M.Tech. (Computer Science) Dissertation Series

Normalized Cuts and Image Segmentation

a dissertation submitted in partial fulfillment of the requirements for the M.Tech. (Computer Science) degree of the **Indian Statistical Institute**

By

BAHADUR DUTTA
M.Tech-Computer Science
Roll No: CS0815

under the supervision of

Prof. Bhabatosh Chanda
Electronics and Communication Science Unit
ISI, Kolkata

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road,
Kolkata-700108, West Bengal, India.
July 2010

Indian Statistical Institute

Certificate of Approval

This is to certify that the thesis entitled “Normalized Cuts and Image Segmentation" By Bahadur Dutta towards partial fulfillment for the degree of M.Tech. in Computer Science at IndianStatistical Institute, Kolkata, embodies the work done under my supervision.

(Prof. Bhabatosh Chanda)
Date: Electronic and Communication Science Unit
Indian Statistical Institute, Kolkata.

ACKNOWLEDGMENT

I take this opportunity to thank **Prof. Bhabatosh Chanda**, Electronics and Communication Sciences Unit, ISI-Kolkata for his valuable guidance and inspiration. His pleasant and encouraging words have always kept my spirits up.

I would like to thank all of my class mates, friends, Pulak Purkait and Rajesh Kumar for their support and motivation to complete this project.

Date:

BAHADUR DUTTA
M.Tech.(CS)
Indian Statistical Institute, Kolkata

Abstract

We propose a novel approach for solving the perceptual grouping problem in vision. Rather than focusing on local features and their consistencies in the image data, our approach aims at extracting the global impression of an image. We treat image segmentation as a graph partitioning problem and propose a novel global criterion, the normalized cut, for segmenting the graph. The normalized cut criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups. We show that an efficient computational technique based on a generalized eigen value problem can be used to optimize this criterion. We have applied this approach to segmenting static images, as well as motion sequences, and found the results to be very encouraging.

Contents

Chapter 1

Introduction

Since there are many possible partitions of the domain I of an image into subsets, how do we pick the right one?.There are two aspects to be considered here. The first is that there may not be a single correct answer. A Bayesian view is appropriate-there are several possible interpretations in the context of prior world knowledge. The difficulty, of course, is in specifying the prior world knowledge. Some of it is low level, such as coherence of brightness, color, texture, or motion, but equally important is mid- or high level knowledge about symmetries of objects or object models. The second aspect is that the partitioning is inherently hierarchical. Therefore, it is more appropriate to think of returning a tree structure corresponding to a hierarchical partition instead of a single “flat” partition.

This suggests that image segmentation based on low level cues cannot and should not aim to produce a complete final “correct” segmentation. The objective should instead be to use the low-level coherence of brightness, color, texture, or motion attributes to sequentially come up with hierarchical partitions. Mid- and high-level knowledge can be used to either confirm these groups or select some for further attention. This attention could result in further repartitioning or grouping. The key point is that image partitioning is to be done from the big picture downward, rather like a painter first marking out the major areas and then filling in the details.

1.1 **Equivalent to Graph Theoretic Problem**.

Our approach is most related to the graph theoretic formulation of grouping. The set of points in an arbitrary feature space are represented as a weighted undirected graph G =(V,E) where the nodes of the graph are the points in the feature space, and an edge is formed between every pair of nodes. The weight on each edge, w(i,j) is a function of the similarity between nodes i and j. In grouping, we seek to partition the set of vertices into disjoint sets V1,V2, . . . ,Vm. where by some measure the similarity among the vertices in a set Vi is high and, across different sets Vi, Vj is low.

1.2 How to Partition The Graph.

To partition a graph, we need to also ask the following questions:

1. What is the precise criterion for a good partition?
2. How can such a partition be computed efficiently?

In the image segmentation and data clustering community, there has been much previous work using variations of the minimal spanning tree or limited neighborhood set approaches. Although those use efficient computational methods, the segmentation criteria used in most of them are based on local properties of the graph. Because perceptual grouping is about extracting the global impressions of a scene, as we saw earlier, this partitioning criterion often falls short of this main goal
.

