case 1: Adding concave point for touching cells



(a) Original concave point (b) After concave point adding

Fig. 2 Adding concave point for touching cells

In some cases, two touching cells may only have one obvious concave point because of the position they touched (Fig. 2(a)). In order to accurately segment the contour, another concave point is added on the opposite side of the contour. The position of the adding point can usually be the half index of the point sequence starting from the original concave point. Because the same type of cells usually has similar size, this concave point adding is appropriate in most cases (Fig. 2(b)).

Case 2: Adding concave point for incomplete cells

Some cells do not have complete contours because they are near the edge of the image and some parts of the cells are cut out of the image. In this case, two concave points are added at the start and the end of the cell contour.

After finding all the concave points, these concave points divide the contour into different segments. Let $L_i=\{p_{i1}, p_{i2}, \ldots, p_{is}\}$ represent one segment of the contour C; s is the number of points on L_i; p_{i1} and p_{is} are two concave points which are used to segment C and form L_i. Let m represent the number of the concave points. That is:

$$C= L_1 + L_2 +\ldots+ L_m. \tag{5}$$

Then, the whole contour is segmented into different segments. Each of these segments can be used for ellipse fitting, and then the fitted ellipses are selected, combined and refined to split the touching cells.

2.2 Ellipse processing

Each image may have multiple touching cells. And, each contour of the touching cells may have multiple segments. The main purpose of ellipse processing is to split the touching cells through processing all the segments of the contour as the correct ellipse which represents one real separated cell. Details of the ellipse processing are given below.

2.2.1 Ellipse fitting

Ellipse fitting has been widely used for object recognition. Most of the cells in our data have circle-like or ellipse-like shape. So, an ellipse can be used to represent a cell. The contour of the touching cells has been separated into different segments. If some segments belong to the same cell, each ellipse fitted by using the points of each segment should have some similar properties. Then, these ellipses can be processed and merged into one ellipse, which achieves the purpose of touching cell splitting. But, before the operation of ellipse processing, each segment should be fitted by an ellipse. The direct ellipse fitting method [12] is used in this paper to fit the segments.

A general conic can be represented by an implicit second order polynomial [12]:

$$F(\boldsymbol{a}, \boldsymbol{x})=\boldsymbol{a} \bullet \boldsymbol{x}=ax^2+bxy+cy^2+dx+ey+f \tag{6}$$

where $\boldsymbol{a}=[a, b, c, d, e, f]^T$, $\boldsymbol{x}=[x^2, xy, y^2, x, y, 1]^T$. $F(\boldsymbol{a}; \boldsymbol{x_i})$ is called the "algebraic distance" of a point (x, y) to the conic $F(\boldsymbol{a}; \boldsymbol{x})=0$. The fitting of an ellipse can be approached by minimizing the sum of squared algebraic distances:

$$D_A(\boldsymbol{a})= \sum_{i=1}^{N} F^2 (\boldsymbol{a}; \boldsymbol{x_i}) \tag{7}$$

of the ellipse curve to the N data points $\boldsymbol{x_i}$ [12].

The direct least square method formulates the ellipse fitting as

$$\text{minimizing } E=\|\mathbf{D}\boldsymbol{a}\|^2 \tag{8}$$

subject to the constraint

$$\boldsymbol{a}^T\mathbf{C}\boldsymbol{a}=1 \tag{9}$$

where

$$\mathbf{D}=[\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}]^T \tag{10}$$

$$\mathbf{C}=\begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

By introducing the Lagrange multiplier λ and differentiating, (8)-(11) can be rewritten as the following linear system

$$\mathbf{D}^T\mathbf{D}\boldsymbol{a}= \lambda\, \mathbf{C}\boldsymbol{a} \tag{12}$$

$$\boldsymbol{a}^T\mathbf{C}\boldsymbol{a}=1 \tag{13}$$

The parameters of an ellipse, which are $\boldsymbol{a}=[a, b, c, d, e, f]^T$, can be obtained through solving the system equation. Then, the fitted ellipse is given by the expression (6) and $\boldsymbol{a}=[a, b, c, d, e, f]^T$.

