

Image Segmentation Using Quadtree-Based Similarity Graph and Normalized Cut

Marco Antonio Garcia de Carvalho, Anselmo Castelo Branco Ferreira,
and André Luis Costa

School of Technology - FT, State University of Campinas - UNICAMP
Rua Paschoal Marmo, 1888, Jd Nova Itália, Limeira, São Paulo, Brazil
magic@ft.unicamp.br

Abstract. The graph cuts in image segmentation have been widely used in recent years because it regards the problem of image partitioning as a graph partitioning issue, a well-known problem in graph theory. The normalized cut approach uses spectral graph properties of the image representative graph to bipartite it into two or more balanced subgraphs, achieving in some cases good results when applying this approach to image segmentation. In this work, we discuss the normalized cut approach and propose a Quadtree based similarity graph as the input graph in order to segment images. This representation allow us to reduce the cardinality of the similarity graph. Comparisons to the results obtained by other graph similarity representation were also done in sampled images.

Keywords: image segmentation; quadtree; graph partitioning; spectral graph.

1 Introduction

The image segmentation issue has been studied by many authors as a way to distinguish different objects from a scene. Regarding this process as a graph partitioning problem, a graph cut, is a promising area and there are a lot of recent studies on this field. Graph cut is a measure that divides a graph into two disjoint sets. Therefore, an initial challenge consists to carry out the image-graph conversion adequately.

The Spectral Graph Theory (SGT) [3] studies the graph's matrix eigenvalues and eigenvectors, their relation with the graph's features and the use of eigenvectors for graph bipartition. There are several applications on this field [7, 14, 16]. The concepts of SGT originated lots of graph partition techniques, such as the Normalized Cut [12] and the Average Cut [13].

The Normalized Cut (NCut) image segmentation technique [12] segments an image by minimizing the cut cost of a weighted graph. This cost is calculated as a fraction of the total edge connections of each partition. One problem in this technique is to defeat the high computational cost demanded as the graph size increases. In order to avoid this problem, there are several ways of generating

the input graph for the segmentation, with different results and applications [12, 10, 2, 5].

In this work, we propose an alternative graph representation as input for the NCut, instead of the commonly used pixel affinity graph. This graph, the Quadtree-based similarity graph, is generated from the quadtree leaves. It uses as decomposition criterion an edge detection operation. We show that this approach reduces the graph size and, consequently, the computational cost. The results are similar to the obtained by the other similarity graphs, using the same technique, the NCut.

This paper is organized as follows. Section 2 introduces the NCut technique and presents some related works. Some kinds of graphs that can be used as input for this technique, including our Quadtree based similarity graph are described in section 3. An overview of the proposed approach is given in section 4. Experiments in sampled images are done in section 5 and further comments of the experiments, conclusions, as well as suggestions of future work are done in section 6.

2 Related Works

The Normalized Cut technique [12] is a theoretic method for graph partitioning. Its goal is to find a balanced cut in a graph, in order to generate two or more sub-graphs. Applying this method for image segmentation is possible with a proper image-graph representation. The subgraphs obtained from graph partitioning represents the image regions.

The Normalized Cut in a graph G is calculated by (1), as follows:

$$\text{NCut}(A, B) = \frac{\text{cut}(A, B)}{\text{SumCon}(A, G)} + \frac{\text{cut}(A, B)}{\text{SumCon}(B, G)}, \quad (1)$$

where A and B are subgraphs, subject to $A \cup B = G$ and $A \cap B = 0$; $\text{cut}(A, B)$ is defined as the total weight of the edges removed from the graph, $\text{SumCon}(A, G)$ is the total weight of the edges connecting nodes from a subgraph A to all nodes in the original graph G ; and $\text{SumCon}(B, G)$ is similarly defined to a subgraph B . The optimal NCut is the one that minimizes (1), but minimizing it is a NP-Complete complexity problem [12]. However, by expanding (1), the authors noticed that it can be minimized using spectral graph properties of the graph's Laplacian Matrix described by Fiedler [6].

