Automatic Image Analysis of Histopathology Specimens Using Concave Vertex Graph

Lin Yang[1,3], Oncel Tuzel[2], Peter Meer[1], and David J. Foran[3]

[1] Dept. of Electrical and Computer Eng., Rutgers Univ., Piscataway, NJ, 08854, USA
[2] Dept. of Computer Science, Rutgers Univ., Piscataway, NJ, 08854, USA
[3] Center of Biomedical Imaging and Informatics, The Cancer Institute of New Jersey, UMDNJ-Robert Wood Johnson Medical School, Piscataway, NJ, 08854, USA

Abstract. Automatic image analysis of histopathology specimens would help the early detection of blood cancer. The first step for automatic image analysis is segmentation. However, touching cells bring the difficulty for traditional segmentation algorithms. In this paper, we propose a novel algorithm which can reliably handle touching cells segmentation. Robust estimation and color active contour models are used to delineate the outer boundary. Concave points on the boundary and inner edges are automatically detected. A concave vertex graph is constructed from these points and edges. By minimizing a cost function based on morphological characteristics, we recursively calculate the optimal path in the graph to separate the touching cells. The algorithm is computationally efficient and has been tested on two large clinical dataset which contain 207 images and 3898 images respectively. Our algorithm provides better results than other studies reported in the recent literature.

1 Introduction

As new therapies emerge for blood cancer screening, it becomes increasingly important to distinguish among subclasses of lymphocytes in advance. Processing the specimen using a reliable, image-based analysis system could reduce the cost and patient morbidity. In image-based analysis the first step is segmentation. However, the traditional methods usually fail to accurately segment touching cells in the digitized hematologic specimens. Touching cells are especially prominent in malignant cases. In Figure 1, we show representative morphologies for benign and five hematologic malignancies (hematoxylin-eosin staining): Chronic Lymphocytic Leukemia (CLL) [1], Mantle Cell Lymphoma, (MCL) [2], Follicular Center Cell Lymphoma (FCC) [3], Acute Myelocytic Leukemia (AML) and Acute Lymphocytic Leukemia (ALL) [2].

The watershed algorithm is the most commonly used method for performing touching object segmentation. However, it suffers from several major drawbacks.

- *Oversegmentation.* The algorithm is sensitive to noise and often produces many oversegmented small regions. Marker-based watershed [4] can partially remedy this issue, but it requires manual selection or accurate estimation of the markers.

D. Metaxas et al. (Eds.): MICCAI 2008, Part I, LNCS 5241, pp. 833–841, 2008.

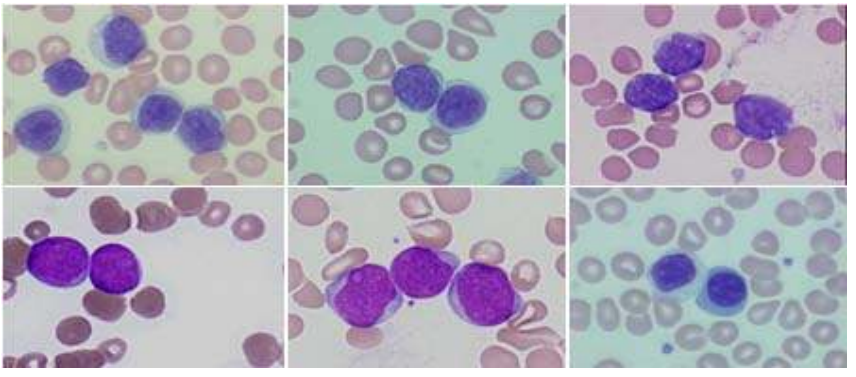


Fig. 1. Some representative morphologies of touching lymphocytes. In the first row, from left to right: CLL, MCL and FCC. In the second row, from left to right: ALL, AML and benign. The specimens were prepared at different hospitals and institutions therefore there exists large variations in staining.

– *Lack of shape prior.* It is generally difficult to include shape priors in the watershed transform. Although there are some efforts [5,6] proposed for specific cases, the general problem still exists.