2.2.2 Ellipse selection

After ellipse fitting, each segment of the contour has a fitted ellipse. If the fitted ellipse does not satisfy the following two conditions, the corresponding segment will be removed from the candidate ellipses for ellipse combination step and will be processed at the ellipse refinement step. The examples of these segments most likely include short segments of the contour.

(1) The algebraic distance $F(\boldsymbol{a}; \boldsymbol{x_{ij}})$ is used here as one basic measure. $\boldsymbol{x_{ij}}$ represents the vector of point p_{ij} on segment L_i. Let $dis(\boldsymbol{a}; L_i)$ be the mean distance between the points in L_i and the fitted ellipse $F(\boldsymbol{a}; \boldsymbol{x})$. If $dis(\boldsymbol{a}; L_i)$ is larger than a pre-set threshold *disTh*, the corresponding ellipse should be removed. Because the fitted ellipse is much different from the segment, it is easy to select *disTh* following prior knowledge or some training programs.

(2) The cells are usually not too slender. So, the ratio (*eRatio*) of the minor axis (*MinA*) and major axis (*MaxA*) of the fitted ellipse should not be too small. Then, if *eRatio* is too small, the corresponding ellipse

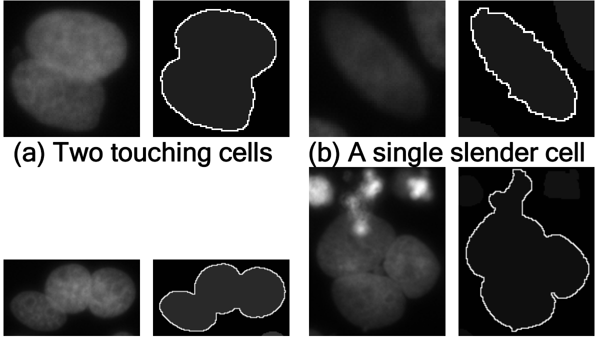
should be removed. Each type of cell has similar properties, so it is easy to constrain *eRatio* of the current type of cells through prior knowledge.

After the ellipse selection, the remaining ellipses fit the corresponding segments well, and they act as the candidate ellipses for ellipse processing.

2.2.3 Ellipse combination

Some contour segments belong to the same cell in the touching cells. Then, if these ellipses, whose corresponding contour segments belong to the same cell, can be combined, the touching cells can be separated.

In this step, every pair of ellipses of the current contour are combined to generate a new combined ellipse by carrying out ellipse fitting on the points of the two segments which belong to the two original ellipses. The newly fitted ellipse is firstly verified through the procedure of ellipse selection. If the newly fitted ellipse is an appropriate ellipse for the two segments, then the newly fitted ellipse and the two previous ellipses are verified by using different rules under various cases to determine whether the two previous ellipses should be combined as the newly fitted ellipse. If the newly fitted ellipse is not an appropriate ellipse for the segments, the newly fitted ellipse will be rejected and the two previous ellipses will be kept. So, before we discuss the algorithm of the ellipse combination, the cases of the touching cells should be analysed to find effective rules. Some of the touching cells and their corresponding contours are demonstrated in Fig.3. Most of the touching cells can be classified into the cases given in Fig. 3.


(a) Two touching cells (b) A single slender cell

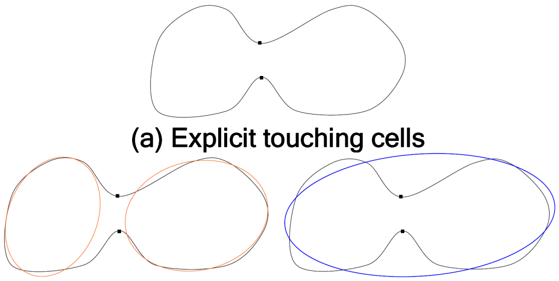
(c) Three touching cells (d) Four touching cells

Fig. 3 Examples of touching cells

Following Fig. 3, there are five cases which should be considered.

