Image Segmentation Using Component Tree and Normalized Cut

Marco Antônio Garcia de Carvalho
André Luis da Costa
Anselmo Castelo Branco Ferreira
Faculty of Technology - FT
University of Campinas - UNICAMP
Limeira-SP, Brazil
magic@ft.unicamp.br

Roberto Marcondes César Júnior
Department of Computer Science - IME
University of São Paulo - USP
São Paulo-SP, Brazil
cesar@ime.usp.br

***Abstract*—Graph partitioning, or graph cut, has been studied by several authors as a way of image segmenting. In the last years, the Normalized Cut has been widely used in order to implement graph partitioning, based on the graph spectra analysis (eigenvalues and eigenvectors). This area is known as Spectral Graph Theory. This work uses a hierarchical structure in order to represent images, the Component Tree. We provide image segmentation based on Normalized Cut, with image representation based on the Component Tree and on its scale-space analysis. Experimental results present a comparison between other image representations, as pixel grids, including multiscale graph decomposition formulation, and Watershed Transform. As the results show, the proposed approach, applied to different images, presents satisfying image segmentation.**

***Keywords*-image segmentation; component tree; watershed transform; graph partitioning; spectral graph.**

I. Introduction

Image segmentation is a classical and basic issue in computer vision and image processing domains. Its goal is to identify regions of interest on images that have one specific meaning to the problem application. Despite the effort made in the academic community, there is no algorithm or approach able to present optimal or correct image segmentation to all kinds of problems.

The relationship between some linear algebra concepts and graph theory provides a promising way to segment an image into meaningful regions and to extract usefull information. This area, known as Spectral Graph Theory, has evolved and encouraged numerous works on digital image processing domain [1]–[7]. Some interesting applications are pattern recognition, testing isomorphism and clustering, besides image segmentation [8], [9]. Spectral Graph Theory is the study of the eigenvalues and eigenvectors of matrices associated with graphs [8] and provides a way to find a graph cut. Graph cuts, or graph partitioning, may be explored to analyze the degree of dissimilarity between parts of a graph that represents an image.

Recently, a common formulation used in image segmentation is the Normalized Cut [1]. The authors propose a new criterion for measuring the quality of an image partition. This criterion computes the cut cost as an fraction of the total edge connections to all nodes in the graph. Normalized Cut (NCut) has been used by different works in the last years [2]–[5].

In this paper, we propose an approach in order to implement image segmentation based on the NCut applied to a hierarchical image representation, the Component Tree (CT). The CT is a graph image representation computed from the cross-section decomposition of the image gray levels [10], [11]. The Component Tree has been proven effective in many applications of image segmentation. Its implementation is relatively simple and the number of nodes and edges is lower than when using graphs. Additionally, the complete image segmentation is achieved by the analysis of the Reverse Component Tree (RCT) and from the scale space analysis generated by CT and RCT. We would like to explore the state of modeling images by trees and its application on NCut segmentation, not previously done. We have compared our approach with three other formulations that uses NCut partitioning: (i) grid pixels affinity [1]; (ii) graphs based on watershed transform [2], [3]; and (iii) a multiscale graph decomposition [5]. As experiments, we have used different kinds of images and the results are promising.

This paper is organized as follows. Section II presents an overview of the NCut and other related works. Section III describes graph representation used in order to model images, including the Component Tree and the Reverse Component Tree representations. Section IV presents the NCut algorithm overview used by our work. We describe the steps needed to segment an image by means of NCut, as well as how we implement the CT and RCT analysis. Finally, experimental results are presented in Section V and comments and conclusions, as well as suggestions for future works are give in Section VI.

II. Related Works

The NCut technique [1] is used to find a balanced cut in a graph, in order to generate two or more subgraphs. In order to apply this method to the image segmentation issue, a graph that represents the image must be created and its nodes or subgraphs will represent the pixels or image regions, respectively. This balanced cut is calculated by (1), as follows:

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

where $\mathrm{cut}(A,B)$ is defined as the total weight of the edges removed from the original graph V, in order to obtain two subgraphs A and B; $\mathrm{SumCon}(A,V)$ is the total weight of the edges connecting nodes from a subgraph A to all nodes in the original graph V; and $\mathrm{SumCon}(B,V)$ is similarly defined to a subgraph B.

