International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

Image Segmentation Based on Graph Theoretical Approach to Improve the Quality of Image Segmentation

Deepthi Narayan, Srikanta Murthy K., and G. Hemantha Kumar

***Abstract*—**Graph based image segmentation techniques are considered to be one of the most efficient segmentation techniques which are mainly used as time & space efficient methods for real time applications. How ever, there is need to focus on improving the quality of segmented images obtained from the earlier graph based methods. This paper proposes an improvement to the graph based image segmentation methods already described in the literature. We contribute to the existing method by proposing the use of a weighted Euclidean distance to calculate the edge weight which is the key element in building the graph. We also propose a slight modification of the segmentation method already described in the literature, which results in selection of more prominent edges in the graph. The experimental results show the improvement in the segmentation quality as compared to the methods that already exist, with a slight compromise in efficiency.

***Keywords*—**Graph based image segmentation, threshold, Weighted Euclidean distance.

I. Introduction

IMAGE segmentation refers to a process of dividing the image into disjoint regions that are meaningful. This process is fundamental in computer vision in that many applications, such as image retrieval, visual summary, and image based modeling can benefit from it. This process is also challenging because segmentation is usually subjective and computation is highly costly. Image is simply a collection of pixels. It is quite difficult to extract the knowledge (the high-level information) directly from the low-level pixel collection. Graph-based segmentation takes into account global image properties as well as local spatial relationships, and results in a region map that is ready for further processing, e.g. region merging or labeling.

As the methods described in the literature are concerned, the graph-based image segmentation method proposed by Felzenszwalb et a1[1] & an improvement to it by Ming Zhang et al [2] are the most efficient ones with satisfactory segmentation results. Though the methods described in [1] & [2] turn out to be efficient, there is need to improve the quality of segmented images. Consequently, we have developed a suitable method that improves the segmentation quality.

The method proposed in [1] uses a simple but effective modification of Kruskal's algorithm. This method addresses the problem of segmenting an image into regions by defining a predicate for measuring the evidence for a boundary between two regions using a graph-based representation of the image. An important characteristic of the method is its ability to preserve detail in low-variability image regions while ignoring detail in high-variability regions.

The method in [1] has several identified drawbacks. Firstly, the internal difference is defined on the extreme values, which is not the accurate description of the components. Secondly, the threshold function requires a user specified parameter k to control the size of the segmented region. However, no quantitative relationship between the value of k and the segment size is given. It is very difficult for users to choose an appropriate value for an expected segmented size.

Ming Zhang et al. [2] proposed a major improvement to [1]. They propose the method for sensor devices which are used for monitoring purposes. It is very helpful to develop a method which is time & space efficient due to the limited resources of sensor devices. The method contributes to the method in [1] by re-defining the internal difference which is used to define the property of the components. The internal difference is re-defined to give a more stable and accurate description of the components. It also re-defines the threshold function which is the key element to determine the size of the components. The threshold function is re-defined so that it can adaptively guide the segmentation process independent of the edge weight scale. This method overcomes all the limitations of the method in [1].

The article on colour metric [3] suggests the use of a weighted Euclidean distance in RGB color space. Explicitly, the methods in [1] & [2] use a Euclidean method to calculate the edges weight in the graph. It is one of the key elements in the construction of the graph, which determines the segmentation result. However, if the color images are considered, it is not just enough to consider the distance between two points. If we consider color RGB images containing red, green & blue components, there is need to associate some weight to each of these components in the

Deepthi Narayan is with the Department of Computer Science & Engineering, PES Institute of Technology, Bangalore, India, as a student of M.Tech. (e-mail: deepthinarayan1981@gmail.com).

Srikanta Murthy K is with the Department of Information Science & Engineering, PES Institute of Technology, Bangalore, India.

G. Hemantha Kumar is with the Department of Studies in Computer Science, University of Mysore, Mysore.

RGB color space. So we propose the use of a weighted Euclidean function described in [3]. This is to ensure that a definite weight is associated with each of the red, green & blue components. We use the weighted Euclidean distance [3] in order to compute the edge weight.

Furthermore, a minor contribution is made to the segmentation algorithm in [2]. The algorithm in [2] is slightly modified so as to select more prominent edges.

These two modifications results in an improvement of the quality of segmented images. The reported experimental results on a well defined set of images demonstrate the effectiveness of the proposed approach.

The rest of the paper is organized as follows. Section II presents the basic improvement to the graph-based method. Section III describes the implementation. Section IV reports the experimental results & discussion. Section V is summary and conclusions.

II. Improvement to the Graph based Method

The image segmentation algorithm described in [2] starts with a trivial segmentation, with each component containing one pixel, and repeatedly merges pairs of components. Since the method in [2] has already addressed the drawbacks of method described in [1], only the method proposed in [2] is analyzed.