Case 1: Explicit touching cells

If the cells are explicitly touching, the ellipse should not be combined. Fig. 4(b) is the result of the ellipse fitting for each segment of the contour of Fig. 4(a). Fig. 4(c) is the result of the newly fitted ellipse. Fig. 4 shows that, three rules can be used to identify this situation.


(a) Explicit touching cells

(b)Two previously fitted ellipses

(c) Newly fitted ellipse

Fig. 4 Ellipse fitting for Case 1

(1) The distance between the center of the newly generated ellipse (C_{new}) and the centers of the two previous ellipses (C_1 and C_2) should be larger than *dMinTh*. It is easy to determine *dMinTh* following the properties of the same type of cells. *dMinTh* is usually close to the length of minor axis of the smallest cell. This rule indicates that the two ellipses are indeed touching cells. Then, they should not be combined.

(2) The mean algebraic distance of all the points in the contour C to the newly generated ellipse E_{new} should be larger than *disTh*. This rule indicates that the newly generated ellipse has large fitting error comparing with the two previous ellipses.

(3) The distance between the centers of the two previous ellipses (C_1, C_2) should be very large.

$$\| C_1 - C_2 \| > (2.5\sim4.0) dMinTh \qquad (14)$$

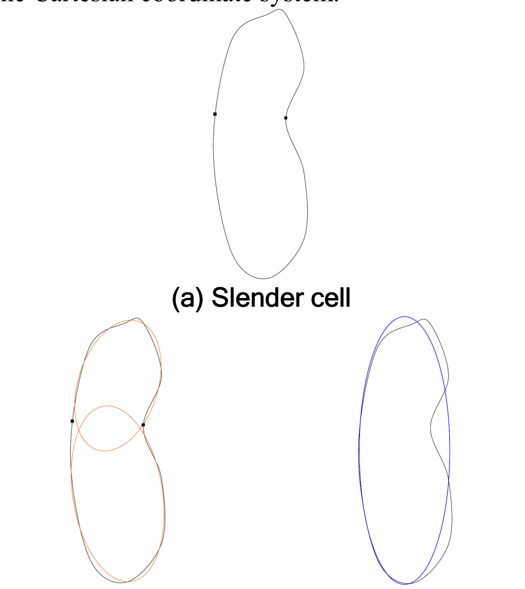
This rule indicates that the two ellipses should be separated. The exact value between 2.5~4.0 should be selected following prior knowledge or some training program.

The three rules of Case 1 indicate that the real touching cells should be separated. So, the two ellipses will not be combined.

Case 2: The slender cells

Although the same type of the cells have similar shape, some cells have more slender shape than other cells during cell evolution. Especially when the cell is in the cycle of cell division, the cell is very slender and may have concave points because of the fluctuation of the contour. This case is shown in Fig. 5(a).

Because of the existence of concave points, the cell is separated into two segments, and the two segments are fitted by the ellipses as shown in Fig. 5(b). The contour really only contains one cell, so the two fitted ellipses should be combined, the combined ellipse generated through refitting the ellipse to all the points in the two segments is shown in Fig. 5(c). As the fitted ellipses in Fig. 5(b) and Fig. 5(c) show, some rules can be used to identify this case. Let α_E represent the angle between the major axis of an ellipse *E* and the X axis of the Cartesian coordinate system.


(a) Slender cell

(b) Fitted ellipses (c) Newly fitted ellipse

Fig. 5 Ellipse fitting for Case 2

(1) The α_E of the newly fitted ellipse (α_{new}) should be close to α_E of the two previous ellipses (α_1, α_2);

$$|\alpha_{new}-\alpha_1|<Th_{angle} \tag{15}$$

$$|\alpha_{new}-\alpha_2|<Th_{angle} \tag{16}$$

Th_{angle} is a pre-set threshold. Because the ellipse is slender, α_E of the newly fitted ellipse and the two previous ellipses are very close. So, Th_{angle} usually should be in the interval $[0, 0.2\pi]$. This rule indicates that, the direction of the major axis of newly fitted ellipse is similar to the cell.

(2) The α_E of the two previous ellipses (α_1, α_2) are much close to each other;

$$|\alpha_2-\alpha_1|<(1.0\sim1.3)Th_{angle} \tag{17}$$

This rule indicates that the two previous ellipses may wrongly separate one cell and the directions of the major axes of the two previous ellipses are similar to the cell.

