ISSN: 0976-9102(ONLINE)
DOI: 10.21917/ijivp.2014.0133

ENHANCED GRAPH BASED NORMALIZED CUT METHODS FOR IMAGE SEGMENTATION

S.D. Kapade[1], S.M. Khairnar[2] and B.S. Chaudhari[3]
[1]*Suresh Gyan Vihar University, India*
E-mail: sn_ghorpade@yahoo.com
[2]*Department of Engineering Science, Maharashtra Academy of Engineering, India*
E-mail: smkhairnar2007@gmail.com
[3]*International Institute of Information Technology, India*
E-mail: bharatc@isquareit.edu.in

Abstract

Image segmentation is one of the important steps in digital image processing. Several algorithms are available for segmenting the images, posing many challenges such as precise criteria and efficient computations. Most of the graph based methods used for segmentation depend on local properties of graphs without considering global impressions of image, which ultimately limits segmentation quality. In this paper, we propose an enhanced graph based normalized cut method for extracting global impression and consistencies in the image. We propose a technique to add flexibility to original recursive normalized two way cut method which was further extended to other graph based methods. The results show that the proposed technique improves segmentation quality as well as requires lesser computational time than the regular normalized cut method.

Keywords:

Image Segmentation, Normalized Cut, Pixel Affinity, Multiscale, Watershed Regions

1. INTRODUCTION

The image segmentation is one of the important steps in computer vision for image retrieval, visual summary, and image base modelling. The goal of segmentation is typically to locate certain objects of interest and is useful in medical imaging, face recognition, fingerprint recognition, automated machine vision, and many other applications. Specifically image segmentation progression in medical imaging has helped medical practitioners in more accurately locating cancer cells, in identification of lung diseases, and in automating gene identification.

Most commonly used image segmentation methods are based on clustering, compression, histogram, edge detection, region growing, and partial differential equations, and graphs [1]. Among these methods, the graph based segmentation approach [2] has attracted many attentions and become one of the most successful research areas in computer vision in recent years. In these methods, the set of points in an arbitrary feature space are represented by weighted undirected graph, $G = (V, E)$ where V is the set of nodes called as pixels and an edge set E contains edges formed by joining every pair of nodes. Weight of each edge $w(i, j)$ is function of similarity between nodes V_i and V_j. Partition the set of nodes into disjoint sets $V_1, V_2, V_3, \ldots V_n$ such that the nodes in V_i has strong affinities between them.

Partitioning poses several challenges such as the precise criteria for good partitions and efficient computations. Most of the graph based methods are based on local properties of the graph and many times, this criterion fails to extract global impressions of image. This paper covers exploration of the graph based normalized cut method with focus on extracting global impression of an image than focusing on local features and their consistencies in the image data.

2. NORMALIZED CUT METHOD

Any graph $G = (V, E)$ can be partitioned into two disjoint sets A, B provided that $|V|$ is greater than 1. The degree of dissimilarity between the sets A and B is sum of all the weights of edges between nodes in A to nodes in B called cut value,

$$Cut(A,B) = \sum_{u \in A, v \in B} w(u,v) \tag{1}$$

The optimal bi-partitioning of a graph is the one that minimizes the cut value. By considering every possible partition, minimum cut for a graph can be obtained, but it is very complex problem. Finding minimum cut is well studied problem and there are several efficient algorithms available for solving it. Wu et al. [3] proposed a clustering method based on minimum cut criterion. In this method, graph is partitioned into k – subgraphs such that the maximum cut across the subgraph is minimized. This problem can be also solved by recursively finding the minimum cuts that bisect the existing segments. However, this criterion is suitable for cutting of small sets of isolated nodes in the graph. This is because by using Eq.(1), cut value increases if the numbers of crossings between the two partitioned segments are more. If two partitions are equally sized, they will be related by more edges than the unequally sized partitions. To avoid this unnatural bias for partitioning, Shi et al. proposed a new measure of disassociation, the normalized cut, Ncut [4]. For a graph partition, $G = A \cup B$ the normalized cut cost is