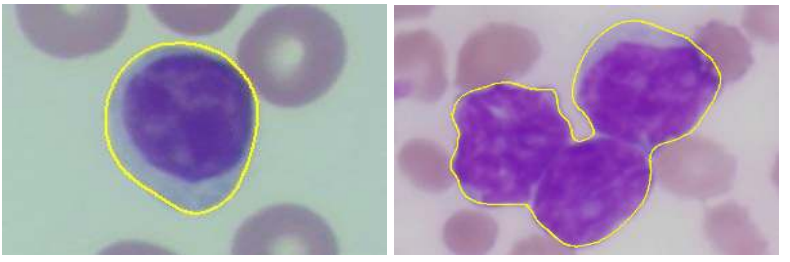
In this paper, we propose a novel algorithm to separate touching cells. The algorithm starts from a deformable model which extracts the boundary contour of the touching cells. The concave vertex graph is constructed using the concave vertices on the contour and the edges detected in the region of touching cells. The segmentation is then treated as an optimal grouping of pixels, which can be solved by recursively searching optimal shortest path in the concave vertex graph.

2 Boundary Contour Extraction

The initial step of the algorithm is to extract the boundary contour of the touching cells. We first apply a L_2E robust estimation [7] to provide a rough estimation of the outer boundaries of the cells inside the region of interest (ROI). A robust gradient vector flow (GVF) snake [8] using Luv [9, Sec. 8.4] color gradients is further applied to extract the objects from the background. Since the deformable models are initialized using the results of robust estimation, the convergence speed is increased and the method can handle topological changes. In this paper, we focus our attention on the touching cases shown in Figure 2b, where the output contour represents the outer boundary of the touching cells.

3 Concave Points and Inner Edges Detection

In Figure 3, we show the construction of the concave vertex graph. The contour found by boundary contour extraction algorithm is shown in Figure 3a. We detect



(a) **(b)**

Fig. 2. The segmentation result of robust color GVF snake. (a) The ROI contains only one cell. (b) The ROI contains the touching cells.

the high curvature points on the contour via [10](Figure 3b). At each point p on the contour a set of triangles are constructed. The points which satisfy

$$d_{\min} \leq |\boldsymbol{a}| \leq d_{\max} \qquad d_{\min} \leq |\boldsymbol{b}| \leq d_{\max} \qquad \alpha \leq \alpha_{\max} \tag{1}$$

where $\alpha = \arccos \frac{|\boldsymbol{a}|^2+|\boldsymbol{b}|^2-|\boldsymbol{c}|^2}{2|\boldsymbol{a}||\boldsymbol{b}|}$, $d_{\min}, d_{\max} = 7, 9$ pixels and $\alpha_{\max} = 150°$ are kept. The candidates are further processed to suppress the local nonmaxima points. The final high curvature points correspond to both concave and convex points. We keep only the concave points, shown as red rectangles in Figure 3c. This can be calculated from the sign of the cross product $\boldsymbol{a} \otimes \boldsymbol{b}$, which has to be negative for concave points.

Canny edge detector is applied inside the cell region and straight line fitting is used to model the edges (Figure 3d). The separating curve combines a pair of convex vertices on the boundary and is enforced to pass through the inner edges.

4 Touching Cells Segmentation

The outer boundary of the touching cells is defined as C, and the region enclosed by C is $R(C)$. The concave points are the set V, e.g. $v1 - v5$ which are shown in Figure 3e. The inner edges are the set E, e.g. shown as white solid lines in Figure 3e and also illustrated by e_i in Figure 3f.

4.1 Concave Vertex Graph

In Figure 3f we construct the concave vertex graph G. Let W be the vertex set consisting of the end points of inner edges E, e.g. w_i and w_j in Figure 3f. The vertices of graph G are then equal to $V \cup W$.

