CLUMP SPLITTING VIA BOTTLENECK DETECTION

Hui Wang, Hong Zhang and Nilanjan Ray

Department of Computing Science
University of Alberta, Edmonton, Alberta, CANADA
{wanghui,zhang,nray1}@cs.ualberta.ca

ABSTRACT

Under-segmentation of an image with multiple objects is a common problem in image segmentation algorithms. This paper presents a novel approach for the splitting of clumps formed by multiple objects due to under-segmentation. The algorithm includes two steps: finding a pair of points for clump splitting, and joining the pair of selected points. In the first step, a pair of points for splitting is detected using a bottleneck rule, under the assumption that the desired objects have roughly convex shape. In the second step, the selected pair of splitting points is joined by finding the optimal splitting line between them, based on minimizing an image energy. The performance of this method is evaluated using images from various applications. Experimental results show that the proposed approach has several advantages over existing splitting methods in identifying points for splitting as well as finding an accurate split line.

Index Terms— Clump Splitting, Bottleneck, Under-segmentation

1. INTRODUCTION

Clump splitting in image segmentation is to divide a segmented region into two or more parts or portions. Due to a common phenomenon that objects of interest tend to clump together in a wide variety of image data, the splitting of a clumped object into constituent objects must therefore be performed to ensure the overall success of a vision task. Although a human operator may be able to detect the constituent objects of interest based on prior knowledge and perception of texture and structure, it is difficult for a computer-based algorithm to do this automatically.

There are two major steps in clump splitting. The first step is to identify candidate points for splitting. The second step is to join the selected points together to split the clump. Most existing clump splitting methods focus only on the first step. Available methods for the first step can be categorized into binary erosion methods and mathematical morphological based methods [1], watershed-based techniques [2], model-based approaches [3, 4, 5] and concavity analysis [6, 7].

A difficulty with morphological based methods is that they may completely erode a constituent object in a clump before a split occurs. While watershed based methods usually tend to over-segment the object of interest, model-based techniques in general are computationally expensive and requires proper initialization of the model parameters. For example, Liu et al. [5] propose a method for deformable shape-based image segmentation by merging/splitting regions in an image based on their agreement with a prior distribution on the global deformation parameters for a particular shape class. This method faces difficult minimization in high dimensional parameter space.

Concavity analysis methods, on the other hand, offer an intuitive way of clump splitting. Such methods have been successfully implemented in a variety of application domains. Bai et al. [8] presents a touching cells splitting algorithm based on concave points and ellipse fitting. It extracts concave points by using concave property of a region with some rules for special cases. Then it applies ellipse fitting after the contour has been segmented by concave points. However, experiments have shown that these concavity analysis methods are only applicable to objects of specific sizes and shapes. They are very sensitive to noise and are not general for different shapes with different sizes. For instance, the contour of the clump usually has many false concave regions, which will affect the performance of these shape based algorithms. Furthermore, to find valid concavity points, the methods need proper initial settings for parameters. They are thus not sufficiently general for clump splitting in a lot of applications. Other recent splitting related methods can be found such as rule-based methods [9], which propose a method based on a concavity analysis which is adaptable to many shapes and sizes and depends on a set of parameters that are obtained from a large set of training samples. However, many samples in [9] are synthetic and there is not a study of the degree of overlapping which the method is capable of dealing with. It will also in some cases be computationally expensive. In addition, concavity points are not always the best candidate points for splitting. We refer to Figure 2(a) and Figure 2(b) to show this point. Bai et al. [8] have proposed an extra step by adding points which are not concave points for splitting, but this extra step is not always accurate and efficient.

On the other hand, after obtaining selected points for splitting, for the second step of clump splitting, most existing methods generally ignore the available edge information, which can help with obtaining accurate boundary. Instead, some splitting algorithms blindly join two selected points with the shortest Euclidean distance. Some other splitting methods apply the short cut rule [10], that is, if boundary points can be joined in more than one way, the short cut rule prefers the path which uses the shortest cut. These methods can cause an inaccurate splitting segmentation. For an application when segmentation accuracy is very important, blindly finding the shortest Euclidean cut between selected points may not be satisfactory.