$$Ncut(A,B) = \frac{Cut(A,B)}{assoc(A,V)} + \frac{Cut(A,B)}{assoc(B,V)} \tag{2}$$

where, $Cut(A, B)$ is the sum of weights of edges removed to split the graph and $Assoc(A, V)$ and $Assoc(B, V)$ are the sum of weights of edges in the nodes of A and B, respectively to all nodes in the original graph G. The Ncut of the disassociation between the groups for partitions of small isolated points will be smaller, since there will be a large percentage of the total connections from the small set to all other nodes.

Similarly total normalized association within the groups for given partition is,

$$Nassoc(A,B)=\frac{assoc(A,A)}{assoc(A,V)}+\frac{assoc(B,B)}{assoc(B,V)} \tag{3}$$

where, *assoc*(*A*, *A*) and *assoc*(*B*, *B*) is the sum of weights of edges connecting nodes within *A* and *B* respectively. This measure determines how strongly nodes within the group are connected to each other.

These unbiased measures of association and disassociation of partition are related as,

$$Ncut\ (A, B) = 2\text{-}Nassoc\ (A, B) \tag{4}$$

Hence the partitioning criterion of minimizing the disassociation between the groups and association within the groups can be satisfied simultaneously.

2.1 OPTIMIZATION OF NCUT

Let G be a graph with node set V partitioned into two sets A and B then the minimum Ncut for a graph with N nodes is calculated as below:

i. Let $d(i) = \Sigma w(i, j)$ weight of all the edges connecting node i to all other nodes j.

ii. Let D be diagonal matrix of degrees,

$$D=\begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_N \end{bmatrix}$$

and W be affinity matrix,

$$W=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{p1} & w_{p2} & \cdots & w_{pN} \end{bmatrix}$$

then the minimum Ncut between A and B is given by the relation,

$$\min Ncut(A,B)=\min\frac{y^T(D-W)y}{y^T Dy} \tag{5}$$

where, y is orthogonal to second smallest eigenvectors v_1, v_2, $v_3,\ldots,v_n$ of $\left[\frac{D-W}{D}\right]$ is called as Rayleigh Quotient [5].

iii. If $y \in R$ then Rayleigh Quotient is minimized by solving the generalized Eigen value problem,

$$(D\text{-}W)y = \lambda y \tag{4}$$

The second smallest eigen vector v_2 gives the solution of the normalized cut problem.

To improve the performance of Ncut methods, first we propose a technique to add flexibility to the parameters in original Recursive Ncut method. We have also extended this approach to other graph based methods such as pixel affinity, multiscale decomposition, and watershed regions to improve their performance by constructing the affinity matrix through local tuning.

2.1.1 Recursive Two Way Cut:

The graph nodes are partitioned into two subsets using threshold value. The cut can recursively be obtained in two partitioned parts and stops when it reaches to previously given Ncut value. For given weighted graph G, summarize the information into the affinity matrix W and degree matrix D. Solve $(D\text{-}W)y = \lambda Dy$ for eigen vectors with the smallest eigen value. Use eigen vector corresponding to the second smallest eigen value to bipartition the graph by finding the splitting point so that the Ncut can be minimized. The number of graphs segmented by this method is controlled directly by the maximum allowed Ncut. This technique is known as recursive two way cut [6].