The graph has two sets of edges E and F. The set E contains the inner edges found by the edge detection algorithm. The set F is constructed with *filling edges* by connecting the vertices in G which are not connected by inner edges, e.g. f_k in Figure 3f. The lengths of the inner edges are set to ϵ (10^{-16}), while the lengths of the *filling edges* in set F are given by the Euclidean distance between the two vertices of the edges.

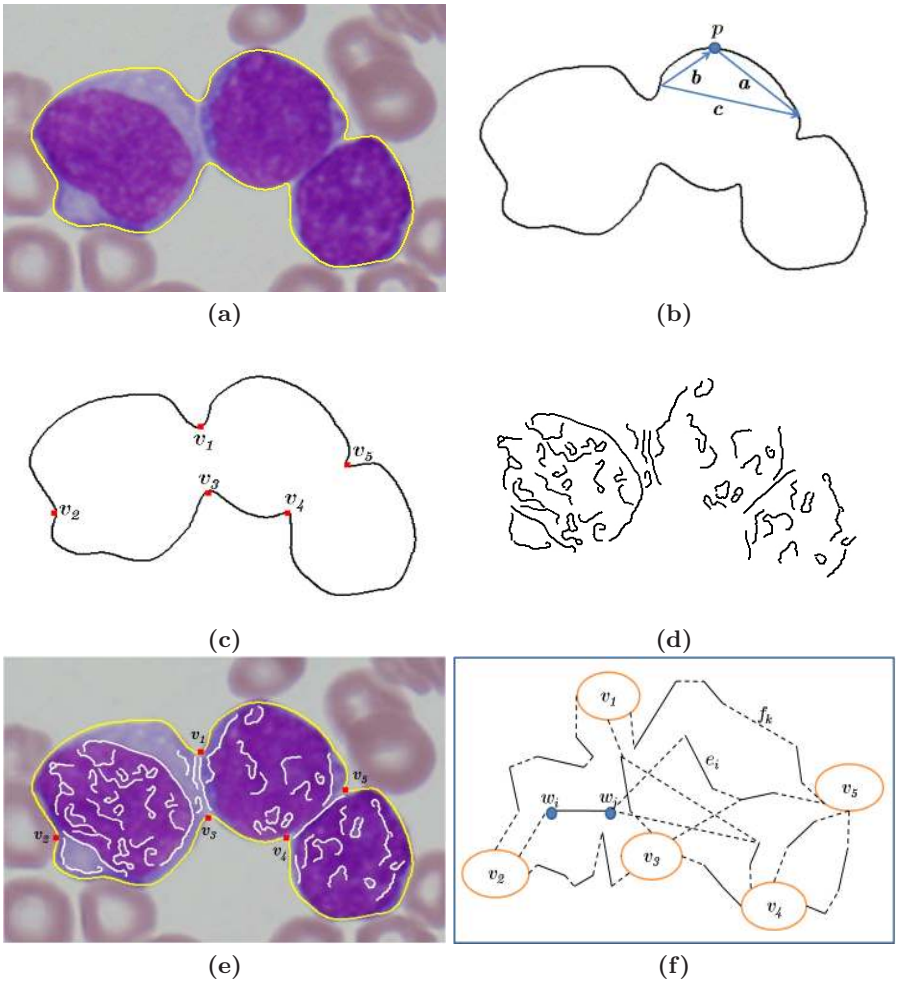


Fig. 3. Construction of the concave vertex graph. (a) The original image with the yellow boundary contour. (b) High curvature points detection. (c) Concave points detection. (d) Inner edges detection. (e) The outer boundary C, concave vertices V and inner edges E, superimposed on the original image. (f) The constructed concave vertex graph G. The filling edges are shown with dotted lines.