The optimal NCut is the one that minimizes (1), which is a NP-Complete complexity problem. However, by expanding (1), the authors noticed that it can be minimized using spectral graph properties described by Fiedler [7].

There is a wide range of recent work in image segmentation using the NCut technique. Some of them focus in performance improving, while others in different approaches to generate the input graph for this technique. In [1], the weighted graph is built taking each pixel as a node and connecting each pair of pixels by an edge.

Monteiro and Campilho [2] proposed the Watershed Normalized Cut, which uses the Watershed image segmentation to generate a region similarity graph. The weights for this graph are given by a function of intensity and contours from each micro region centroids. The region similarity graph is then used as input for NCut, instead of using the pixel similarity graph that demands a higher computational processing. The Watershed region similarity graph is either used in our previous work [3] in comparison to the primitive input for NCut, and applied in the segmentation of yeast cells images. The weight function used considers the area and the average grayscale level of each region. Ma *et al* [4] used the graph generated by the Watershed Transform to segment texture images.

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, which yields in a better segmentation. 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 linear complexity.

Tao *et al* [6] presented a novel thresholding algorithm that uses the NCut measure. The graph weight matrix is now based on the gray levels of the image, reducing in this way the size of the affinity matrix and consequently requiring a low computational cost. The threshold is done by first building a pixel affinity matrix, followed by using this matrix to build another matrix M, where $M_{(i,j)}=\mathrm{cut}(V_i,V_j)$ and i, j are gray levels. The NCut is then calculated to each threshold value, with all the parameters of the NCut given by the matrix M. If the NCut value belonging to a given gray level t is lower than a fixed one, the optimal threshold value used to separate the objects from the background is t.

III. 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 or pixels. This structure is called a Similarity Graph.

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

A. Pixel Affinity Graph

In this technique, 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 [1], [5], [12].

The intensity and position grouping cue assumes that close-by pixels with similar intensity are most likely to belong to the same object. The measure of similarity regarding this grouping cue is given by (2) [1], [5]:

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

where $\alpha=\big|\big|P_i-P_j\big|\big|$ and $\beta=\big|\big|I_i-I_j\big|\big|$ 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 as 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 intervening contours grouping cue evaluate the affinity between two pixels by measuring the image edges between them. The measure of similarity regarding this grouping cue is given by (3) [5]:

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

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

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

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

1) Multiscale Graph Decomposition: The 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 their spatial separation, as in (5):

$$W = W_1 + W_2 + \ldots + W_s, \quad (5)$$

where W represents the graph weights $\mathrm{w}(i,j)$ 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. The G_r value is a tradeoff between the computation cost and the segmentation result. The decomposition graph above can alleviate this situation. 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.

B. The Watershed Transform

Originated by mathematical morphology, the Watershed Transform [13] treats the gradient image as a topographic surface. The image is flooded from a set of selected sources (also called regional minima) until the whole image has been flooded, with dams buildup between different "lakes" before them meet, generating, in this way, the watershed lines and watershed regions.

Hierarchical Watershed creates a set of nested partitions, i.e., a hierarchy. In this case, a partition at a fine level is obtained by merging regions of the coarse partition [14]. The watershed problem can be modeled using graphs. The flooded gradient image is represented by a full-connected weighted neighborhood graph, where a node represents a catchment basin of the topographic surface. We use Hierarchical Watershed in order to reduce the number of nodes (super segmentation problem) that is originated by the primitive Watershed in the correspondent graph.

After the conversion, a weight function that takes into account the area and average grayscale level of each region is used in the similarity graph to set the edges weights between them.

C. The Component Tree

The Component Tree is a representation of a grayscale image based on the cross-section decomposition (thresholding) between its minimum and maximum gray levels. There exists a relation of inclusion between components at sequential gray levels in the image. A cross-section or threshold is defined as a binary image given by (6) [10], [11]:

$$F_k = \{x \in F / \mathrm{F}(x) \geq k\}, \quad (6)$$

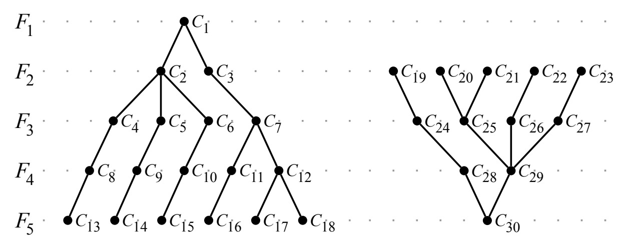
where F is an image and F_k is a section k (level) of F.