1.3 Brief review of the work.

we propose a new graph-theoretic criterion for measuring the goodness of an image partition-the normalized cut. We introduce and justify this criterioninSection2.The minimization of this criterion can be formulated as a generalized eigen value problem. The eigenvectors can be used to construct good partitions of the image and the process can be continued recursively as desired (Section 2.1). Section 3 gives a detailed explanation of the steps of our grouping algorithm. In Section 4, we show experimental results. The formulation and minimization of the normalized cut criterion draws on a body of results from the field of graph theory (Section 5). We conclude in section 6.

Chapter 2

Grouping in Graph Partition

2.1 Cut Of The Graph.

A graph G =(V,E).can be partitioned into two disjoint sets, A,B: $A \cup B = V$, $A \cap B = \emptyset$. by simply removing edges connecting the two parts. The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed. In graph theoretic language it is called Cut:

$$cut(A,B) = \sum_{u \in A, v \in B} W(u,v) \quad \text{.................................(1)}$$

2.2 Optimal Bipartition.

The optimal bi-partitioning of a graph is the one that minimizes this cut value. Although there are an exponential number of such partitions, finding the minimum cut of a graph is a well-studied problem and there exist efficient algorithms for solving it

2.3 Wu and Leahy Method.

Wu and Leahy proposed a clustering method based on this minimum cut criterion. In particular, they seek to partition a graph into k-sub graphs such that the maximum cut across the subgroups is minimized. This problem can be efficiently solved by recursively finding the minimum cuts that bisect the existing segments. As shown in Wu and Leahy's work, this globally optimal criterion can be used to produce good segmentation on some of the images.

However, as Wu and Leahy also noticed in their work, the minimum cut criteria favors cutting small sets of isolated nodes in the graph. This is not surprising since the cut defined in (1) increases with the number of edges going across the two partitioned parts. Fig. 1 illustrates one such case. Assuming the edge weights are inversely proportional to the distance between the two nodes, we see the cut that partitions out node n1 or n2 will have a very small value. In fact, any cut that attritions out individual nodes on the right half will have smaller cut value than the cut that partitions the nodes into the left and right halves.

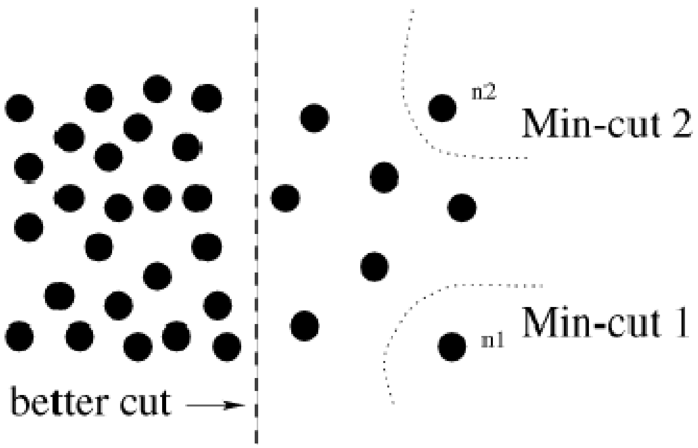

Fig. 1. A case where minimum cut gives a bad partition.

2.4 Normalized Cut:

To avoid this unnatural bias for partitioning out small sets of points, we propose a new measure of disassociation between two groups. Instead of looking at the value of total edge weight connecting the two partitions, our measure computes the cut cost as a fraction of the total edge connections to all the nodes in the graph. We call this disassociation measure the normalized cut (Ncut):

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \quad \text{...(2)}$$

$$assoc(A,V) = \sum_{u \in A, t \in V} W(u,t).$$

Where assoc(A,V) is the total connection from nodes in A to all nodes in the graph and assoc(B, V) is similarly defined. With this definition of the disassociation between the groups, the cut that partitions out small isolated points will no longer have small Ncut value, since the cut value will almost certainly be a large percentage of the total connection from that small set to all other nodes. In the case illustrated in Fig. 1, we see that the cut1 value across node n1 will be 100 percent of the total connection from the node.

we can define a measure for total normalized association within groups for a given partition.

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

where assoc(A,A) and assoc(B,B)are total weights of edges connecting nodes within A and B, respectively. We see again this is an unbiased measure, which reflects how tightly on average nodes within the group are connected to each other.