The Dijkstra algorithm is used to find the shortest path p_{ij} between v_i and v_j. The length of the p_{ij}, $\|p_{ij}\|$, is given by the total length of the *filling edges* f_k in p_{ij} because the length of real inner edges is set to be ϵ

$$\|p_{ij}\| = \sum_{f_k \in p_{ij}} length\,(f_k)\,. \tag{2}$$

In Figure 3f, as an example, we can see $\|p_{12}\| > \|p_{13}\|$ because p_{12} traverse longer *filling edges* than p_{13}. The defined path lengths enforce the segmentation

Input: Given the region of interest (ROI) containing touching cells.
- Extract the boundary contour C, detect the concave points V, the inner edges E in $R(C)$, construct the concave vertex graph G.
- for each vertex $v(i) \in V$
  - Find the path p_{ij} and calculate the length $\|p_{ij}\|$ using (2).
- Initialize $mincost = +\infty$ and $Q = \varnothing$.
- while (V is not empty)
  - for each vertex $v(i) \in V$
    - for each vertex $v(j \neq i) \in V$
      - Apply the path p_{ij} to separate the graph G in to L and R.
      - Calculate the cost c using (6) and save in Q.
  - Sort Q and pick up the path p_{ij} with the lowest cost c.
  - if ($c < 1.5 * mincost$)
    - Record path p_{ij} and the region $R(C, p_{ij})$ with cost c in the $result$.
    - The edges and zero degree vertices in the $R(C, p_{ij})$ are removed from G.
    - Set $mincost = c$ and $Q = \varnothing$.
  - else return $result$.

Alg. 1. The algorithm to separate touching cells using concave vertex graph

to follow inner edges since the trivial solution to directly connect two concave vertices using only *filling edges* in graph G would provide a longer path.

After the Dijkstra algorithm is applied, we find all the shortest pathes among concave vertices, p_{ij}, which are valid candidates to separate touching cells. The key idea of our algorithm is to treat the touching cells segmentation as recursively searching for the best path p_{ij} in G, which minimizes a cost function specifically designed to prefer cell-like object-cut.

4.2 Cost Function

We are looking for perceptually "good" segmentation of touching cells. For this purpose, we design the cost function to represent the clues that surgical pathologists use for judgement.

- The cells should be objects which are perceptually salient, since humans intend to separate such objects in an image. A good definition of saliency is proposed in [11] based on the Gestalt laws [12]. We apply the minimum of two saliency costs

$$c_s = \min\left(\frac{\|p_{ij}\|}{\sqrt{area_L\,(C, p_{ij})}}, \frac{\|p_{ij}\|}{\sqrt{area_R\,(C, p_{ij})}}\right) \tag{3}$$

  where $\|p_{ij}\|$ is the length defined in (2), each path p_{ij} in G divides $R(C)$ into two regions L and R, and the min function in (3) selects the region with the smallest cost. The $area(C, p_{ij})$ denotes the area enclosed by C and path p_{ij}.
- The cells are objects which are close to elliptical shape and can be modeled by ellipse fitting using points on C and p_{ij}. The ratio between the long and

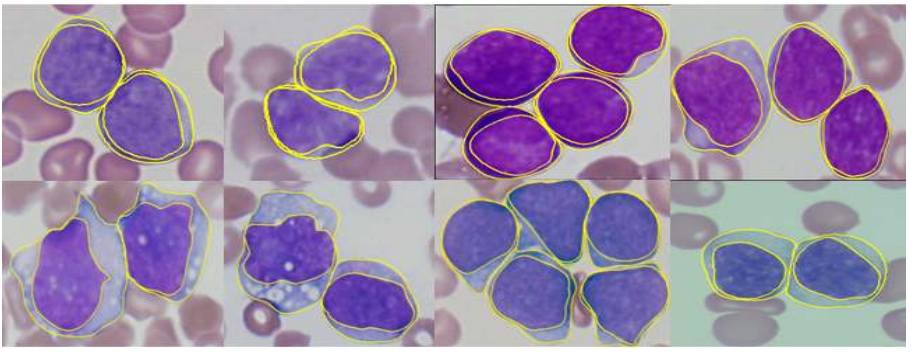


Fig. 4. The segmentation results using the concave vertex graph

short axes is recorded as tg. The segmented objects are expected to provide a ratio tg in the range $[tg_1, tg_2]$, in which case the $dist\,(tg, [tg_1, tg_2]) = 0$. Otherwise, we define $dist\,(tg, [tg_1, tg_2]) = \min\,(|tg - tg2|\,, |tg - tg1|)$.

$$c_g = \min\left(\frac{1}{1+\exp\left(-dist\,(tg_L, [tg_1, tg_2])\right)}, \frac{1}{1+\exp\left(-dist\,(tg_R, [tg_1, tg_2])\right)}\right) \tag{4}$$

where the L and R have the same definition as (3). The tg_1 and tg_2 represent the lower bound and upper bound of the long axes to short axes ratio.