The Connected Components (CC) of the different cross-sections may be organized in order to form a tree structure. We say that the two CCs C_{k+1} and C_k are linked when C_{k+1} is a subset of C_k (the inclusion relation). The CC of the first cross-section – F_{min} – corresponds to the whole image domain and it's called root. Fig. 1b shows the CT of the grayscale image depicted in Fig. 1a.

1) Reverse Component Tree: The traditional CT is formed only by the *1's* CCs, once the cross-section used for the root corresponds to the minimal graylevel. However, there is still information on the cross-sections related to the *0's* CCs that are not included in the traditional CT. For some particular cases these CCs hold more relevant information than the *1's* CCs. Therefore, we build a Reverse Component Tree where two CCs C_k and C_{k-1} are linked when C_{k-1} is a subset of C_k. In this case, the root of the tree is formed by the CC of the last cross-section – F_{max}. Fig. 1c shows the RCT of the grayscale image depicted in Fig. 1a.

1	1	4	5	5	1	3	1
1	1	1	4	4	2	5	1
4	5	4	3	1	1	1	1
3	3	1	1	1	1	1	1
1	1	1	1	1	1	5	3
3	4	4	5	1	5	5	3
1	5	2	3	3	5	3	3
1	3	1	1	1	1	1	1

(a)



(b) (c)

Figure 1. (a) A grayscale image (b) the Component Tree of Fig. 1a (c) the Reverse Component Tree of Fig. 1a.

2) Modeling the Similarity Graph: The CT and the RCT are used together to model the similarity graph. However, not all cross-sections need to be used once there is much redundant information across them. The choice of the cross-sections that will be processed can be done by determining the most relevant cross-section and then, selecting the adjacent cross-sections until the sum of CCs of all selected cross-sections is higher or equal than a l value, where l is the maximum number of CCs that could be processed with the available memory. Determining the most relevant cross-section is not trivial. In our model it is given by being close to the middle cross-section and having the higher number of CCs.

A subgraph is created for each selected cross-section, with the nodes set corresponding to its CCs and the edges set obtained from mutual linking the nodes, in order to form a complete graph. After that, the final similarity graph is build by linking the subgraphs with the edges from both the CT and the RCT. The weights for the edges are determined by a combination of the attributes differences between the nodes that it links.

The main goal of this modeling is to keep very similar linked CCs in subsequent cross-sections in the same region. To achieve this goal, the weights of the edges that links the subgraphs needs to be adjusted to increase the similary between them.

IV. 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 [1]. 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 [1], [15].

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 briefly in section III.
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 $X = [X_1, X_2, .., X_K]$ 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.

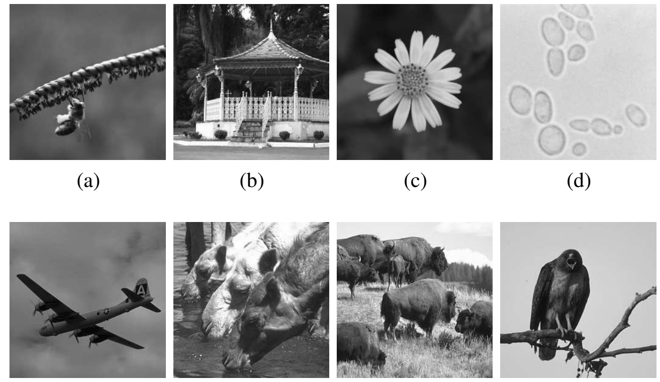
If using the CT / RCT modeling, you must indicate the number of regions and select the partition that corresponds to the best segmentation.

V. Experiments

We use in our experiments a set of 15 randomly chosen images from the Berkeley Image Database [16] and more 10 images from a particular database. The images from Berkeley database needed to be cropped to 256 x 256 pixels. Fig. 2 shows 8 selected images from 25 used in the experiments.

The original implementations of the Pixel Affinity and Multiscale segmentation techniques were provided by the authors [1], [5]. The Hierarchical Watershed Transform was implemented by us on previous works [14].