Another important property of this definition of association and disassociation of a partition is that they are naturally related.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$= \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)}$$

$$= \left(2 - \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}\right)\right)$$

$$= (2 - Nassoc(A,B))$$

Hence, the two partition criteria that we seek in our grouping algorithm, minimizing the disassociation between the groups and maximizing the association within the groups, are in fact identical and can be satisfied simultaneously. In our algorithm, we will use this normalized cut as the partition criterion.

2.5 Computing the Optimal Partition

Given a partition of nodes of a graph, V, into two sets A and B, let x be an N .=total number of node of the graph dimensional indicator vector, xi = 1 if node i is in A and -1, otherwise. Let . d(i) be the total connection from node i to all other nodes. With the definitions x and d, we can rewrite Ncut (A,B) as:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$= \frac{\sum_{x_i>0,x_j<0} -W_{i,j}x_ix_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0,x_j>0} -W_{i,j}x_ix_j}{\sum_{x_i<0} d_i}$$

Where $d(i) = \sum_j W_{i,j}$

Let D be an N X N diagonal matrix with d on its diagonal be an N XN symmetrical matrix with $W_{(i,j)} = w_{i,j}$

$$k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i}$$

and 1 be an N X 1 vector of all ones. Using the fact $\frac{1+x}{2}$ and $\frac{1-x}{2}$ are indicator vectors for $x_i > 0$ and $x_i < 0$, respectively, we can rewrite 4[Ncut(x)] as:

$$= \frac{(1+x)^T(D-W)(1+x)}{k1^TD1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k)1^TD1}$$

$$= \frac{x^T(D-W)x + 1^T(D-W)1}{k(1-k)1^TD1} + \frac{2(1-2k)1^T(D-W)x}{k(1-k)1^TD1}$$

Let

$$\alpha(x) = x^T(D - W)x$$

$$\beta(x) = 1^T(D - W)x$$

$$\gamma(x) = 1^T(D - W)1$$

And M = $1^T D1$,we can then further expand the above the above equation as :

$$= \frac{(\alpha(x)+\gamma)+2(1-2k)\beta(x)}{k(1-k)M}$$

$$= \frac{(\alpha(x)+\gamma)+2(1-2k)\beta(x)}{k(1-k)M} - \frac{2(\alpha(x)+\gamma)}{M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$$

Dropping the last constant term ,which in this case equal to 0,we get

$$= \frac{(1-2k+2k^2)(\alpha(x)+\gamma)+2(1-2k)\beta(x)}{k(1-k)M} + \frac{2\alpha(x)}{M}$$

$$= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(x)+\gamma)+\frac{2(1-2k)}{(1-k)^2}\beta(x)}{\frac{k(1-k)}{(1-k)^2}M} + \frac{2\alpha(x)}{M}$$

Letting b = $\frac{k}{1-k}$ and since $\gamma = 0$ it becomes,

$$= \frac{\left(1+b^2\right)(\alpha(x)+\gamma)+2\left(1-b^2\right)\beta(x)}{bM} + \frac{2b\alpha(x)}{bM}$$

$$= \frac{\left(1+b^2\right)(\alpha(x)+\gamma)}{bM} + \frac{2\left(1-b^2\right)\beta(x)}{bM} + \frac{2b\alpha(x)}{bM} - \frac{2b\gamma}{bM}$$

$$= \frac{\left(1+b^2\right)(x^T(D-W)x+1^T(D-W)1}{b1^TD1} + \frac{2\left(1-b^2\right)1^T(D-W)x}{b1^TD1} + \frac{2bx^T(D-W)x}{b1^TD1} - \frac{2b1^T(D-W)1}{b1^TD1}$$

$$= \frac{(1+x)^T(D-W)(1+x)}{b1^TD1} + \frac{b^2(1-x)^T(D-W)(1-x)}{b1^TD1} - \frac{2b(1-x)^T(D-W)(1+x)}{b1^TD1}$$

$$= \frac{[(1+x)-b(1-x)^T](D-W)[(1+x)-b(1-x)]}{b1^TD1}$$

Setting y =(1=x)-b(1-x) it is easy to see that

$$y^TD1 = \sum_{x_i>0} d_i - b\sum_{x_i<0} d_i = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Since b = $\frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$ and