– The cells are objects which have biologically reasonable areas. Following the definition above, we use ta_1 and ta_2 to represent the lower bound and upper bound of the cell area.

$$c_a = \min\left(\frac{1}{1+\exp\left(-dist\,(ta_L, [ta_1, ta_2])\right)}, \frac{1}{1+\exp\left(-dist\,(ta_R, [ta_1, ta_2])\right)}\right). \tag{5}$$

– The final cost c is the weighted sum

$$c = \lambda_1 c_s + \lambda_2 c_g + \lambda_3 c_a \qquad \sum_{i=1}^{3} \lambda_i = 1. \tag{6}$$

The optimal values of coefficients are selected as $\lambda_1 = 0.5, \lambda_2 = 0.3$ and $\lambda_3 = 0.2$, which are learned in an offline process using a training set and held constant throughout the experiments.

4.3 Algorithm

Using the concave vertex graph G and the cost function c, the method is described in Algorithm 1. It is recursively applied to separate touching cells until all the region $R(C)$ are allocated to the segmented cells. The algorithm only separates the cytoplasm of the touching cells. Since the colors of nuclei and cytoplasm are distinct, they can be easily separated. In order to provide smooth boundaries, we apply the quadratic splines to postprocess the boundaries of each segmented cell.

Table 1. Segmentation accuracy(%) using the concave vertex graph. The $accuracy_c$ and $accuracy_n$ represent the segmentation accuracy for cytoplasm and nuclei respectively.

	Benign	CLL	MCL	FCC	AML	ALL
$accuracy_c$ (%) of touching cells	90.1	90.8	86.4	86.9	86.3	85.2
$accuracy_n$ (%) of touching cells	92.3	91.2	88.1	88.7	87.5	87.9
$accuracy_c$ (%) of all cells	92.5	91.7	87.2	89.1	88.5	87.6
$accuracy_n$ (%) of all cells	95.8	92.8	90.1	91.0	88.9	89.2

Table 2. The segmentation accuracy(%) using the watershed algorithm and the concave vertex graph

	Mean	Variance	Median	Min	Max	80%
Watershed	74.3	9.8	75.1	65.4	82.7	72.9
Concave Vertex Graph	88.9	5.1	90.2	75.2	95.5	87.1

5 Experiments

The cell database consists of a mixed set of 86 hematopathology cases: 18 Mantle Cell Lymphoma (MCL), 20 Chronic Lymphocytic Leukemia (CLL), 9 Follicular Center Cell Lymphoma (FCC), 18 Acute Lymphocytic Leukemia (ALL), 19 Acute Myelocytic Leukemia (AML), and 19 benign cases. For each case, there are varying number of cell images from 10 to 90. In total there exists 3898 cell images in our complete database. All the cases were generated from the archives of City of Hope Hospital in California, University of Pennsylvania of School of Medicine, Spectrum Health System, Grand Rapids, MI and Robert Wood Johnson Medical School, University of Medicine & Density of New Jersey.

The imaging platform for the experiments consisted of an Intel-based workstation interfaced with a high-resolution Olympus DP70 camera equipped with 12-bit color depth on each color channel and 1.45 million pixel effective resolution. The system also includes a single 2/3 inch CCD digital camera, an Olympus AX70 microscope equipped with a Prior 6-way robotic stage, motorized objective turret and a magnification changer.

We compare the segmentation results with manually segmentation. Two sets of experiments are performed.