(3) The minor axes of the two previous ellipses ($MinA_1$, $MinA_2$) are close to each other.

$$|MinA_1-MinA_2|<2.0dMinTh \tag{18}$$

This rule indicates that the sizes of two previous ellipses are similar to each other because they fitted the same cell.

(4) The centers of the two previous ellipses should be close to each other.

$$\|C_1-C_2\|<(2.0\sim3.0)dMinTh \tag{19}$$

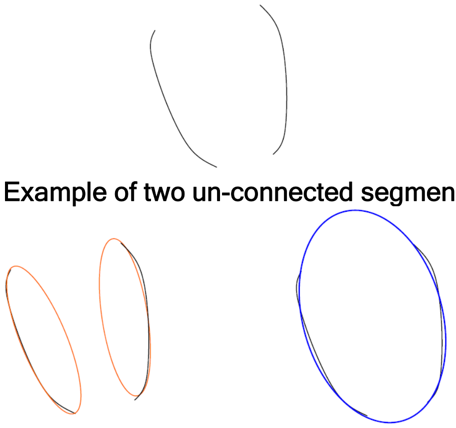
(5) The distance between the center (C_{new}) of the newly fitted ellipse and the centers of the two previous ellipses should be smaller than 1.5$dMinTh$.

Rules (4) and (5) indicate that the two previous ellipses are close to each other, they may be combined. The exact number between 2.0~3.0 in rule (4) can be selected following the prior knowledge.

The five rules of Case 2 indicate that the two previous ellipses have similar size and similar direction in the major axis which is close to the original cell. Also, the direction of the major axis of newly fitted ellipse is similar to the cell, closes to the two previous ellipses and fits the cell well. So, the two previous ellipses should be combined as one newly fitted ellipse.

Case 3: Two un-connected segments with similar fitted ellipses

In the case of complicated touching cells, the touching cells may have more than two cells touching each other (Fig. 3). So, the boundary of one cell may be separated into different unconnected segments. Each of the segments has one fitted ellipse, and these ellipses should be combined into one ellipse which corresponds to the original cell. This case is shown in Fig. 6(a). The two fitted ellipses and the newly generated ellipse through fitting the points of the two segments are shown in Fig. 6(b) and Fig. 6(c).


(a) Example of two un-connected segments

(b) Fitted ellipses from two separated segments

(c) Newly fitted ellipse

Fig. 6 Ellipse fitting for Case 3

Fig. 6(b) shows that the two previously fitted ellipses are similar to each other because they fit the same cell. Fig. 6(c) shows the newly fitted ellipse fits the two segments very well compared with the two separate fittings. The two ellipses should be combined. Two rules are used to identify this case.

(1) The minor axes of the two previous ellipses ($MinA_1$, $MinA_2$) should be smaller than $dMinTh$. The two segments are a small part of the contour of the same cell. So, the fitted ellipses should be small. This rule demonstrates this situation.

(2) The major ($MaxA_1$, $MaxA_2$) and minor axes of the two previous ellipses should be very close to each other.

$$|MaxA_1-MaxA_2|<dMinTh \tag{20}$$

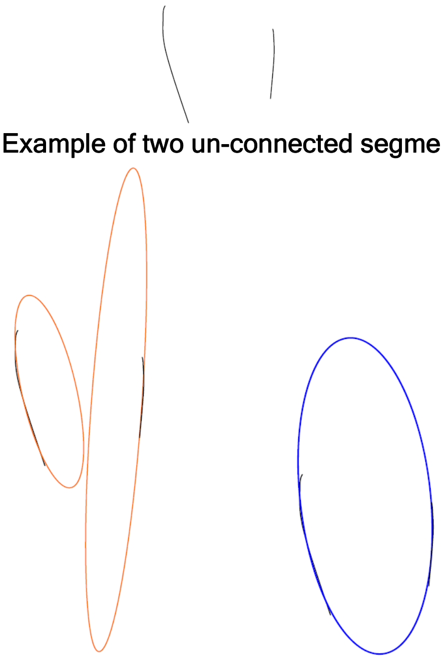
$$|MinA_1-MinA_2|<0.5dMinTh \tag{21}$$

This rule indicates that the two previous ellipses have similar shapes. It is because they belong to the same cell.

