Hierarchical Merging of Adjacent Subtrees from Delaunay Triangulation with Centers of Superpixels

Eu-Tteum Baek, Yo-Sung Ho
School of Information and Communications,
Gwangju Institute of Science and Technology (GIST)
123 Cheomdangwagi-ro, Buk-gu, Gwangju, Republic of Korea
Email: {eutteum, hoyo}@gist.ac.kr

***Abstract*— Image segmentation is used in computer vision, medical imaging, and biological imaging to locate object boundaries and to group similar pixels together to form a set of coherent image regions. The important factors of clustering are similarity, proximity, and good continuation, which lead to visually meaningful segmentation. On the contrary, there are some problems of visual grouping such as over-segmentation, inaccuracy, and time-consuming tasks. Among the problems, we concentrate on reducing manual settings and avoiding the over-segmentation. In the paper, we propose a segmentation method which is merging hierarchically partial trees of superpixels. Given the superpixels, we determine the each center of superpixels to each node, and construct a Delaunay triangulation to compute which regions are adjacent. Similar regions are joined by using a similarity measure. An important chacteristic of the algorithm is its ability to reduce the over-segmentation and to preserve detail.**

Keywords—: image segmentation, superpixel, Delaunay triangulation

I. Introduction

In the computer vision or pattern recognition fields, image segmentation is treated as an important issue, and has a long history, starting with Gestalt psychology which identified several factors that lead to human perceptual grouping: similarity, proximity, continuity, symmetry, parallelism, closure and familiarity [1]. The key factors has been concerned as cues for many clustering algorithms. In recent years, the research about image segmentation is highly focused by people, and a lot of algorithms have been presented. This is often the first step in image processing, with the end results heavily relying in the segmentation quality.

Image segmentation is the process of grouping an image into multiple sets of pixels. The objective of image segmentation is to decompose an image into meaningful partitions, to simplify, or change the representation of an image into something that is easier to analyze [2]. Each of the pixels in a grouping are similar in terms of some characteristic, as intensity, color, pattern, or texture. Image segmentation can be grouped into two major types: (1) contour based approaches, and (2) region based approaches. Region based approaches manage to extract parts of the image pixels into group corresponding to coherent image characteristic such as brightness, color and texture. Contour based approaches usually start to detect edges, followed by a linking process that to apply curvilinear continuity. From a practical viewpoint, Image segmentation can also be classified into two broad families according to local or global searches. Most of the early techniques were based on local search techniques. Local based approaches declare an edge a pixel with high gradient, or make a merge/split decision based on a local strategy. However, due to the advantage of global functions which take information from the whole image into account at the same time, the global functions have been used increasingly in this field. For example, Markov random fields or variation formulations are lent [3-4]. There are many approaches to group the pixels. The K-means algorithm is an iterative algorithm that is used to partition an image into K clusters [5]. It is easy to implements, and the algorithm is usually very fast. As it is heuristic algorithm, it does not guarantee that it will converge to the global optimum, and the result may depend on the initial values. Graph partitioning methods are an effective tools for image segmentation since they model the impact of pixel neighborhoods on a given cluster of pixels. Normalized cuts, the generic graph based method can be used with many different features and affinity formulations, and provides regular segments [6]. There are some weaknesses such as high storage requirement, time complexity, and bias towards partitioning into equal segments. Superpixels algorithms group pixels into perceptually meaningful regions. SLIC superpixels is a simple and efficient method to decompose an image in visually homogeneous regions [7]. It is based on a spatially localized version of k-means clustering. Rother et al. proposed Grabcut which is a kind of interactive foreground extraction using Iterated graph cuts [8]. If the user selects a rectangle loosely around an object or distinguishes between the regions of background and foreground manually, then the rest of the selected region becomes the background.