The experiments were executed according to the steps described in section IV. The connection radius for the Pixel Affinity graph was $r = 10$ and the edges weights were given by (3). In the Watershed Transform, we generated an over segmented image, and then applied the Hierarchical Watershed for limiting the number of regions to aproximately 1200 regions. Then a region adjacency graph was generated from the Hierarchical Watershed regions. For the Multiscale approach were used one radius for each scale, which were $r_1 = 2$, $r_2 = 3$ and $r_3 = 7$. The Component Tree was generated as described in sections III-C and III-C1. Then, a similarity graph was created from it as explained in section III-C2, with $l = 2500$. The attributes used to build our CT similarity graph were difference of area, distance, standard deviation of the gray levels and density. For the edges that links the subgraphs, the weights were multiplied by a factor given by (7):



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2. Original images selected from those used in our experiments: (a-d) from particular database, respectively, bee, coreto, flower and yeast cells; (e-h) from Berkeley database, respectively, 3096, 16077, 38092 and 42049 [16].

$$f = \frac{NC_{Fi} + NC_{Fj}}{2 + |\, \mathrm{d}(i) - \mathrm{d}(j)|}, \quad (7)$$

where NC_{Fi} and NC_{Fj} are the number of CCs of the cross-sections that has the nodes i and j respectively; and $\mathrm{d}(i)$ and $\mathrm{d}(j)$ are the degrees of nodes i and j respectively, related to the CT.

The k-way NCut was configured to generate exactly 30 regions for each experiment. All experiments generated as result one single image partition with 30 regions, except for the experiments using the CT image-graph representation, which generated multiple image partitions, with a variant number of regions on each one. These results with multiple partitions were due to each cross-section represent the entire image area. Therefore, if n cross-sections are selected to compose the similarity graph, then there will be n image partitions as result.

The experimental results for the original images presented in Fig. 2 are shown in Fig. 3, for the particular database images, and Fig. 4, for the Berkeley database images. Note that for the CT results, just one partition of each experiment was chosen to be shown. In this case, the ones with a better segmentation.

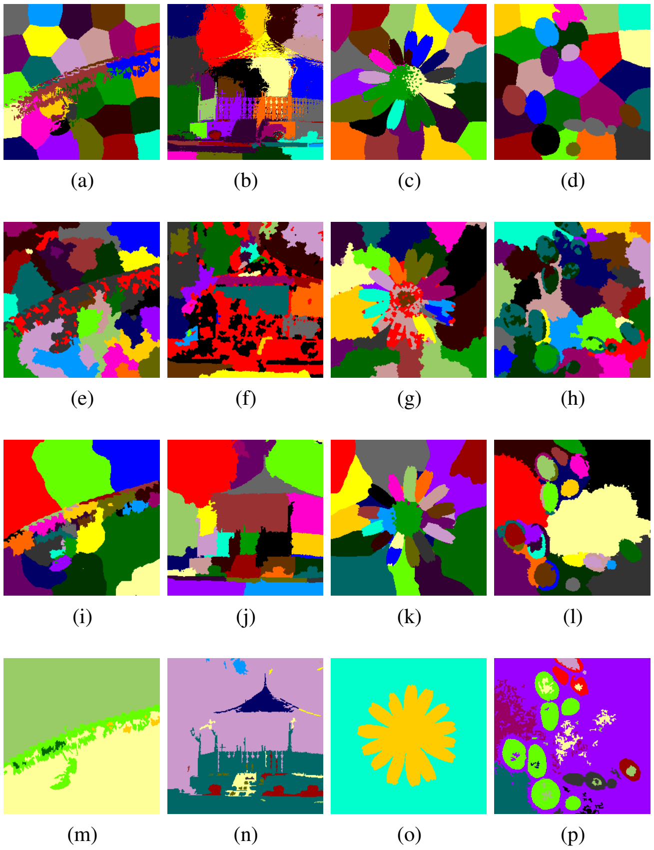


Figure 3. Ncut image segmentation results obtained by different graph representations for the imagens showed in Fig. 2a, Fig. 2b, Fig. 2c and Fig. 2d: (a-d) Pixel Affinity graph (e-h) Hierarquical Watershed region adjacency graph (i-l) Multiscale Graph Decomposition (m-p) Component Tree.