2.1.1.1 Illustration of Recursive Two Way Cut:

a. Construct weighted graph $G = (V, E)$ for given image by considering each pixel as a node and connecting each pair of pixels by an edge. The weight of an edge is similarity between the pair of pixels. Define weight $w_{i,j}$ of an edge connecting to two nodes i and j by using brightness value of the pixel and their spatial location as,

$$w_{i,j}=e^{\frac{-\|F(i)-F(J)\|_2^2}{\sigma F}}\cdot\begin{cases} e^{\frac{-\|X_i-X_i\|_2}{\sigma X}} & if\|X_i-X_i\|_2<r \\ 0 & otherwise \end{cases} \tag{6}$$

where, X_i s the spatial location of node i, $F(i)$ is a feature vector based on intensity, color or texture of node i, σ_F and σ_X are spatial tuning parameters respectively, and w_{ij} is an entry in affinity matrix W.

b. Solve for the eigen vectors with the smallest eigen value of the system

$$(D\text{-}W)\mathrm{y} = \lambda Dy \tag{7}$$

This generalized eigen system can be transformed into standard eigen value problem as

$$D^{-\left(\frac{1}{2}\right)}(D-W)D^{-\left(\frac{1}{2}\right)}y=\lambda y \tag{8}$$

Solving standard eigen value problem for all eigen vectors takes $O(n^3)$ operations, where n is the number of nodes in the graph. Such a large number of operations are impractical for segmentation applications but graph partitioning has property that the graphs are locally connected and the resulting eigen systems are not too dense as well as only the few top eigen vectors are required for partitioning and the precision requirement for the eigen vectors is low. This property of graph partitioning reduces the computations to $O\left(n^{3/2}\right)$.

c. After computing the eigen vectors partition the graph into two pieces using the second smallest eigen vector.

d. Run the algorithm on two partitioned parts or equivalently use top eigen vectors to subdivide the graphs based on those eigen vectors. The recursion stops when Ncut value exceeds certain limits.

We applied the above steps on a sample image as shown in Fig.1(a). The segmented image obtained by using second smallest of the ninth eigen vector is as shown in Fig.1(b).


(a) (b)

Fig.1(a). Original image, (b). Segmented image

2.1.2 Pixel Affinity Graph:

In this method, each pixel is taken as graph node and two pixels within *r* distance are connected by an edge [7]. Similarity between the connected pixels reflects weight of an edge. The measure of similarity, W_{ip} for grouping cue is given by,

$$W_{ip}(i,j)=\begin{cases} e^{-\left[\left(\frac{\alpha^2}{d_p}\right)+\left(\frac{\beta^2}{d_i}\right)\right]}, & if\ \alpha<r \\ 0, & otherwise \end{cases} \tag{9}$$

where, α is position and β is intensity difference between pixels *i* and *j*, *r* is graph connection radius, d_p and d_i are the corresponding scale parameters Another grouping cue, W_{ic} related to the intervening contours is given by,

$$W_{ic}(i,j)=\begin{cases} e^{-\left[\frac{max(x\in line\ f(i,j))}{d_c}\right]^{-\epsilon}} & if\ \alpha<r \\ 0, & otherwise \end{cases} \tag{10}$$

where, line $f(i, j)$ is straight line joining pixels *i* and *j*, and ϵ is square of edge strength at location *x*. These two grouping cue can be combined as,

$$W_{ipc}(i,j)=\sqrt{W_{ip}(i,j)\cdot W_{ic}(i,j)+\alpha W_{ic}(i,j)} \tag{11}$$

where, α is constant.

2.1.3 Multiscale Graph Decomposition:

Decomposition of multiple scales proposed by Benzit *et al.* [8], in which graph links can be separated into different scales,

$$W = W_1+W_2+W_3+\ldots+W_s \tag{12}$$

where, W_s contains affinity between pixels with particular range. $W_s(i, j) \neq 0$ only if $G_{r,\,s\text{-}1} < r_{ij} \leq G_{r,s}$ where, $r_{ij} = ||X_i\text{-}X_j||^2$. For parallel segmentation across scales form the partitioning matrix *X* and multi-scale affinity matrix *W* as below,

$$X=\begin{bmatrix} X_1 \\ \vdots \\ X_s \end{bmatrix},\ W=\begin{bmatrix} W_1^c & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_s^c \end{bmatrix} \tag{13}$$

where, $X_s\in\{0, 1\}^{Ns\times K}$ partitioning of matrix at scale *s*, $X_s(i, k) = 1$ node *i* of I_s is from partition *k*. To find the cross scale interpolation matrix $C_{s,\,s+1}$ between the nodes in layer I_s and nodes in coarser layer I_{s+1} as,

$$C_{s,s+1}(i,j)=\begin{cases} \frac{1}{|N_i|}, & if\ j\in N_i \\ 0, & otherwise \end{cases} \tag{14}$$