$$y^TDy = \sum_{x_i>0} d_i + b^2\sum_{x_i<0} d_i$$

$$= b\sum_{x_i>0} d_i + b^2\sum_{x_i<0} d_i$$

$$= \mathrm{b}\left(\sum_{x_i<0} d_i + b\sum_{x_i<0} d_i\right)$$

$$= \mathrm{b}1^TD1$$

Putting everything together we have

$$min_x \text{Ncut}(x) = min_y \frac{y^T(D-W)y}{y^T D y} \quad \text{...(5)}$$

With the condition y(i) $\in \{1, b\}$ $y^T D1 = 0$.

Note that the above expression is the Rayleigh Quotient. If y is relaxed to take on real values, We can minimize (5) by solving the generalized eigen value system,

$$(D\text{-}W)y = \lambda\, Dy \text{...(6)}$$

However, we have two constraints on y which come from the condition on the corresponding indicator vector x. First consider the constraints y^TD1=0.We can show this constraints on y automatically satisfied by the solution of the generalized eigensystem. We will do first transforming (6) into a standard eigen system and showing the corresponding is satisfied there. Rewrite (6) as

$$D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}z = \lambda z \text{ ....(7)}$$

Where z=$D^{\frac{1}{2}}y$. One can easily verify that z_0=$D^{\frac{1}{2}}$ 1 is an eigen vector of (7) with eigen value of 0. Furthermore , $D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}$ is symmetric positive semidefinite matrix. since (D-W) ,also called the Laplacian matrix is known to be positive semidefinite. Hence z_0 is the smallest eigen vector of (7) and all eigen vector are perpendicular to each other. In particular z_1 is the second smallest eigen vector is perpendicular to z_0 . Hence y_0=1 is the smallest eigen vector with eigen value 0 in (6).

Theorem:

Let A be a real symmetric matrix. Under the constraints that x is orthogonal to the j-1 smallest eigen vector $x_1\ x_2\ x_3\ .....x_{j-1}$ the quotient $\frac{x^T Ax}{x^T x}$ is minimized by the second smallest eigen vector x_j and its minimum value is the corresponding eigen value λ_j.

Chapter 3

3. Grouping Algorithm

3.1 Algorithm

Our grouping algorithm consists of the following steps.

1. Given image I we partition the image into some number of region by watershed algorithm. After that for each region we are taking mean value of that region and each mean value is node of the constructed weighted graph.

2. The weighted graph G =(V,E) and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.

3. Solve $(D-W)x = \lambda Dx$ for eigenvectors with the smallest eigen values.

4. Use the eigenvector with the second smallest eigen value to bipartition the graph.

5. Decide if the current partition should be subdivided and recursively repartition the segmented parts if necessary.

3.2 Example Of Brightness Image

1. Construct a weighted graph G =(V,E) by taking each pixel as a node and connecting each pair of pixels by an edge. The weight on that edge should reflect the likelihood that the two pixels belong to one object. Using just the brightness value of the pixels and their spatial location, we can define the graph edge weight connecting the two nodes i and j as:

$$W_{i,j} = e^{\frac{\|F_i - F_j\|^2}{\sigma_1{}^2}} * \begin{cases} e^{\frac{\|x_i - x_j\|^2}{\sigma_x{}^2}} & if \quad \|x_i - x_j\|_2 < r \\ 0 & \text{otherwise} \end{cases}$$

2. Solve for the eigenvectors with the smallest eigen values of the system.

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}x = \lambda x$$

3. Once the eigenvectors are computed, we can partition the graph into two pieces using the second smallest eigenvector. In the ideal case, the eigenvector should only take on two discrete values and the signs of the values can tell us exactly how to partition the graph. However, our eigenvectors can take on continuous values and we need to choose a splitting point to partition it into two parts.

4. After the graph is broken into two pieces, we can recursively run our algorithm on the two partitioned parts. Or, equivalently, we could take advantage of the special properties of the other top eigenvectors as explained in the previous section to subdivide the graph based on those eigenvectors. The recursion stops once the Ncut value exceeds certain limit.

.