The methods in [1] & [2] use Euclidean distance to calculate the edge weight. However, if the color images are considered, it is not just enough to consider only the distance between two points. There is need to associate a definite weight to each of the red, green & blue components. So, we consider the weighted Euclidean equation given in [3].

Given an image represented by a weighted graph, the largest edge weight is denoted by W_{max}, the smallest edge weight by W_{min}. Num_c is the number of components in the image. Parameter k can be regarded as the expected number of components. Larger k produces more components. The threshold function is defined as,

$$T(C) = (W_{max} - W_{min} / |C|) * (Num_c / k) \qquad (1)$$

The internal difference is defined such that it gives a more accurate description of component C. The internal difference C is defined as the average edge weight in the minimal spanning tree of C. N is the number of edges in the minimal spanning tree of C.

$$Int(C) = 1 / N * \sum_{e \in MST(C, E)} W(e) \qquad (2)$$

The proposed method considers both of these equations and constructs the graph using the same method described in [2].

However, the segmentation algorithm proposed in [2] is slightly modified so that more prominent edges are selected. An additional condition is added when components are merged. The condition checks that the size of the component & weight of the edge to be merged is less than the threshold values of both the components connected by the edge. This selective merge of components results in selecting prominent edges, which the method proposed in [2], has not incorporated.

So the additional condition that needs to be added to the existing algorithm in [2] would be:

```
if (( sizeof the component to be merged  && edge  to be merged )< threshold values of both the components connected by the edge))
{
    merge the two components
    Increment the counter used to count the number of
    components
}
```

The addition of this condition for merging of the components & the use of weighted Euclidean function to compute the weight of the edge results in better quality of the segmented images, with a slight compromise in efficiency.

III. The Implementation

In this implementation, the image is smoothed by a Gaussian filter to remove noise. The edge weight is computed by using the following equation [3].

$$|\Delta C| = \sqrt{2 \times \Delta R^2 + 4 \times \Delta G^2 + 3 \times \Delta B^2} \qquad (3)$$

where ΔR, ΔG & ΔB represent the difference between the intensity values of red, green & blue components in a 2 dimensional space. 2,4, & 3 represent the weights assigned for red, green & blue components respectively.

After the graph is segmented, the segmentation result is post-processed by combining all the components with size below a user-specified threshold.

We borrowed some of the code from the author's implementation in [1], like the Gaussian filter and the data structure to store the components. We did the following modifications:

The nearest neighbor graph method specified in [4] is implemented Each pixel has edges connecting 10 nearest neighbors on the 5-dimensional feature space (r, g, b, x, y), where the triplet (r, g, b) represent the color value of the pixel and the pair *(x,* y) represent the location of the pixel. We used the weighted Euclidean formula [3] given in equation (3). For efficiency, we use the nearest neighbors within distance 10.

Finally, in our implementation, we did not do any post-processing and achieved an improvement in the quality of segmented images as reported in section IV.

IV. Experimental Results and Discussions

In the experiment, our method is compared with the work described in [1] & [2], which are fast ones with satisfactory segmentation result. As our proposed method aims at improving the quality of the segmented images, we do not compare the running time of the approaches. Only the segmentation result and the segmented images are compared.

We use the test images used by the authors in [1] & [2] and also from the Vision Texture Database [4].

The first example is shown in Fig. 1 (a); the input is a tower image. According to human perception, we expect the sky to be one component, the background grass to be another component & the tower to be a component. Fig. 1 (b) is the segmented image produced by the method described in [1] with parameter value 300. Fig. 1 (c) shows the segmented image obtained from [2] with parameter value 80. Fig. 1 (d) shows the result from the proposed method. It can be observed that for the same parameter value, the proposed method gives better results.

Fig. 1 a Input tower image

Fig. 1 (b) Segmented image with k=300 from [1]

The second example is shown in Fig. 2 (a); the input is a leopard image. We expect the leopard to be segmented from the background. Fig. 2 (b) shows the result obtained using [1] using parameter value 300. Fig. 2 (c) shows the output obtained using [2] with parameter value 20. Fig. 2 (d) shows the output image obtained using the proposed method.

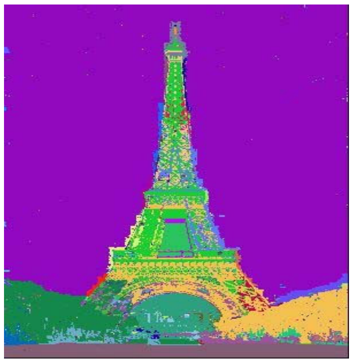

Fig. 1 (c) Segmented image with k=80 from [2]

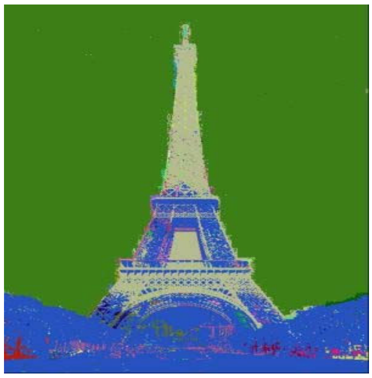

Fig. 1 (d) Segmented image with k=80 from proposed method

The experiments were also carried out with test images from Vision Texture database [8]. Fig. 3 (a) shows the input gray scale image. Fig. 3a shows the result from [1], Fig. 3 (b) presents the results from [2] & Fig. 3 (c) shows the results from the proposed method with parameter value 150. For gray level images, results from [2] & the proposed methods both yield good results with the parameter value 20.