The cross scale segmentation constraint matrix *C* is written as,

$$C=\begin{bmatrix} C_{1,2} & -I_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & C_{s,s-1} & -I_s \end{bmatrix} \tag{15}$$

The cross scale segmentation constraint $CX = 0$.

The constrained Normalized Cut is given by,

$$Maximize \in(X)=\frac{1}{k}\sum_{l=1}^{k}\frac{X_l^TWX_l}{X_l^TDX_l} \tag{16}$$

2.1.4 Watershed Regions based Similarity Graph:

Watershed transformation is a morphological based tool for image segmentation and it can be classified as a region-based segmentation approach. In hierarchical watershed method [9], the flooding process starts with given threshold value t_v that represents some relief feature. So, some initial regions will be flooded which yields desired number of partitions, which can be modeled using graph. The flooded gradient image is represented by connected weighted neighborhood graph, where node is the catchment basin of the topographic surface. After conversion, weight function to construct affinity matrix is,

$$w_{iw}(i,j)=e^{-|I_{mi}-I_{mj}|} \tag{17}$$

where, I_m is the density of watershed regions *i* and *j*.

2.2 ENHANCEMENT IN NCUT METHODS

The Ncut algorithm first reads an image of size $n \times n$ and constructs an intensity matrix corresponding to the pixels in an image where intensity matrix consists of feature values or the intensity values of the pixel. Then the graph function computes the affinity matrix of an image by setting default values to the parameters as $\sigma_F = 0.1$, $\sigma_X = 0.3$ and $r = 10$. Parameter σ_F is tuning parameter which controls magnitude of the feature intensity difference involved in computing w_{ij}. From Eq.(6), it can be observed that for smaller values of σ_F, weight w_{ij} is less resulting into closely grouped pixels and more local segmentation and vice versa. The tuning parameter σ_X controls degree of the spatial feature involved in computing w_{ij}. However because of fixed values of σ_F and σ_X in two ways recursive cut method, in many cases the quality of segmentation is compromised. As a result, it achieves global segmentation which is not perceptive to local variations in the image [10]. To achieve improved performance, we correlated the features values around pixel *i* and *j* by modeling σ_F as,

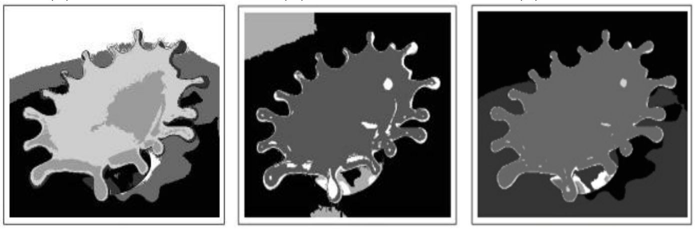
$$\sigma_F = [\sigma(F(i),r).\sigma(F(j),r)]^{-1} \tag{18}$$

where, $\sigma(F(i),r)$ and $\sigma(F(j),r)$ are the standard deviations of neighborhood features around pixel *i* and pixel *j* respectively, around radius *r*. σ_F defined in Eq.(18) will capture the correlation of neighboring features between pixel *i* and pixel *j* while determining the weights of edges. For fixed radius, local variations of features around pixel *i* will be less for smaller values of $\sigma(F(i),r)$, similarly features around *j* will be less for smaller values of $\sigma(F(j),r)$. As well as for low variations in combined local features around pixel *i* and pixel *j*, $\sigma(F(i),r)$. $\sigma(F(j),r)$ will also be smaller and hence improved w_{ij}. This meets to the aim of strong weight connections between the identical neighboring pixels in the affinity matrix *W* resulting better segmentation quality with linear complexity.