The two rules of Case 3 indicates that the two previous ellipses have small size and have similar shape, so the two previous ellipses should be combined as one newly fitted ellipse.

Case 4: Two un-connected segments with different fitted ellipses

Different from Case 3, two un-connected segments belonging to one same cell can also have two very different fitted ellipses because of the heavy touching of the cells and different segments of the contour. But, the two different ellipses should also be combined. This case is shown in Fig. 7(a).



(a) Example of two un-connected segments

(b) Fitted ellipses (c) Newly fitted ellipse

Fig. 7 Ellipse fitting for Case 4

The two fitted ellipse and the newly generated ellipse through fitting the points of the two segments are shown in Fig. 7(b) and Fig. 7(c).

Because the two fitted ellipses just fit part of the contour, the two previous ellipses may have a slender shape. Also, the two previous ellipses are very different in their major axes. And, the newly fitted ellipse (Fig. 7(c)) fits the segments well compared with the two previous fittings (Fig. 7(b)). All of these mean the two previous ellipses should be combined. Two rules are used to identify this case.

(1) The minor axes of the two previous ellipses ($MinA_1$, $MinA_2$) should be smaller than $dMinTh$. This rule indicates that the two previous ellipses have slender shape.

(2) The difference of the minor axes of the two previous ellipses should be small.

$$|MinA_1 - MinA_2| < 0.05dMinTh \qquad (22)$$

This rule indicates that the two previous ellipses have slender shape and also indicates that the two previous ellipses have similar shape at the direction of minor axis because they belong to the same ellipse.

The two rules indicate that the two previous ellipses have slender shape and similar shape. Also, the combined ellipse fits the segments well. So, they should be combined.

Case 5: Other cases

Except for the previous four cases, there may be some other special cases. In these cases, if the two segments belong to one cell, the newly fitted ellipse should fit the two segments well. So, they should satisfy the following rule.

(1) The mean algebraic distance (*dis*) of the newly fitted ellipse should be close to the *dis* of the two previously fitted ellipses. Usually, they are almost the same. Then, the newly fitted ellipse fits the two segments as well as the two previously fitted ellipses.

This rule indicates that, if the two segments satisfy this rule, the newly fitted ellipse fits the two segments as well as the two previously fitted ellipses and the two segments will not belong to two explicit touching cells. So, they should be combined. Otherwise, the two segments should be separated.

After the analysis of these cases, the contour of the possible touching cells can be processed following these cases to combine the ellipses belonging to the same cell. Moreover, the purpose of Case 2 is to identify a slender cell. Then, if the shape of the slender cell is too slender to be an ellipse, the difference between the newly fitted ellipse and the cell contour may be larger than other cases. So, *disTh* in Case 2 for newly fitted ellipse verification may be different from the original pre-set *disTh* and can be determined by the prior knowledge.

After the ellipse combination, the touching cells are separated and each separated cell is represented by an ellipse.

2.2.4 Ellipse refinement

Because some segments are not similar to a section of an ellipse or have small number of points which do not fit an ellipse well, the fitted ellipses of these segments are usually very small or very large or very slender. Then, after the ellipse selection, these segments are removed from the candidate ellipses. These segments were not processed in the ellipse combination step. But, obviously, these segments are part of the contour of the touching cells, and they should belong to one of the cell. So, these segments should also be processed.

In this section, each unprocessed segment is combined into one existing ellipse to generate a new ellipse, the fitting error of the new ellipse are calculated and compared. The cell whose fitting error of the corresponding new ellipse is smaller than *disThRe* is the cell that the unprocessed segment should belong to.

After ellipse refinement, all the segments of the contour of the touching cell are processed. *disThRe* is used to identify the appropriate ellipse following the prior knowledge of the cell. The segments which belong to the same cell are combined as one ellipse. So, the cells are separated as different cells represented by different ellipses.