Chapter 4

Experiment

4.1 How To Construct The Graph.

We have applied our grouping algorithm to image segmentation based on brightness, color, texture, or motion information. In the monocular case, we construct the graph G=(V,E) by taking the mean of each region as a node and define the edge weight $W_{(i,j)}$ between node i and j as the product of a feature similarity term and spatial proximity term

:

$$W_{i,j} = e^{\frac{\|F_i - F_j\|^2}{\sigma_1{}^2}} * \begin{cases} e^{\frac{\|x_i - x_j\|^2}{\sigma_x{}^2}} & if \quad \|x_i - x_j\|_2 < r \\ 0 & \text{otherwise.} \end{cases}$$

where X(i) is the spatial location of node i, and F(i) is a feature vector based on intensity, color, or texture information at that node defined as:

1. F(i)= 1, in the case of segmenting point sets,
2. F(i)= I(i), the intensity value, for segmenting brightness images,
3. F(i)= [v, v. s. sin(h), v. s. cos(h)](i) where h, s, v are the HSV values, for color segmentation,

. Note that the weight $W_{(i,j)}$ = 0 for any pair of nodes i and j that are more than r pixels apart

4.2 Computation Time

The running time of the normalized cut algorithm is O(m,n) where n is the total number of region segmented by watershed algorithm, and m is the number of steps Lanczos takes to converge

.

Chapter:5

Result 1:

Fig: 5.1.(a)

Fig 5.1.(b)

Fig 5.1.(c)

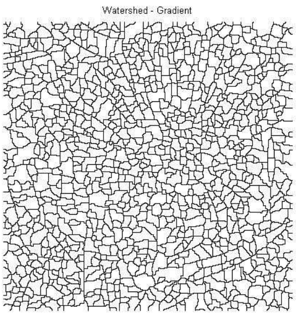

Fig 5.1.(w)

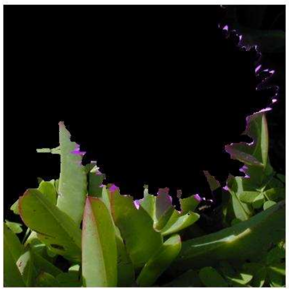

Fig 5.1.(d)

Fig 5.1.(e)

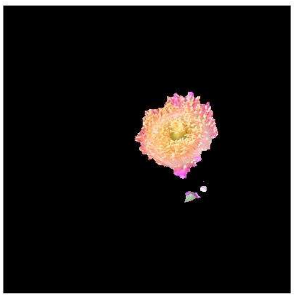

Fig 5.1.(f)

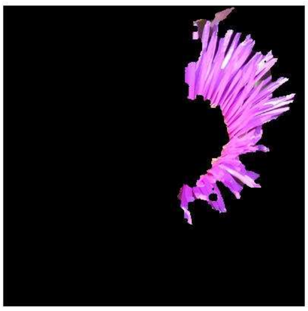

Fig 5.1.(g)

5.1.(a) Input Image.5.1.(b) to 5.1.(g) are the output image .Here $\sigma_i = 50, \sigma_x = 40$,r=50.
Fig 5.1.(w) is the image segmentation after watershed algorithm.

Result 2:

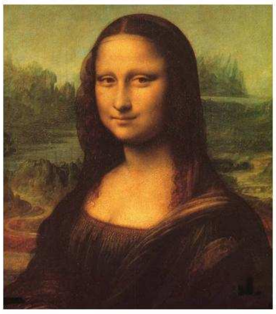

Fig 5.2.(a)

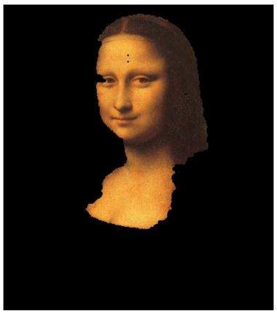

Fig 5.2.(b)

Fig 5.2.(c)

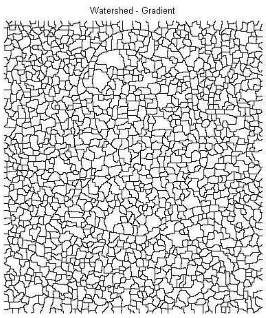

Fig 5.2.(w)