Fig. 2 (a) Input Leopard image

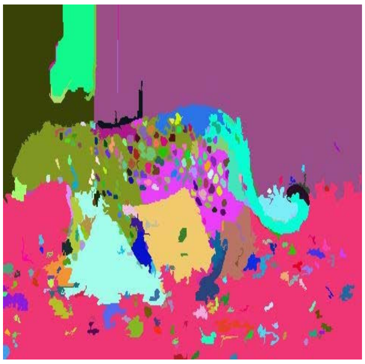

Fig. 2 (b) Segmented image with k=300 from [1]

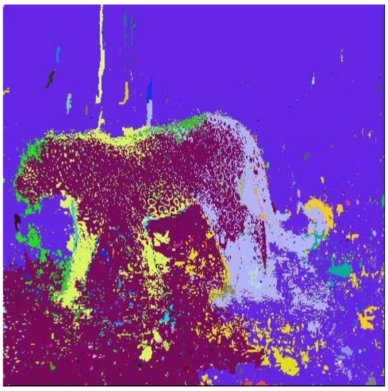

Fig. 2 (c) Segmented image with k=80 from [2]

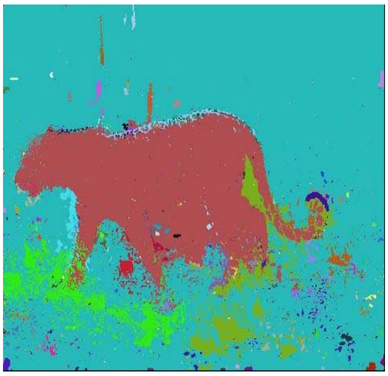

Fig. 2 (d) Segmented image with k=80 from the proposed method

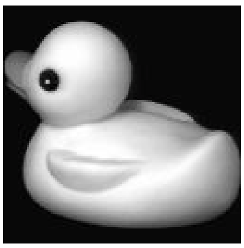

Fig. 3 (a) Input swan image

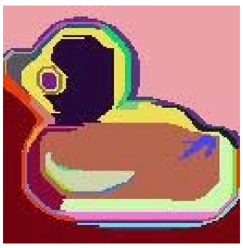

Fig. 3 (b) Result from [1] with k=150

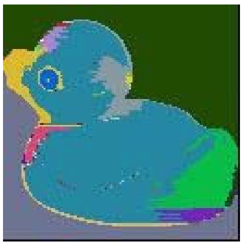

Fig. 3 (c) Result from [2] with k=20

Fig. 3 (d) Result from the proposed method with k=20

V. SUMMARY AND CONCLUSIONS

In this paper, we compared the existing segmentation approaches in terms of image features, similarity measurement & segmentation algorithm in order to improve the segmentation quality. We analyzed the graph based segmentation methods in [1] & [2]. We propose an improvement to the methods [1] & [2].

- We proposed the use of weighted Euclidean distance in order to compute the edge weight for RGB color images.
- We also modified the segmentation algorithm slightly so that more prominent edges are selected.

Finally, the reported experiments on a well defined set of images demonstrate the effectiveness & the improved segmentation quality of the proposed approach.

REFERENCES

[1] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Graph-Based Image Segmentation," International Journal of Computer Vision, Vo.59, No.2, 2004.

[2] Ming Zhang, Reda Alhajj, "Improving the Graph-Based Image Segmentation Method "Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'O6), 2006, IEEE.

[3] Thiadmer Riemersma, Color metric, available at http://www.compuphase.com/cmetric.htm

[4] S.Arya and D.M. Mount, "Approximate nearest neighbor searching", Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 271-280, 1993.

[5] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, Vo1.22, No.8, pp.888- 905, 2000.

[6] C.T. Zahn, "Graph-theoretic methods for detecting and describing gestalt clusters", IEEE Transactions on Computing, vol 20, pages 68-86, 1971.

[7] P.F. Felzenszwalb and D.P. Huttenlocher," Image segmentation using local variation" Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 98-104, 1998.

[8] Test images for experimenting from Vision Texture Database. available at http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[9] Test images from Berkeley Segmentation Dataset: Images available at http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images/color/134052.html

[10] Jing Dong Wang, PhD. thesis on graph based image segmentation, Hong Kong University, 2007.

[11] Gonzales R C and Woods R E, Digital Image Processing, 2nd ed., Pearson Education Asia, 2002.