There is a wide range of recent work in image segmentation using the NCut technique. In [12], the similarity graph is built by taking each pixel as a node. Then, the node pairs within a given radius r are connected by an edge. Monteiro and Campilho [10] proposed the Watershed Normalized Cut, which uses the regions from the Watershed image segmentation as nodes for the similarity graph. The Watershed region similarity graph is either used in [2] for comparison with the primitive pixel affinity graph in yeast cells images segmentation.

The primitive NCut enhancement was also studied and applied by many researchers. Cour *et al* [5] proposes a NCut adaptive technique that focus on the

computational problem created by long range graphs. The authors suggested the use of multi-scale segmentations, decomposing a long range graph into independent subgraphs. The main contribution of this technique is that larger images can be better segmented with a linear complexity. Sun and He [15] purposed the use of the multiscale graph decomposition, partitioning the image graph representation at the finest scale level and weighting the graph nodes using the texture features.

3 Graph Representation

A graph representation of the image is needed to perform the NCut segmentation approach. Basically, this representation is done by an undirected weighted graph $G = (V, E, W)$, where: (i) V is the nodes set, where each node corresponds to a region or a pixel of the image; (ii) E is the edges set, where each edge links two nodes, and consequently, make a relationship between two regions or pixels of the image; (iii) and W is the weights set, where each weight is related to an edge and corresponds to a measure of similarity between the regions on the relationship. This structure is called the Similarity Graph.

There are several techniques to construct the similarity graph of a image. Some of these techniques, used by us in this work, are described in the following subsections.

3.1 Pixel Affinity Graph

Each pixel is taken as a graph node, and two pixels in a r distance are connected by an edge. The edges weights should reflect the similarity between the pixels connected by them. The grouping cue used in the similarity function will reflect the overall quality of the segmentation. Some of them are the intensity, position, and contours [12, 5, 8].

The intensity and position grouping cue assumes that close-by pixels with similar intensity are most probably to belong to the same object. The measure of similarity regarding this grouping cue is given by (2) [12, 5]:

$$W_{IP}(i, j) = \begin{cases} e^{-\left(\frac{\alpha^2}{d_p}\right) - \left(\frac{\beta^2}{d_i}\right)}, & \text{if } \alpha_2 < r \\ 0, & \text{Otherwise} \end{cases} \quad (2)$$

where $\alpha = ||P_i - P_j||$ and $\beta = ||I_i - I_j||$ are respectively the distance and the difference of intensity between pixels i and j ; r is a given distance (also called graph connection radius); and d_p and d_i could be set with the variance of the image pixels positions and intensity. This grouping cue used separately often gives bad segmentations because some natural images are affected by the texture clutter.

The measure of similarity regarding the intervening contours grouping cue is given by (3) [5]:

$$W_C(i, j) = \begin{cases} e^{-\left(\frac{\max_{x \in \text{line}(i, j)} \varepsilon^2}{d_c}\right)}, & \text{if } \alpha_2 < r \text{ ,} \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

where $\text{line}(i, j)$ is a straight line joining pixels i and j and $\varepsilon = \|\text{Edge}(x)\|$ is the image edge strength at location x .

These two grouping cues can be combined as shown by (4) [5]:

$$W_{IPC}(i, j) = \sqrt{W_{IP}(i, j) W_C(i, j)} + W_C(i, j). \quad (4)$$

Multiscale Graph Decomposition. The Multiscale Graph Decomposition algorithm [5] works on multiple scale of the image to capture coarse and fine level details. The construction of the image segmentation graph is given according to $W = W_1 + W_2 + \dots + W_s$, where W represents the graph weights $w(i, j)$ and s , the scale, i.e., each W_s is an independent subgraph. Two pixels i, j are connected only if the distance between them is lower than G_r . As G_r value is a tradeoff between the computation cost and the segmentation result. W_s can be compressed using recursive sub-sampling of the image pixels. This compression is not perfect, but he has the advantage of the computational efficiency.

3.2 Quadtree-Based Similarity Graph

The term Quadtree is used to describe a class of hierarchical data structures whose common property is that they are based on recursive decomposition of space. They can be classified on the following bases [11]: (i) the type of data that they are used to represent, *i. e.*, points, regions, volumes, etc; (ii) the principle guiding the decomposition, that can be fixed or based on the input data; (iii) the resolution, that can be variable or not.