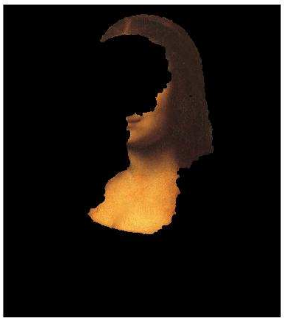

Fig 5.2.(d)

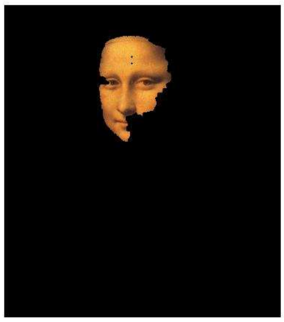

Fig 5.2.(e)

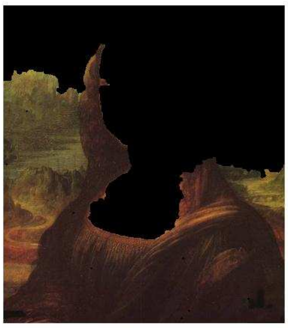

Fig 5.2.(f)

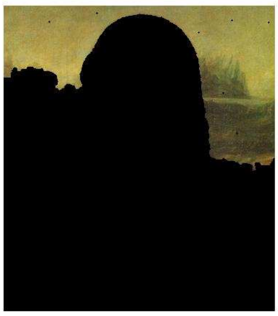

Fig 5.2.(g)

Fig 5.2.(a) Input image. Fig 5.2.(b) to Fig 5.2.(g) are output image. Here $\sigma_i = 23, \sigma_x = 8, r = 75$. Fig 5.2.(w) is the image segmentation after watershed algorithm.

Chapter 6.

Conclusion

We developed a grouping algorithm based on the view that perceptual grouping should be a process that aims to extract global impressions of a scene and provides a hierarchical description of it. By treating the grouping problem as a graph partitioning problem, we proposed the normalized cut criteria for segmenting the graph. Normalized cut is an unbiased measure of disassociation between subgroups of a graph and it has the nice property that minimizing normalized cut leads directly to maximizing the normalized association, which is an unbiased measure for total association within the subgroups. In finding an efficient algorithm for computing the minimum normalized cut, we showed that a generalized eigenvalue system provides a real valued solution to our problem. A computational method based on this idea has been developed and applied to segmentation of brightness, color, and texture images. Results of experiments on real and synthetic images are very encouraging and illustrate that the normalized cut criterion does indeed satisfy our initial goal of extracting the “big picture” of a scene.

Refferences

[1] N. Alon, “Eigenvalues and Expanders,” Combinatorica, vol. 6, no. 2,1986

[2] A. Blake and A. Zisserman, Visual Reconstruction. MIT Press, 1987

[3] R.B. Boppana,”Eigenvalues and Graph Bisection:An Average-Case Analysis,” Proc. 28th Symp. Foundations of Computer Science.1987

[4] J. Cheeger, “A Lower Bound for the Smallest Eigenvalue of the Laplacian,” Problems in Analysis, R.C. Gunning,1970.

[5] F.R.K. Chung, Spectral Graph Theory.,Am.Math.Soc,1997

[6] I.J. Cox, S.B. Rao, and Y. Zhong, “Ratio Regions: A Technique for Image Segmentation” 1996.

[7] W.E. Donath and A.J. Hoffman, “Lower Bounds for the Partitioning of Graphs,”1973.

[8] R. Van Driessche and D. Roose, “An Improved Spectral Bisection Algorithm and Its Application to Dynamic Load Balancing,”1995

[9] M. Fiedler, “A Property of Eigenvectors of Nonnegative Symmetric Matrices and Its Applications to Graph Theory,”1990

[10]. G.H. Golub and C.F. Van Loan, Matrix Computations. John Hopkins Press, 1989.

[11] J.Shi and J.Malik,”Normalized Cuts and ImageSegmentation,”Proc.IEEEConf. Computer Vision and Pattern Recognition.1997.

[12] J.Shi and J Malik,”Motion Segmentation and tracking using Normalized Cuts”, Computer Vision.1998.

[13] Z.Wu and R.Leahy,”An optimal Graph Theoretic Approach to Data Clustering: Theroy and its application to Image Segmentation”IEEE Pattern Analysis and Machine Intelligence.1993.