It is very difficult to define what a good segmentation is. A result of image segmentation depend on a lot of things. A crucial problem in segmentation is that of splitting up into too few regions called under-segmentation or too many regions called over-segmentation. Under-segmentation ignores to extract important boundaries, and segment an image into too few regions. Whereas over-segmentation increases the chances that boundaries of importance have been extracted, it does so at the cost of creating many insignificant boundaries. In this paper, we address the segmentation method merged hierarchically partial trees of superpixels which forms an over-segmentation of an image. Because superpixels preserve the meaningful

978-1-4799-8641-5/15/$31.00 ©2015 IEEE

boundaries, a result of proposed method guarantees to preserve the important boundaries.

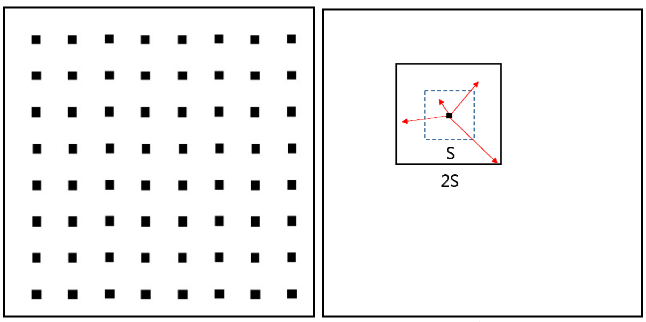
II. SUPERPIXEL

Superpixels extraction is a process of partitioning a digital image into a number of homogeneous segments, which can be used instead of the rigid pixel structure. There are desirable properties to approach an optimal image result. Superpixels ensure the boundary accuracy and the convergence to the real target boundary. We expect that the process of Superpixels needs to maintain low-complexity. The most importance property is the quality of the results.

Given the k cluster center points which the intervals of the image plane are fixed. The size of the interval is $S = \sqrt{N / k}$ so as to generate superpixels. To reduce the noise pixel, the cluster center points move to the points where the texture maps have lowest gradient. In limited regions, we calculate pixels similarity. In order to speed up the process of superpixels, we shrink the size of limited regions, but to increase accuracy, we need to widen the size of limited regions. Typically, the size is used in a region 2S *2S around the cluster center points.

The measurement of similarity between pixels commonly relies on color differences, distance, or texture. Superpixel procedure uses the CIELAB color space and the location of the center pixels to construct vectors $C_i=[\, l_i, a_i, b_i, x_i, y_i]^T$. For each cluster center C_i, computer the distance D between C_i for each pixel i. D is the distance measure, which pixel around the cluster center C_i belong to the cluster. The distance measure D is represented as

$$
\begin{aligned}
d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\
d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\
D &= \sqrt{d_c^2 + (\frac{d_s}{S})^2 m^2}
\end{aligned}
\tag{1}
$$



(a) (b)

Fig. 1. Regular grid and a limited regions for searching.

In practice, we select the regular cluster centers as Fig. 1(a) shown at first. Fig. 1(b) calculates the measurement of similarity from each cluster center to pixels within a 2S x 2S region.

Fig. 2. Superpixels (k=225).

III. SUPERPIXEL AGGREGATION

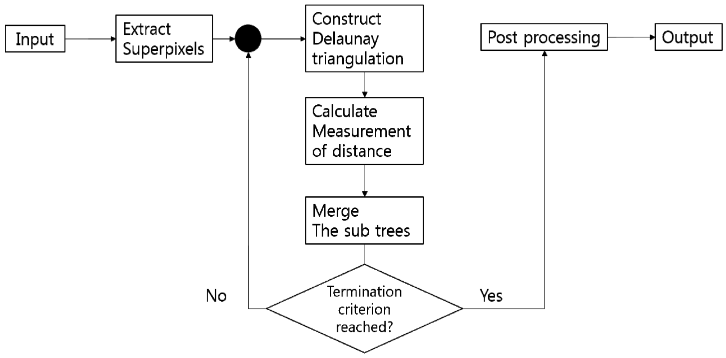


Fig. 3. Flow chart of whole process

The system overview is shown in Fig. 3. We separate our system into three main steps. The outline of the algorithm is as follows: The first step is extracting superpixels. We need to set up the number of superpixels we would like. The next step is constructing Delaunay triangulation to connect each superpixel with the adjacent superpixels. After measurement of distance is calculated, the subtrees are merged. If the termination criterion is not reached, then go to the construction step, otherwise go to the next step. Finally, after doing post processing, we can get the result.