In this paper, we propose a novel clump splitting method which consists of two major parts: (i) identifying points for splitting, and (ii) joining the selected points for splitting. For the first part, we focus on clumps which consist of two constituent objects, with their splitting points occurring at bottleneck positions of the clump. For this step the only assumption made is that the objects to be split into are expected to have roughly convex shape. Based on our survey on existing applications that require splitting, to the best of our knowledge, this is the most common case and most existing clump splitting methods deal with this case. For the second part, we find an optimal split curve between the selected points from the first step, based on minimizing an image energy which corresponds to finding a connecting curve that visually best split the constituent objects of interest.

2. PROPOSED METHOD

2.1. Proposed procedure

The proposed clump splitting procedure works as follows:

Step 1: Apply bottleneck rule described in Section 2.2 to find a pair of points for splitting.

Step 2: Take a small local image patch from the original image, around the two selected points found in Step 1, and perform the method described in Section 2.3 to find the right split line between the two selected points.

Note that Step 1 occurs on the binary segmented image and Step 2 occurs on the original image. We will address each step in details in the following sections.

2.2. Identify points for splitting via bottleneck rule

The first step in our proposed procedure is to identify a pair of points for splitting. To address the drawbacks from existing methods and identify points for splitting more robustly and efficiently, we propose to apply a bottleneck rule.

Based on observations, a pair of points for splitting usually occur at a bottleneck position of a connected object, under the assumption that the desired objects have roughly convex shape. Therefore, we assume that the splitting points occur at bottleneck positions of the binary clump. Visually, the term bottleneck refers to the narrowest position on the shape of a bottle. Thus, bottleneck is the most likely place for congestion to occur, slowing down the flow of liquid from the bottle. In clump splitting, the bottleneck refers to the part where two objects are connected, that is, the bottleneck part can be defined by a pair of points.

To find the position where a bottleneck occurs on an object, let A and B represent any two different points on the contour of the object. We first define a cost function between points A and B for splitting as:

$$E_s(A,B) = \frac{dist(A,B)}{\min\{length(A,B), length(B,A)\}} \quad (1)$$

where $dist(A,B)$ represents the Euclidean distance between points A and B, $length(A,B)$ denotes the clockwise geodesic length from point A to B on the boundary of the clump, $\min\{length(A,B), length(B,A)\}$ represents the smaller value between these two. Here we assume that for any clump, the boundary should be one closed curve, and the line joining points A and B should be inside the clump.

Then, we define A^* and B^* to be a pair of points defining the position of the bottleneck on the contour when the cost E_s is minimized:

$$(A^*, B^*) = \arg\min_{A,B} \frac{dist(A,B)}{\min\{length(A,B), length(B,A)\}} \quad (2)$$

Figure 1 shows an example of points A^* and B^* found via bottleneck rule.

Fig. 1. Examples of points found via bottleneck rule. The red crosses indicate points A^* and B^*.

Note that even though we are only dealing with the case of finding a pair of points at one bottleneck position in this paper, our method can be extended easily to finding multiple pairs of points for splitting. For objects which need to be split more than once, the proposed bottleneck rule should still work as long as each pair of the splitting points occurs at bottleneck positions of the clump. We will leave the determination of the number of constituent objects in a clump for our future work.

2.3. Cut from selected points via weighted shortest path

As mentioned in the introduction, most existing splitting methods ignore any available edge information. However, available edge information can sometimes be helpful in obtaining a more accurate object boundary. We refer to Figure 3(a) and 3(b) to demonstrate this in our experiments section.

Therefore, we propose to cut from the pair of points found in the previous step by the weighted shortest path, to be described next. We will show that the proposed method not only makes use of helpful edge information, but also can work in a similar way to existing methods when edge information is not evident.

To begin with, we propose to form a small local image patch $\mathbf{I}$ from the original image, around the two selected points A^* and B^*. The size of the image patch $\mathbf{I}$ is defined as proportional to the object size. Intuitively, our goal is to find the strongest edge that connects the two splitting points in the local image patch $\mathbf{I}$. This leads to the following energy function:

$$e(\mathbf{I}) = \left(\left| \frac{\partial}{\partial \mathbf{x}} \mathbf{I} \right| + \left| \frac{\partial}{\partial \mathbf{y}} \mathbf{I} \right| \right) \tag{3}$$