In order to represent an image through a Quadtree, its regions should be recursively decomposed into exact four new disjoint regions, when they satisfy a defined criterion. The initial region corresponds to the whole image and is associated to the tree root node [4, 11].

Defining the criterion to decompose the regions of the Quadtree is not a trivial task. There are different criteria that can be used, as standard deviation or entropy of image gray levels [4]. We found that using the image edges for guiding the regions decomposition was very adequated, because: (i) the edge detection operation drastically reduce the size of data to be processed, while at the same time preserves the structural information about object boundaries [1]; (ii) the edge detection results in a binary matrix. Then, became trivial to define that a region should be decomposed when it is not formed entirely by 1's or 0's [11]. Figure 1 shows one grayscale image with 256 x 256 pixels, its edge detection by Canny filter and a reconstruction based on regions associated with

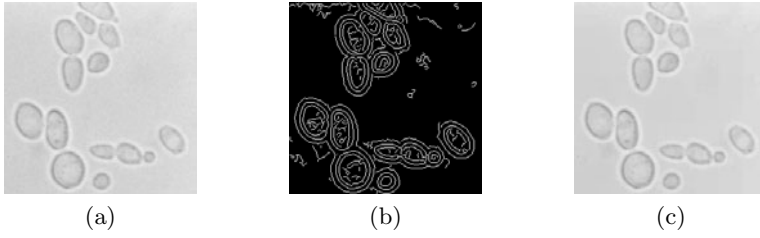


Fig. 1. (a) Original image with 65536 pixels (256 x 256) (b) edge detection resulting from applying Canny filter to 1a (c) image reconstruction with 14749 regions

the Quadtree leaves. Can be observed that the image reconstruction, showed in Fig. 1c, is very similar to the original one.

The main goal of using a Quadtree image representation is to reduce the similarity graph size, used as input to the NCut segmentation technique. For this purpose, the input graph will be generated with basis on the regions associated to the Quadtree leaves. Each region will be associated to a graph node. For instance, for the image showed in Fig. 1a, the resulting similarity graph using the Quadtree-based approach has 14749 nodes, about 22,5% of the total nodes obtained by using the Pixel Affinity approach (65536 nodes).

The number of regions obtained by the proposed technique will vary in function of the image data. Also, the parameters of the edge detection filter can be manually specified, in order to change its sensibility. It means that the number of nodes on the similarity graph can be influenced by the choice of the edge detector parameters.

4 Algorithm Overview

The segmentation based on NCut technique can be applied by two distinct methods: recursive 2-way NCut and k -way NCut. The first one uses the second smallest eigenvector of the graph Laplacian's matrix L , where $L = D - W$ with W being the weight matrix and D a diagonal degree matrix, to recursively bipartite the similarity graph [12]. The k -way NCut uses the K first eigenvectors of the graph Laplacian's matrix L to directly generate a number K of desired partitions [12].

The image segmentation process using k -way NCut is described in the following steps:

1. Given an input image, compute the Similarity Graph $G = (V, E, W)$ using one of the techniques described in section 3.
2. Build the weight matrix W and the degree matrix D from the Similarity Graph.
3. Solve $(D - W)x = \lambda Dx$
4. Discretize the K first eigenvectors into X , where $X = [X_1, X_2, \dots, X_K]$ and $X_N[i] = 1$ iff node i belongs to the partition N .
5. Use X for the distribution of the graph nodes into the K partitions.

5 Experiments

We used in our experiments a set of 15 randomly chosen images from the Berkeley Image Database [9] and more 10 images from a particular database. In order to make a regular decomposition, the quadtree implementation requires squared images with size 2^n , where n is a positive integer. The images of the Berkeley Database needed to be cropped to 256 x 256 pixels due to this restriction.

The experiments were executed according to the steps described in section 4. For the Pixel Affinity Graphs and the Quadtree-based Similarity Graphs, the similarity between node's relations was calculated by (3), while for the Multiscale Decomposition Graphs it was calculated by (4). The connection radius was, respectively, $r = 10$, $r = 20$ and $r = \{2, 3, 7\}$. Note that for the Multiscale method was generated three scales, and there are one correspondent radius for each scale. Given the irregularity of the region's size, the connection radius between them on the Quadtree representation is given by (5)

$$Radius = \frac{\max(RS_a, RS_b)}{2} + r, \quad (5)$$

where RS_a and RS_b are the sizes of the two regions being connected and r is the radius given by the user. The k -way NCut was used with $K = 30$, yielding to 30 regions for each images segmentation. Fig. 2 shows five selected results.