3. Experimental Results

To demonstrate the efficient performance of the proposed algorithm, the algorithm is applied on 4 image sets. Each set has 10 to 30 images. The algorithm is implemented on a computer with Intel Core 2.13GHz CPU, 2GB RAM. Each image was firstly processed by the cell contour extraction; and the contour of the cells, including the single cell or touching cells, are extracted. Then, the contours are used by our algorithm to split the touching cells. Some examples are demonstrated below.

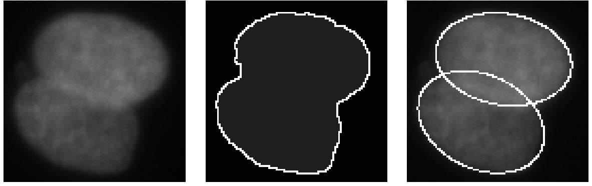


Fig. 8 Splitting result of simple touching cells

Fig. 8 is an example of two touching cells. The algorithm can accurately fit the contour of the touching cells and split the contour into separated cell contours.

Fig. 9 is the example of a slender cell. The algorithm can accurately fit the contour of the cell and treat the single slender cell as a single cell.

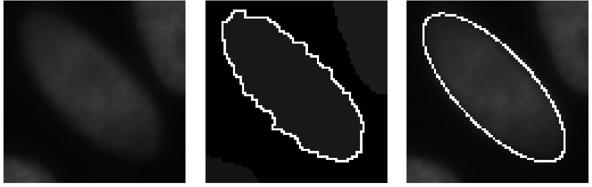


Fig. 9 Result of a slender cell

Fig. 10 is an example of multiple touching cells in which the three cells are slightly touching. The algorithm also fits the contour well as separated ellipses and splits them.

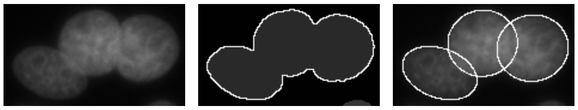


Fig. 10 Example of multiple touching cells

Fig. 11 is another example of multiple touching cells in which the cells touch each other. The algorithm also splits them well.

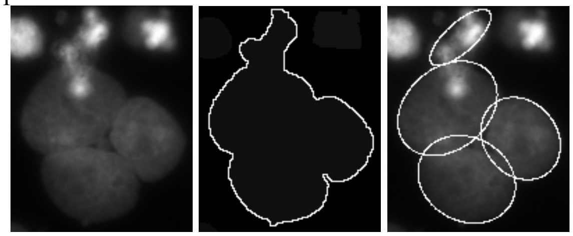


Fig. 11 Another example of multiple touching cells

Fig. 12 is an example of multiple touching cells caused by thresholding. Some of the cells do not really touch each in the original images, but because of the thresholding the extracted contour of the cells are touched. More importantly, the two touching cells at the middle are heavily touched. They can not be easily separated by simple marker based watershed segmentation, because it is difficult to find two separated marker for the two touching cells. But, our proposed algorithm can separate all the touching cells well, including the two heavily touching cells.

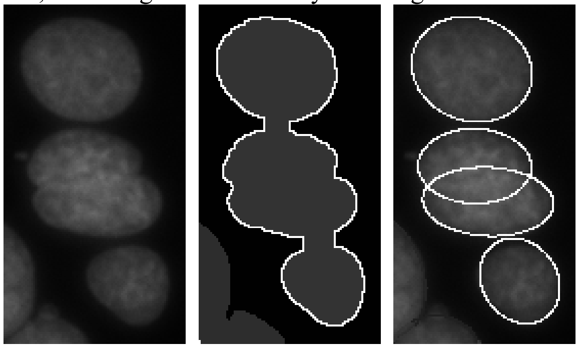


Fig. 12 Example of heavily touching cells and thresholding leading to touching cells

Fig. 13 is example of partial cell located at the edge of the image. The algorithm will add concave points at the contour of the cells. Then, this type of cell will also be processed by our algorithm. The results of the algorithm are demonstrated in Fig. 13 in which the cells are well fitted by the ellipses.