The experiments showed that, in general, regarding a given partition selected by the user, the CT segmentation approach produces better or equivalent results than the other approaches. Only on some cases one of the other segmentation approaches produced a segmentation result that could be considered better than the CT approach results. However, the CT results stayed satisfactory. The results for Fig. 2a and Fig. 2b, presented in Fig. 3, shows such cases.

The CT segmentation result obtained for Fig. 2c, presented in Fig. 3o, separate the entire flower from the background, while the other approaches super segment the subject. This super segmentation behavior presented by the other techniques can also be clearly observed on the segmentation results obtained for Fig. 2e and Fig. 2h. To make these results more similar to the ground truth segmentations, shown respectively in Fig. 4q and Fig. 4t, there will be necessary to perform a region union operation; our approach already presents segmentation results quite similar to the ground truth segmentations, provided by Berkeley database. We have also executed experiments by the other approaches in order to segment the images in the same number of regions obtained by the optimal CT segmentation partitions. However, the results stayed unsatisfactory.

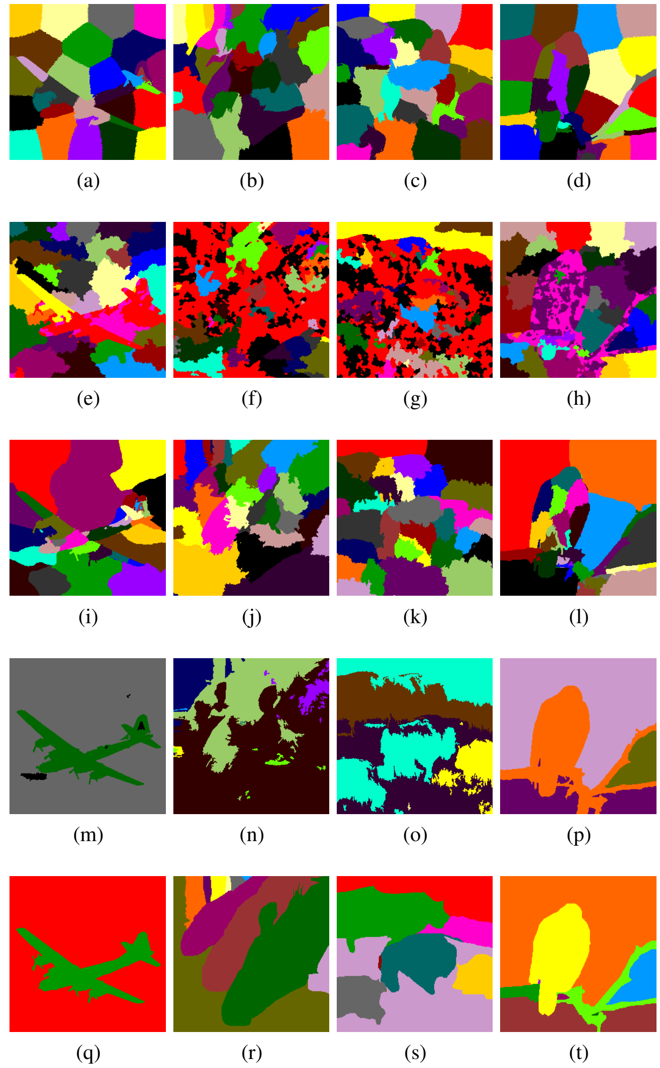


Figure 4. Ncut image segmentation results obtained by different graph representations for the imagens showed in Fig. 2e, Fig. 2f, Fig. 2g and Fig. 2h. (a-d) Results by Pixel Affinity graph (e-h) Results by Hierarquical Watershed region adjacency graph (i-l) Results by Multiscale Graph Decomposition (m-p) Results by Component Tree. (q-t) Ground Truth segmentations from Berkeley database.

The results obtained for Fig. 2d, in particular, presents good segmentations by the Pixel Affinity and the Multiscale approaches. However, they failed on segmenting some yeast cells, while the CT approach segmented all cells, as shown in Fig.3p. In Fig. 3d, can be observed that the three cells at the top of Fig. 2d do not was appropriately segmented by the Pixel Affinity approach. The Multiscale segmentation result, shown in Fig. 3l, presents bad segmentation for cells located at left-bottom of Fig. 2d.

There were some segmentation results that were unsatisfactory for all approaches, like the ones associated to Fig. 2f and Fig. 2g. Nevertheless, the CT result for Fig. 2g, shown in Fig. 4o, is better than the other results, according to the ground truth segmentation, shown in Fig. 4s.