We use the original implementations of the Pixel Affinity and Multiscale segmentation provided by the authors [12, 5].

As observed on experiments, the NCut with Quadtree-based Similarity Graph presented results as good as with the Pixel Affinity. There are minor differences on the resulting segmentations with these two techniques. However, our technique has a lower computational cost due to the reduced number of nodes on the similarity graph. For the 25 images used in the experiments, the average number of nodes on the similarity graph was 18755, about 28.62% of the total number of pixel on the images. Table 1 shows statistics about the nodes quantity for our particular database and from the Berkeley Database.

Table 1. Number of nodes for Quad-Tree Based Similarity Graph

	Mean	Higher	Lower
Particular Database	16177,6	24478	9040
Berkeley Database	20473,4	30751	9700

When comparing our technique with the Multiscale method, the differences on the results are more expressive for some images, see Fig. 2c and Fig. 2d, once, in this work, these technique uses different similarity functions. Nevertheless, these two techniques proposes a way to reduce the computational cost of NCut segmentation. The main advantage of our technique is that the similarity graph size can be controlled. However, this control is limited by impacts on the

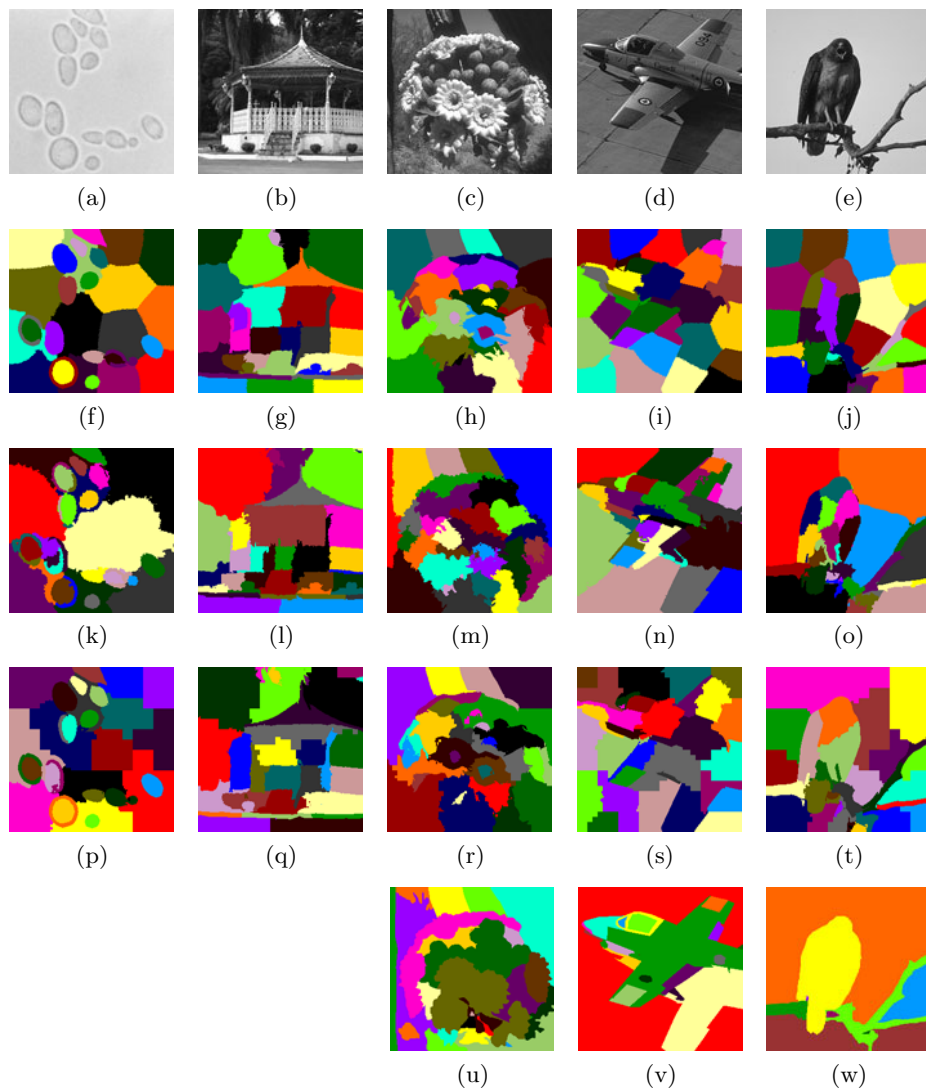


Fig. 2. Ncut image segmentation results obtained by different graph representations. (a-e) Original images, respectively: Yeast Cells and Coreto from particular database and 19021, 37073, 42049 from Berkeley Database. (f-j) Results from Pixel Affinity Graph. (k-o) Results from Multiscale Graph Decomposition. (p-t) Results from Quadtree-based Similarity Graph. (u-w) Ground Truth segmentations for the Berkeley images.

segmentation quality. It is important to note that the overall quality of the proposed technique rely on the edge detection filter efficiency. The edge detection uses Canny filter.

6 Conclusion

In this paper we proposed a novel input similarity graph in order to segment images by NCut approach. We showed that the utilization of the Quadtree-based Similarity Graph provides similar results when compared to the two classical similarity graph representations. Experiments on real images (particular and Berkeley databases) show that the new representation had the advantage of significantly reducing the number of graph nodes.

Using regions instead of pixels seems to be a better strategy to segment images by NCut approach. We would like to explore the Quadtree-based Similarity Graph and compare its results with other obtained from regions similarity graphs.