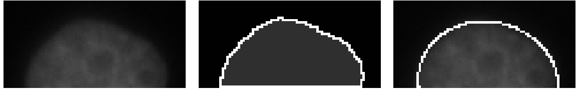


Fig. 13 Example of partial cells

These experimental results directly show the efficient performance of our proposed algorithm.

In addition, the splitting result of our algorithm can be used for different applications, such as texture measurement, obtaining accurate contour and so on.

For texture measurement, using the intensity of the touching cells not in the overlapping area of ellipses can give more accurate result, because the intensities of the touching area given by the overlapping ellipses are contaminated by the intensities of other touching cells. Also, the centres of the fitted ellipses can act as markers for further segmentation of the cell through different marker based methods [9, 10] to find the accurate contours of the cells. Therefore, our algorithm is efficient and useful.

4. Conclusions

A novel touching cells splitting algorithm by using concave points and ellipse fitting is proposed in this paper. Because of the contour pre-processing, the concave points are efficiently calculated and the contour is separated into different segments. Then, the different segments are processed as each single cell through the ellipse processing which is based on the properties of the fitted ellipses. Therefore, because of the efficient calculated concave points and the case study of different segments, the ellipse fitting can be used for touching cells splitting. Experimental results show that our algorithm can efficiently split different types of circle-like or ellipse-like touching cells.

Acknowledgments

We thank the reviewers for their valuable comments and suggestions. We are grateful to Yan Li at Peking University, Beijing, China, and Ryan Lagerstrom of CSIRO, Australia, for many helpful discussions and comments. We thank Dadong Wang at CSIRO for his help in the coding. This work is partly supported by China Scholarship Council and Innovation Foundation of Beijing University of Aeronautics and Astronautics for PhD Students.

References

[1] K.Z. Mao, P. Zhao and P. Tan, “Supervised learning-based cell image segmentation for P53 immunohistochemistry,” *IEEE Transactions on Biomedical Engineering*, Vol. 53, No. 6, 2006, pp. 1153-1163.

[2] C.D. Ruberto, A. Dempster, S. Khan and B. Jarra, “Analysis of infected blood cell images using morphological operators,” *Image and Vision Computing*, Vol. 20, 2002, pp. 133-146.

[3] W. Wang, “Binary image segmentation of aggregates based on polygonal approximation and classification of concavities,” *Pattern Recognition*, Vol. 31, No. 10, 1998, pp. 1503-1524.

[4] Y.X.C. Jin, S.H.O. Jayasooriah and R. Sinniah, “Clump splitting through concavity analysis,” *Pattern Recognition Letters*, Vol. 15, 1994, pp. 1013-1018.

[5] S. Kumar, S.H. Ong, S. Ranganath, T.C. Ong and F.T. Chew, “A rule-based approach for robust clump splitting,” *Pattern Recognition*, Vol. 39, 2006, pp. 1088-1098.

[6] N. Kharma, H. Moghnieh, J. Yao, Y.P. Guo, A. Abu-Baker, J. Laganiere, G. Rouleau and M. Cheriet, “Automatic segmentation of cells from microscopic imagery using ellipse detection,” *IET Image Processing*, Vol. 1, No. 1, 2007, pp. 39-47.

[7] C. Evans, M. Berman and H. Talbot, “Offline fast object splitting,” CSIRO Mathematical and Information Sciences, North Ryde, NSW, Australia, Tech. Rep. 04/55, 2004.

[8] G. Zhang, D.S. Jayas, and N.D.G. White, “Separation of touching grain kernels in an image by ellipse fitting algorithm,” *Biosystems Engineering*, Vol. 92, No. 2, 2005, pp. 135-142.

[9] B. Appleton, and H. Talbot, “Globally minimal surfaces by continuous maximal flows,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 28, No. 1, 2006, pp. 106-118.

[10] L. Vincent, and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 13, No. 6, 1991, pp. 583-598.

[11] P. Soille, *Morphological image analysis-principle and applications*, Springer, Germany, 1999.

[12] A. Fitzgibbon, M. Pilu, and R.B. Fisher, “Direct least square of ellipse,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 21, No. 5, 1999, pp. 476-480.