3. RESULTS AND DISCUSSION

By using Eq.(18), we varied σ_F {1.0, 0.5, 0.1, 0.05, 0.01, 0.005} and observed segmentation results as shown in Fig.2(a)-Fig.2(f). It illustrates that as σ_F decreases, segmentation becomes more detailed. The algorithm is more sensitive to the value of σ_F and its different values can give sound segmentations in different parts of image.

(a). $\sigma_F = 1.0$ (b). $\sigma_F = 0.5$ (c). $\sigma_F = 0.1$



(d). $\sigma_F = 0.05$ (e). $\sigma_F = 0.01$ (f). $\sigma_F = 0.005$

Fig.2(a) – (f). Segmentation results for different values of σ_F

Similarly for pixel affinity graph, by using Eq.(10), we considered a range of d_c in between 1.0 - 0.001 and r in between 1 and 10. The best segmentation was obtained for d_c = 0.1 and r = 10. For multiscale graph decomposition, Eq.(11) is solved for same range of d_c whereas d_i was varied from 0.1 - 0.005 with α_2 = 1. The best segmentation was achieved at d_c = 0.09 and d_i = 0.005. Watershed region affinity matrix is generated by using connected weighted graph with many regions obtained from hierarchical watershed as input graph. For analysis, we used Berkeley Segmentation Dataset and Benchmark [11] to ease comparison of manual and machine based image segmentation. To compare the results to ground truth boundaries, we need to threshold boundary maps multiple times, at each level it yields two values viz. Precision (P) and Recall (R). Harmonic mean of precision and recall can be summarized in terms of F-measure as 2P.R/(P+R). Precision, Recall, and F-measure as well as time complexity were calculated for each segmented image for all the methods discussed above.

To determine the overall performance of the algorithm, Berkeley's benchmark [12] combines the individual scores from all local segmentations of each image in a single final score. The results shown in Fig.3 demonstrate the final scores obtained by using our approach for Ncut based segmentation methods. It shows that multiscale graph decomposition performs better than other methods. The performance of multiscale graph decomposition is even better than that of combining hierarchical multiscale graph decomposition demonstrated by [13]. For other Ncut based methods, our approach also achieves fairly good performance for most of images considered.

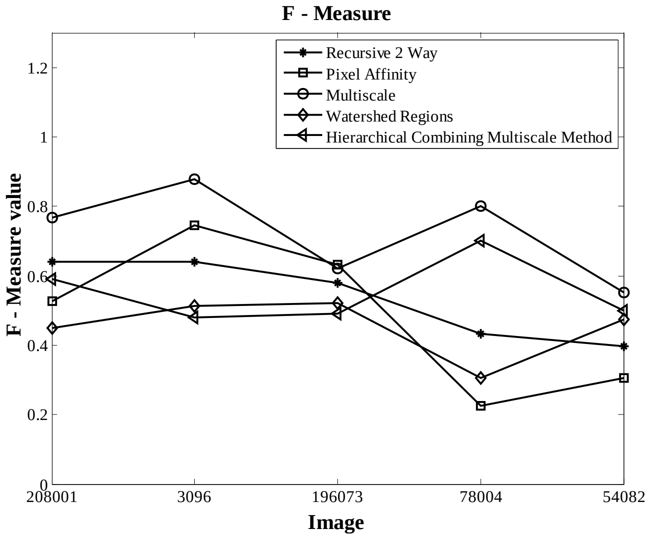


Fig.3. F-Measure for various images (208001, 3096, 196073, 78004, 54082)

The time complexity is an important parameter in Ncut image segmentation methods. We carried out time complexity computations for different images with above discussed methods are as shown in Fig.4.