Acknowledgments. This work is supported by CAPES Brazilian Agency and FAPESP (2009/10266-2).

References

1. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8(6), 679–698 (1986)
2. Carvalho, M.A.G., Ferreira, A.C.B., Pinto, T.W., Cesar Jr., R.M.: Image segmentation using watershed and normalized cuts. In: *Proc. of 22th Conference on Graphics, Patterns and Images (SIBGRAPI)*. Rio de Janeiro, Brazil (2009)
3. Chung, F.: *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, vol. 92. American Mathematical Society, Providence (1997)
4. Consularo, L.A., Cesar Jr., R.M.: Quadtree-based inexact graph matching for image analysis. In: *Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2005)*, Natal - Brazil, pp. 205–212 (2005)
5. Cour, T., Bénézit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR 2005*, vol. 2, pp. 1124–1131 (2005)
6. Fiedler, M.A.: Property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czech Math Journal* 25(100), 619–633 (1975)
7. Malik, J.: Visual grouping and object recognition. In: *Proc. of 11th International Conference on Image Analysis and Processing*, pp. 612–621 (2001)
8. Malik, J., Belongie, S., Shi, J., Leung, T.: Textons, contours and regions: cue integration in image segmentation. In: *Proc. of IEEE International Conference on Computer Vision*, Corfu, Greece, pp. 918–925 (1999)
9. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. of 8th Int'l Conf. Computer Vision*, vol. 2, pp. 416–423 (July 2001)
10. Monteiro, F.C., Campilho, A.: Watershed framework to region-based image segmentation. In: *Proc. of IEEE 19th International Conference on Pattern Recognition - ICPR*, pp. 1–4 (2008)
11. Samet, H.: The quadtree and related hierarchical structures. *ACM Computing Surveys* 16(2), 187–261 (1984)

12. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-22(8), 888–905 (2000)
13. Soundararajan, P., Sarkar, S.: Analysis of mincut, average cut and normalized cut measures. In: *Workshop on Perceptual Organization in Computer Vision* (2001)
14. Spielman, D.: Spectral graph theory and its applications. In: *Proc. of 48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 29–38 (2007)
15. Sun, F., He, J.P.: A normalized cuts based image segmentation method. In: *Proc. of II International Conference on Information and Computer Science*, pp. 333–336 (2009)
16. Tolliver, D.A., Miller, G.L.: Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR 2006*, vol. 1, pp. 1053–1060 (2006)