One particularity of our approach is that it results in several different partitions, instead of only one final partition. Therefore, this set of partitions can be analyzed, manually or automatically, in order to achieve the best image segmentation, according to a given application.

A problem in the use of CT is its high computational cost of processing and memory. In this case, it is desirable to use efficient algorithms for the CT construction, particularly because it is also necessary to build up the RCT.

VI. Conclusion

This paper presented an image segmentation method based on image-graph representation and graph cut. The graph partitioning was obtained by means of spectral graph analysis and Normalized Cut.

We proposed the use of Component Tree and Reverse Component Tree as image representation as well as we applied the Normalized Cut in these structures. Experimental results showed that good segmentations are obtained using different ways to represent images by graphs. We showed that the approach that uses the CT and RCT gives the best results. Experiments on real images (particular and Berkeley databases) show that the CT/RCT representation had the advantage of reducing the number of graph nodes.

Our ongoing works aims to explore the CT and RCT image representation in some specific application. Other measures of similarity between the CT/RCT nodes should also be explored in future work, such as contour. Also, other image-graph conversion methods can be used, such as k-means region similarity graph and Quadtree Decomposition. We are especially interested in comparing our results with those obtained by other modeling techniques that use regions as nodes in graphs. The performance of the segmentation method using recursive 2-way or k-way NCut can also be studied.

Acknowledgments

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

References

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and MachineIntelligence*, vol. PAMI-22, no. 8, pp. 888–905, 2000.

[2] F. C. Monteiro and A. Campilho, "Watershed framework to region-based image segmentation," in *Proc. of IEEE 19th International Conference on Pattern Recognition - ICPR*, 2008, pp. 1–4.

[3] M. A. G. Carvalho, A. C. B. Ferreira, T. W. Pinto, and R. M. Cesar-Jr, "Image segmentation using watershed and normalized cuts," in *22th Conference on Graphics, Patterns and Images (SIBGRAPI)*, Rio de Janeiro - Brazil, 2009.

[4] X. Ma and W. Wan, "Texture image segmentation on improved watershed and multiway spectral clustering," in *Proc. of International Conference on Audio, Language and Image Processing - ICALIP*, 2008, pp. 1693–1697.

[5] T. Cour, F. Bénézit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR'05*, vol. 2, 2005, pp. 1124–1131.

[6] W. Tao, H. Jin, Y. Zhang, L. Liu, and D. Wang, "Image thresholding using graph cuts," *IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans*, vol. 38, no. 5, pp. 1181–1195, 2008.

[7] M. A. Fiedler, "Property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory," *Czech Math Journal*, vol. 25, no. 100, pp. 619–633, 1975.

[8] D. Spielman, "Spectral graph theory and its applications," in *Proc. of 48th Annual IEEE Symposium on Foudations of Computer Science*, 2007, pp. 29–38.

[9] D. A. Tolliver and G. L. Miller, "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'06*, vol. 1, 2006, pp. 1053–1060.

[10] M. A. G. Carvalho and T. W. Pinto, "Comparing yeast cells segmentation through hierarchical trees," in *Proc. Of International Conference on Computer Vision Theory and Applications - VISAPP*, vol. 1, Setúbal - Portugal, 2006, pp. 515–518.

[11] V. Mosorov and T. M. Kowalski, "The development of component tree for grayscale image segmentation," in *Proc. of International Conference on Moderns Problems of Radio Engineering, Telecommunications and Computer Science – TECSET*, Slavsko – Ukraine, 2002, pp. 252–253.

[12] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, contours and regions: cue integration in image segmentation," in *Proc. of IEEE International Conference on Computer Vision*, Corfu - Greece, 1999, pp. 918–925.

[13] F. Meyer and P. Maragos, "Multiscale morphological segmentations based on watershed, flooding and eikonal pde," *Scale-Space*, pp. 1–12, 1999.

[14] M. A. G. Carvalho, "Hierarchical image analysis through the tree of critical lakes (in portuguese)," Ph.D. dissertation, Electrical and Computer Engineering Faculty, University of Campinas – UNICAMP, Brazil, 2004.

[15] S. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. of the Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, p. 313.

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "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, July 2001, pp. 416–423.