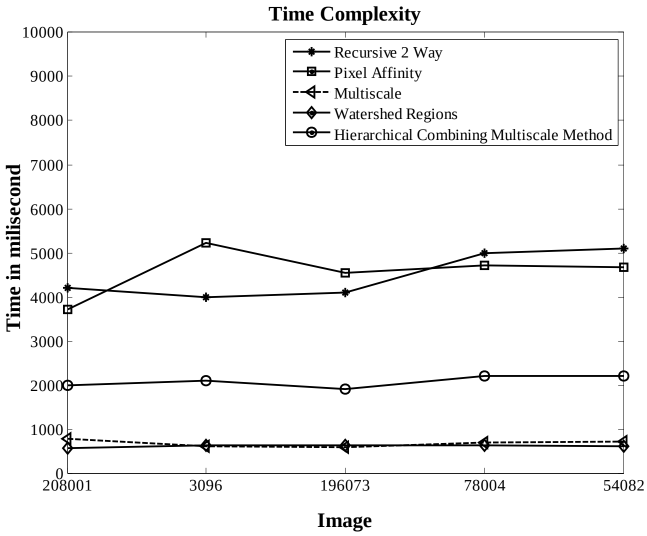


Fig.4. Time Complexity Computations for various images (208001, 3096, 196073, 78004, 54082)

It shows that multiscale and watershed segmentation methods consume less computational power and their performance is almost same for both for all the images considered and it is better than that of computed by [13]. It also indicates that the time complexity for pixel affinity and recursive two way cut methods is sensitive to image.

4. CONCLUSION

The graph based methods generally performs segmentation on the basis of local properties of image. For segmenting the images in some applications where detailed extraction of features is necessary, consideration of global impression along local properties is inevitable. We have proposed an enhanced technique which allows considering both, local as well as global features during normalized cut based segmentation to meet the requirement of precise segmentation. This was achieved by correlating the feature values around neighboring pixels for determining weights of edges of the graph. This creates strong weight connections between the identical neighboring pixels in the affinity matrix resulting better segmentation quality with linear complexity. The result shows that the final score of multiscale graph decomposition is superior to the score obtained for other methods and even better than that of combining hierarchical multiscale graph decomposition. The technique also has lesser computational time complexity.

REFERENCES

[1] J. Tou and R. Gonzalez, "*Pattern Recognition Principles*", Addison-Wesley, 1974.

[2] J. Morris, M. de J. Lee and A.G. Constantinides, "Graph theory for image analysis: an approach based on the shortest spanning tree", *IEE Proceedings Communications, Radar and Signal Processing*, Vol. 133, No. 2, pp. 146-152, 1986.

[3] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, pp. 1101-1113, 1993.

[4] J. Shi and J. Malik, "Normalized cuts and image segmentation", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 731-737, 1997.

[5] G. H. Golub and C. Loan, "*Matrix Computations*", John Hopkins Press, 1989.

[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] D. Martin, C. Fowlkes and J. Malik, "Learning to detect natural image boundaries using local brightness, colour, and texture cues", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, pp. 530-549, 2004.

[8] F. Benezit, T. Cour and J. Shi, "Spectral segmentation with multi-scale graph decomposition", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 1124-1131, 2005.

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

[10] Q. B. Xi, "An Improved Segmentation Algorithm based on Normalized Cut", *Proceedings of the 2nd International Conference on Computer Engineering and Technology*, Vol. 7, pp. 293-296, 2012

[11] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves", *Proceedings of the 23rd International Conference on Machine learning*, pp. 233-240, 2006.

[12] The Berkeley Segmentation Dataset and Benchmark, Available:http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

[13] Marco Antonio Garcia Carvalho and André Luis Costa, "Combining hierarchical structures on graphs and normalized cut for image segmentation", *New Frontiers in Graph Theory, Intech*, pp. 389-406, 2012.