- *The 207 touching cases of the histopathology cell image dataset.*
- *The complete database which contains 3898 histopathology cell images.*

Figure 4 shows some segmentation results. In Table 1 we present the segmentation accuracies for the six different classes of lymphocytes in two set of experiments. We obtained an average accuracy 88.9% on the touching cells dataset and 90.1% on the complete database.

Only a limited number of recent literature addresses the issue of touching cells segmentation in histopathology images using hematoxylin staining in high resolution (60× in our case). The watershed algorithm [4] is widely accepted for

touching object segmentation and successfully used in segmenting histopathology images [13]. We compared our method with watershed using the 207 touching cell image dataset and listed the results in Table 2. The 80% column in Table 2 represents the sorted 80% highest accuracy of all the results, and is commonly used by doctors to evaluate the usability of the system. The experiments demonstrate the superior performance of the presented approach.

6 Conclusion

In this paper, a novel segmentation algorithm has been proposed to address the challenges of touching cell segmentation in hematologic specimens. The results are validated using real clinical data containing six classes of hematologic blood cell images. We compare our algorithm with watershed and experimentally show the superior performance of the proposed algorithm.

For general pixel grouping problem using a normal graph, the optimization problem is NP-hard. Only certain cost function can be *approximately* solved using algorithm like normalized cut [14] in polynomial time. In our algorithm, the cost function is designed to meet the domain specific requirements. The concave vertex graph, which utilize the concave points of the outer contour, reduce the search space to the shortest pathes in the constructed graph G. Based on a MATLAB implementation, the algorithm can finish in less than 2 seconds for an 128×128 image.

References

1. Rozman, C., Montserrat, E.: Chronic lymphocytic leukemia. The New England Journal of Medicine 333(16), 1052–1057 (1995)
2. Cotran, R., Kumar, V., Collins, T., Robbins, S.: Pathologic basis of disease, 5th edn. W.B. Saunders Company, Philadelphia (1994)
3. Aisenberg, A.: Coherent view of non-Hodgkin's lymphoma. J. Clin. Oncol. 13, 2656–2675 (1995)
4. Moga, A.N., Gabbouj, M.: Parallel marker-based image segmentation with watershed transformation. Journal of Parallel and Distributed Computing 51(1), 27–45 (1998)
5. Grau, V., Mewes, A.U.J., Alcaniz, M., Kikinis, R., Warfield, S.K.: Improved watershed transform for medical image segmentation using prior information. ITMI 23(4), 447–458 (2004)
6. Nguyen, H.T., Ji, Q.: Improved watershed segmentation using water diffusion and local shape priors. CVPR 1, 985–992 (2006)
7. Scott, D.W.: Parametric statistical modeling by minimum integrated square error. Technometrics 43, 274–285 (2001)
8. Yang, L., Meer, P., Foran, D.: Unsupervised segmentation based on robust estimation and color active contour models. IEEE Trans. on Information Technology in Biomedicine 9, 475–486 (2005)
9. Wyszecki, G., Stiles, W.S.: Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd edn. Wiley, Chichester (1982)

10. Chetverikov, D., Szabó, Z.: A simple and efficient algorithm for detection of high curvature points in planar curves. In: The 23rd Workshop of the Austrian Pattern Recognition Group, pp. 175–184 (1999)
11. Stahl, J.S., Wang, S.: Convex grouping combining boundary and region information. ICCV 2, 946–953 (2005)
12. Elder, J.H., Goldberg, R.M.: Ecological statistics of Gestalt laws for the perceptual organization of contours. Journal of Vision 2(4), 324–353 (2002)
13. Adiga, P.S.U., Chaudhuri, B.B.: An efficient method based on watershed and rule-based merging for segmentation of 3D histo-pathological images. J. Pattern Recognition 34(7), 1449–1458 (2001)
14. Cai, W., Chung, A.C.: Multi-resolution vessel segmentation using normalized cuts in retinal images. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4191, pp. 928–936. Springer, Heidelberg (2006)