A. Delaunay triangulation

Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation, and it tends to avoid skinny triangles [9]. Given the some points resulted by SLIC superpixels algorithm, we construct Delaunay triangulations [7]. The Delaunay triangulation has the property that the circumcircle of every triangle does not contain any points of the triangulation. The algorithm is also broken down to four main steps. (1) Let P be a set of n points in the Plane, (2) The Voronoi diagram Vor(P) is the subdivision of the plane into Voronoi cells V(p) for all p $\in$ P, (3) Let G be the dual graph of Vor(P), (4) The Delaunay graph DG(P) is the straight line embedding of G. a Voronoi diagram is a subdivision of a plane into regions based on distance to points in a specific subset of the plane. The Delaunay triangulation is the dual structure of the Voronoi diagram. Fig. 4 illustrates the step of the Delaunay triangulations (1)-(4) and circumcircle of some triangles (5).

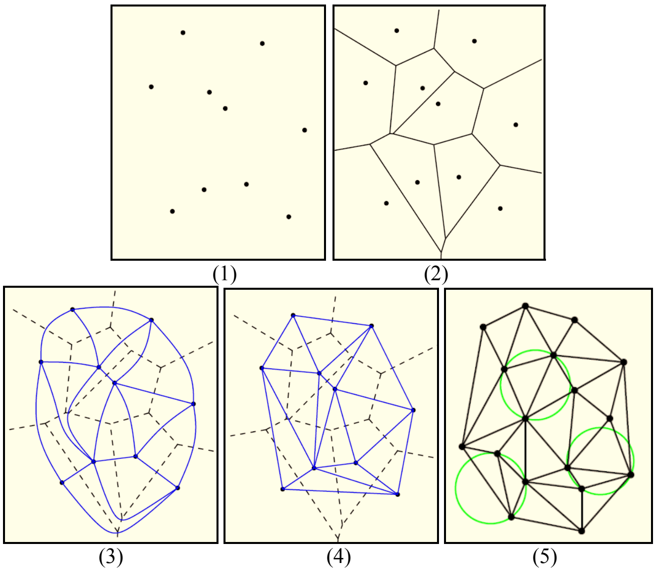


Fig. 4. Delaunay triangulations, Voronoi diagram, and circumcircle of some triangles.

B. Binary tree

A binary tree data structure has degree two. A binary tree is made of nodes, where each node has at most two children. The topmost node in the tree is called the root. Every node in a tree is connected by a directed edge from exactly one other node called a parent. On the other hand, each node can be connected to arbitrary number of nodes, called children. Nodes with no children are called leaves.

In order to combine the group of regions, we use binary tree. The Fig. 5 represents that some binary sub-trees are merged. In the Fig. 5, the merged tree is unbalanced and not sorted. There are some merging methods. We select simple method to merge sub-trees. In practical, we create two sorted arrays by in-order two sub-trees, and merge two sorted arrays, then build the binary tree from sorted array.

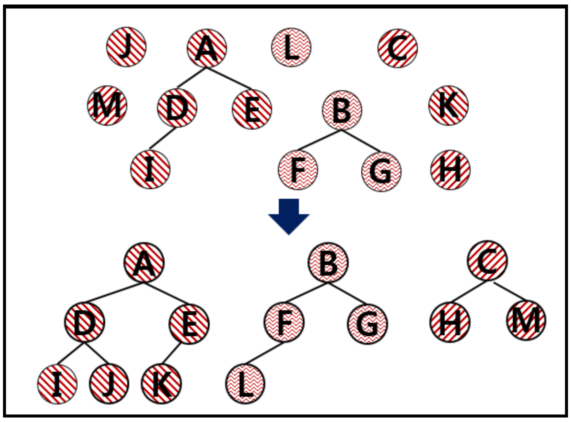


Fig. 5. Merging binary subtrees - three labeled binary trees of size 6, 4, and 3 and height 3, with root nodes whose value are A, B, and C.