Given such an energy function, finding a path between point A^* and B^* which maximizes the energy function corresponds to finding a connecting curve that visually best splits the constituent objects of interest. Formally, we define a cut $\mathbf{c}$ to be a path connecting points A^* and B^* on $\mathbf{I}$. The pixels on $\mathbf{c}$ will therefore be $\mathbf{I}_\mathbf{c} = \{\mathbf{I}(\mathbf{c}_i)\}_{i=1}^{L}$ where L is the length of the cut $\mathbf{c}$, and $\mathbf{c}_i$ represents the position of a pixel at location i on the cut. We then define the cost of a splitting cut $\mathbf{c}$ between points A^* and B^* to be $E(\mathbf{c}) = \sum_{i=1}^{L} 1/e(\mathbf{I}(\mathbf{c}_i))$. Therefore, an optimal cut c* should minimize this cut cost:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} E(\mathbf{c}) = \arg\min_{\mathbf{c}} \sum_{i=1}^{L} \frac{1}{e(\mathbf{I}(\mathbf{c}_i))} \tag{4}$$

Interestingly, minimizing $E(\mathbf{c})$ is equivalent to finding the weighted shortest path between points A^* and B^* on $\mathbf{I}$. In our implementation, we apply the efficient Dijkstra algorithm [11] to compute it.

Note that energy function (3) could also be written in different ways, as long as it captures the edge information from the original image. Also, when edge information is not adequate, i.e. assuming no edge information is present for determining a good cut, the geodesic distance between points A^* and B^* found by (4) will be the shortest Euclidean distance between them, which is similar to most existing methods.

3. EXPERIMENTAL RESULTS

This section presents the performance of our method and compare the performance with other clump splitting methods. We show our experimental results visually in two parts: (i) results in identifying points for splitting (Figure 2); (ii) results in joining the selected pair of points (Figure 3).

A total of 77 splitting cases have been tested from oil sand images and 38 splitting cases from blood cell images. The overall performance is evaluated quantitatively using the metric of the probability of correct detection (PCD), which is defined as the percentage of correctly split objects [8] and the results are presented in Table 1. We can see that our method obtains a much higher PCD than the concavity-based method [8]. This is expected since our method does not find all concavity points, but rather a pair of points which occur at the bottleneck position. Therefore our method has great advantage in splitting cases in which one or both of the points are not necessarily concave points, or are not concave points with highest concavity value [8]. In general, both our method and concavity-based method [8] obtain better results with the blood cell images than oil sand images. This is due to the smoother object boundary in blood cell images segmentation than in oil sand images, which are very noisy.

From Figure 2, we can see that our bottleneck-based method works for situations when points for splitting include non-concave points, or when concave points with very high concavity value is not the optimal candidate for splitting. Figure 3 shows some final visual results for applying our splitting method on oil sand images and blood cell images. Figure 3(a) and 3(b) clearly show that available edge information is helpful for accurate clump splitting, comparing to only connecting two points with a shortest Euclidean distance. Bai et al. [8] apply ellipse fitting for the second part of the splitting, which only applies to blood cells and thus is not comparable to our results.

Table 1. Detailed performance

	Probability of correct detection	
Image set	Our method	Concavity-based method [8]
Oil sand images	75%	43%
Blood cell images	92%	61%

4. CONCLUSIONS AND FUTURE WORK

We have proposed a novel algorithm to perform clump splitting in this paper. The proposed method consists of two steps. Both steps aim at overcoming drawbacks from existing methods. In the first step it finds pairs of splitting points based on the simple observation that splitting cuts usually occur at the bottleneck part of a connected clump. Thus, a pair of bottleneck points can be found when a proposed cost function between two points is minimized. In the second step, the proposed algorithm finds accurate cut by minimizing a proposed energy function based on image level information. This step helps the splitting steps to obtain more accurate splitting results than existing methods

Our method has several advantages over existing methods. First, our method does not need parameter tunings on finding the dominant concavity points compared to most existing methods. Second, our method can deal with special situations when normal concavity/distance based methods need to add points for splitting. Finally, our method takes available weak edge information into account and finds a more accurate cut than shortest Euclidean cut. The proposed method has been shown to be robust by accurately splitting clumps from various applications. Future work includes dealing with clumps

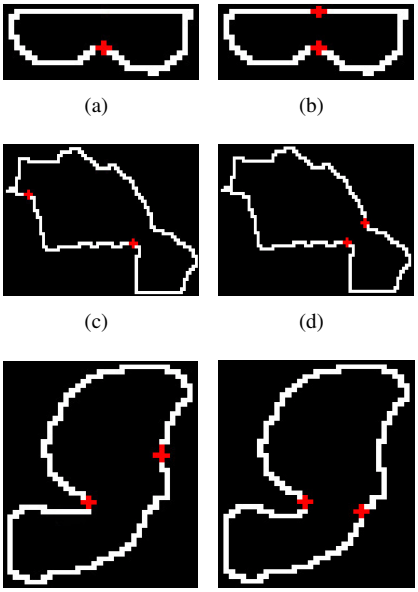

Fig. 2. Results for a pair of selected points for splitting via bottleneck rule. The first row shows blood cell clumps and the second and third row show oil sand clumps. The left column shows example results using the algorithm in [8]. The right column shows our result. The red crosses indicate the selected points.

which need to be split more than once.