C. Hierarchical merging of adjacency regions

The spatial proximity is employed to supervise the merging process, and the color similarity is used to determine the regions are combined or not. In order to calculate the color similarity, we assume that the distributions of pixel values in among the regions are similar to a Gaussian distribution. If the mean of a cluster is in threshold intervals, we determine that the regions are similar, and merge the subtrees by using the binary tree. The threshold intervals of color similarity are represented as

$$\begin{aligned} &If\ \frac{\varsigma_l^{\ j}}{2} < m_l^{\ i} < \frac{\varsigma_l^{\ j}}{2},\ then\ \alpha = 1, otherwise\ \alpha = 0 \\ &If\ \frac{\varsigma_a^{\ j}}{2} < m_a^{\ i} < \frac{\varsigma_a^{\ j}}{2},\ then\ \beta = 1, otherwise\ \beta = 0 \\ &If\ \frac{\varsigma_b^{\ j}}{2} < m_b^{\ i} < \frac{\varsigma_b^{\ j}}{2},\ then\ \delta = 1, otherwise\ \delta = 0 \end{aligned} \tag{2}$$

If $\alpha \ \square\ \beta \ \square\ \partial = 1$ in (2), we merge the similar regions. However, we use Delaunay triangulation, and merge the regions iteratively. Some meaningful regions do not join the other subtree. Therefore, we use Canny edge detection to extract the edges in the image, and then Calculating the amount of overlap between regions. The overlap region can be meaningful boundaries.

IV. Experimental Results

To evaluate the proposed algorithm in the paper, we employ the test database: Berkeley segmentation dataset. Some segmentation results from the experimental images are shown in Fig. 6. The segmentation results demonstrate that our algorithm is not an ideal method which can computer accurate regions, but it generated meaningful regions. Fig. 6(d) is used for ground truth to evaluate segmentation algorithms. When we compare with 6(d) and others, the results of the previous approaches show edges in the object, but proposed results show that the edges are reduced.

V. Conclusion

In this paper, we present a new segmentation method for color texture image. In order to avoid over-segmentation, we consider similarity and proximity of each center of the cluster. Given the centers of superpixels, we determine the center of superpixels to each subtree, and construct a Delaunay triangulation to compute which regions are adjacent. Similar clusters are merged by using the similarity measure. We use the binary tree to merge the each region iteratively. Experimental results show that our method provides more accurate segmentation compared with other segmentation methods. However, the results demonstrate that our algorithm is not an ideal method. Therefore, future researches need to find a more accurate measurement of similarity.

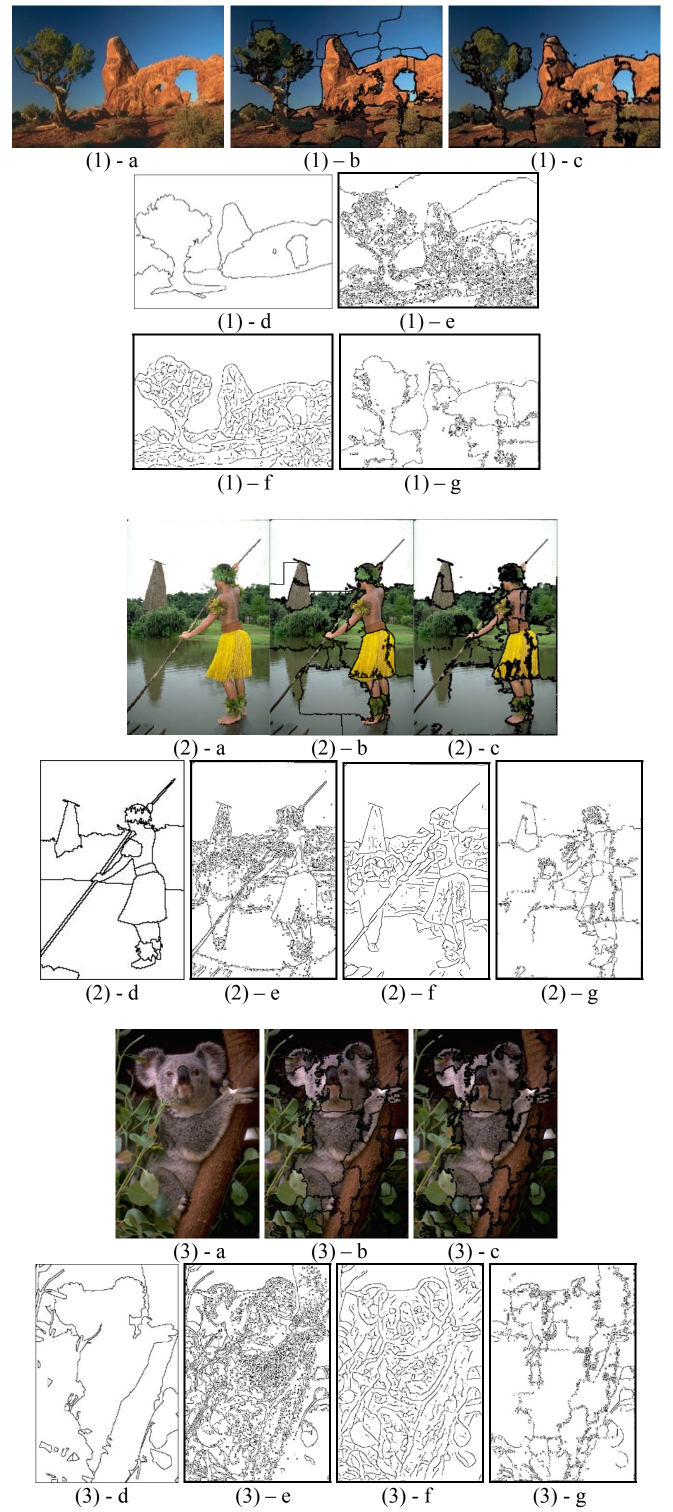


Fig. 6. Comparison of some segmentation results used BCT[10] and human boundaries in database Berkeley Segmentation Dataset. (a) Original images, (b) Segmentation results of proposed meth-od. (c) Segmentation results of proposed method with post processing, (d) Multi-Human seg-mentation results (e) Boundaries of k-means segmentation, (f) Boundaries of BCT, (g) Bounda-ries of proposed results.

ACKNOWLEDGMENT

This research was supported by the ‘Cross-Ministry Giga KOREA Project’ of the Ministry of Science, ICT and Future Planning, Republic of Korea(ROK). [GK13C0100, Develop-ment of Interactive and Realistic Massive Giga-Content Tech-nology]

REFERENCES

[1] M. Wertheimer. ‘Laws of organization in perceptual forms (partial translation),” A Sourcebook of Gestalt Psychology, pp. 71–88, 1938.

[2] L. G. Shapiro and G. C. Stockman “Computer Vision,” New Jersey, Prentice-Hall, pp. 279-325, 2001.

[3] S. Geman, and D. Geman, “Stochastic relaxation, Gibbs distribution, and the Bayesian retoration of images” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, pp. 721–741, 1984.

[4] D. Mumford, and J. Shah, “Optimal approximations by piecewise smooth functions, and associated variational problems”, International Journal of Communications on Pure and Applied Mathematics., vol. 42, no. 5, pp. 577–684, 1989.

[5] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y.Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” Proc. Computer Vision and Pattern Recognition, vol. 29, no. 9, pp.881-892, 2002.

[6] J. Shi, and J. Malik. "Normalized cuts and image segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, 2000.

[7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, 2012.

[8] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp.1222-1239, 2001.

[9] D. Lee, and B. J. Schachter. "Two algorithms for constructing a Delaunay triangulation," International Journal of Computer & Information Sciences vol. 9, no. 3, pp.219-242, 1980.

[10] T. Leung, J. Malik, “Representing and recognizing the visual appearance of materials using three dimensional textons,” International Journal of Computer Vision, vol. 43, no. 1, pp. 29–44, 2001.

[11] C. Fowlkes, M. David, and M. Jitendra. "Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches." Proc. Computer Vision and Pattern Recognition, vol. 32, pp. 8092-8096, 2003.