5. REFERENCES

[1] Rafael C. Gonzalez and Richard E. Woods, *Digital image processing*, Prentice Hall, 2 edition, January 2002.

[2] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in *Proceedings of the International Workscla hop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, 1979, pp. 17–21.

[3] H.H.S. Ip and R.P.K. Yu, "Recursive splitting of active contours in multiple clump segmentation," *Electronics Letters*, vol. 32, no. 17, pp. 1564–1566, August 1996.

[4] Shoichi Araki, Naokazu Yokoya, Hidehiko Iwasa, and Haruo Takemura, "A new splitting active contour model based on crossing detection," in *Proceedings of the Second Asian Conference on Computer Vision*, 1995, vol. II, pp. 346–350.

[5] Lifeng Liu and Stan Sclaroff, "Shape-guided split and merge of image regions," in *IWVF-4: Proceedings of the 4th International Workshop on Visual Form*. 2001, pp. 367–377, Springer-Verlag.

[6] T. T. E. Yeo, X. C. Jin, Siew Hui Hui Ong, and R Sinniah Jayasooriah, "Clump splitting through concavity analysis," *Pattern Recognition Letters*, vol. 15, no. 10, pp. 1013–1018, October 1994.

[7] W. X. Wang, "Binary image segmentation of aggregates based on polygonal approximation and classification of concavities," *Pattern Recognition*, vol. 31, no. 10, pp. 1503–1524, 1998.

[8] Xiangzhi Bai, Changming Sun, and Fugen Zhou, "Splitting touching cells based on concave points and ellipse fitting," *Pattern Recognition*, vol. 42, no. 11, pp. 2434–2446, 2009.

[9] Saravana Kumar, Sim Heng Ong, Surendra Ranganath, T C Ong, and Fook Tim Chew, "A rule-based approach for robust clump splitting," *Pattern Recognition*, vol. 39, no. 6, pp. 1088–1098, June 2006.

[10] Manish Singh, Gregory D. Seyranian, and Donald D. Hoffman, "Parsing silhouettes: The short-cut rule," *Percept Psychophys*, vol. 61, pp. 636–660, 1999.

[11] Edsger W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

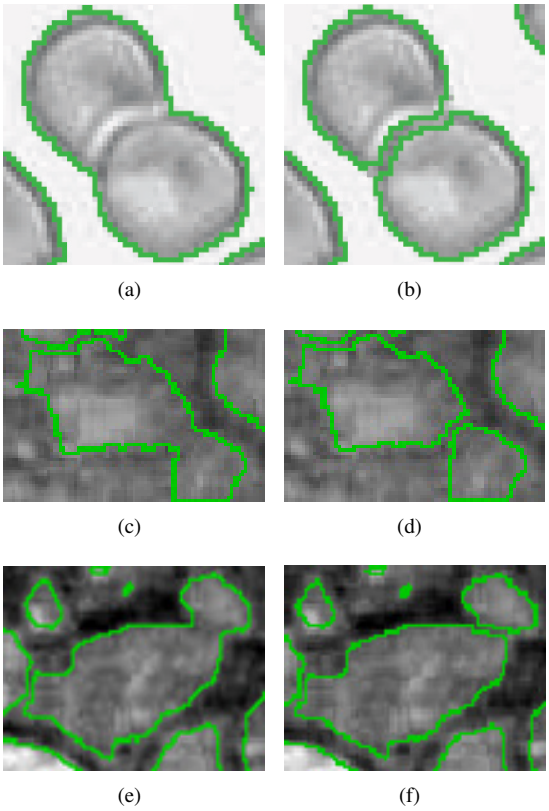

Fig. 3. Visual results after splitting for blood cell images (a) and (b) and oil sand images (c) through (f). The left column shows result before splitting. The right column shows result after splitting.