NOVEL GRAPH BASED METHOD FOR IMAGE SEGMENTATION

[1] Dr. S. V.KASMIR RAJA, [2] A. SHAIK ABDUL KHADIR, [3] Dr. S. S. RIAZ AHAMED

[1] Dean (Research), SRM University, Chennai, TamilNadu, India
[2] Lecturer (SG), Dept of Computer Science, Khadir Mohideen College, Adirampattinam-614701, TamilNadu, India-614701
[3] Principal, Sathak Institute of Technology, Ramanathapuram,TamilNadu, India-623501.
Email: ssriaz@ieee.org , ssriaz@yahoo.com

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. At the heart of unsupervised clustering and semi-supervised clustering is the calculation of matrix Eigen values (eigenvectors) or matrix inversion. In generally, its complexity is $O(N^3)$. By using Fast Lanczos Method in Normalized cut Method, we improve the performance to O(N log N). We have applied this approach to segmenting static images, as well as motion sequences, and found the results to be very encouraging.

Keywords : *Grouping, image segmentation, graph partitioning., unsupervised clustering*

1. INTRODUCTION

Nearly 75 years ago, Wertheimer [1] pointed out the importance of perceptual grouping and organization in vision and listed several key factors, such as similarity, proximity, and good continuation, which lead to visual grouping. However, even to this day, many of the computational issues of perceptual grouping have remained unresolved. In this paper, we present a general framework for this problem, focusing specifically on the case of image segmentation.

Prior literature on the related problems of clustering, grouping and image segmentation is huge. The clustering community [3] has offered

computation of large matrix. Solving systems of linear equations Ax = b and computing eigenvalues and eigenvectors of large matrices $Ax=\lambda x$ are two fundamental computation for clustering problem. They also have great practical uses in other machine learning problems. For example, for semi-supervised clustering, to label unlabeled points

us agglomerative and divisive algorithms; in image segmentation, we have region-based merge and split algorithms. The hierarchical divisive approach that we advocate produces a tree, the dendrogram. While most of these ideas go back to the 1970s (and earlier), the 1980s brought in the use of Markov Random Fields [2] and variational formulations [6], [4], [5].

Data Clustering and graph segmentation are important operations for machine learning and computer vision. It includes both unsupervised and semi-supervised clustering. For unsupervised clustering, spectral method[8][9][10][7] has been the focus of considerable research. However, all these methods suffer from the slow can be transformed into a problem of solving linear systems. And for the spectral clustering problem, it turns out to be a problem of computing the largest k eigenvectors of the weighted matrix W. Hence how to solve these two problems correctly and fast becomes very important.

The rest of the paper is organized as follows. Section 2 describes various methods. Detailed steps discussed in Section 3. In Section 4,we explained experiment conducted with results . Lastly ,we have given conclusion .

2. GROUPIMG AS GRAPH PARTITIONING

Wu and Leahy [11] proposed a clustering method based on this minimum cut criterion. In particular, they seek to partition a graph into k-subgraphs 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.

2.1 Iterative methods

Iterative methods are dominant in computing large matrices because direct methods are either impossible or too slow hence infeasible in practice. First, there is no direct method for eigenvalue problems when dimension of the matrix is greater than 5. Any eigenvalue solvers must be iterative. On the other hand, direct methods for solving linear systems like Gaussian elimination require $O(N^3)$ operations, which is too time-consuming. Iterative methods are approximated methods, which only require $O(N^2)$ operations. They can compute solutions much faster with errors which can be tolerant. In practice, this is often good enough. The eigenvectors can be used to construct good partitions of the image and the process can be continued recursively.

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 $V_1, V_2 \ldots V_m$, where by some measure the similarity among the vertices in a set V_i is high and, across different sets V_i, V_j is low.

Given a matrix A and a vector b, the associated Lanczos sequence is the set of vectors: b, Ab, A^2b, A^3b . . . The corresponding Lanczos subspaces are the spaces spanned by successively larger groups of these vectors in the Lanczos sequence.

2.2 Fast Gauss method

The fast Gauss transform introduced by Greengard and Strain is an important variant of more general fast multipole methods. It's for the Sum-Product problem within O(NlogN) operations, while the direct method requires $O(N^2)$ operations. The multiplication of some particular form matrix (Standard Gaussian Kernel Matrix) and a positive vector can be translated to the Sum-Product problem. For the standard Gaussian kernel matrix A,

$$Aij=\exp(-\|xi-yj\|^2/\sigma^2)$$

Therefore, the i^{th} element of A * w is the sum-product of the i^{th} row of A and the weight vector w.

The average running time for the fast method is O(NlogN) , in the worst case it's $O(N^2)$ and O(N) for the best case. During the computation, the matrix A never need to be constructed explicitly, hence space required can be reduced from $O(N^2)$ to O(N).

2.3 Fast Krylov methods

Krylov subspace methods provide an iterative algorithm running at O(N2) due to direct multiplication of a matrix and a vector. Fast Gauss method allows this multiplication to reduced to O(NlogN) for standard Gaussian kernel matrices. Hence it is naturally to combine these two methods to solving linear systems and computing eigen problems for this particular type of matrices. As a result, we could reduce running time from $O(N^3)$ to O(NlogN). Because this type of matrices occur a lot in machine learning problems, it will be useful if we can combine these two algorithms and still solve original problems correctly. We already have a proof on their running time.

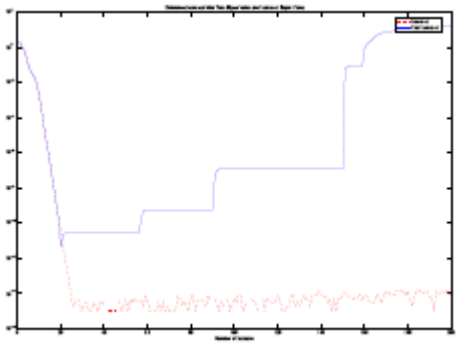
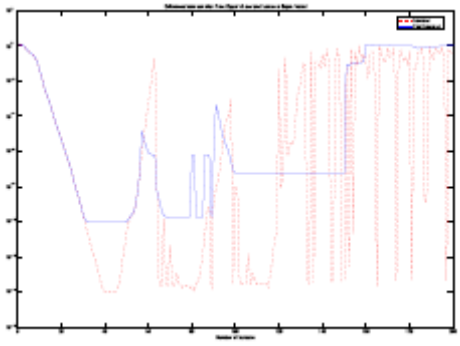


Fig 1 Eigenvalue of Lanczos



Fg2 First Eigenvector of Lanczos

From Figure1 and Figure2, we see for the fast lanczos and lanczos algorithm, they are almost the same at the first 20 iterations and differences between partial solutions to true solution drop below 10−10. After that, Fast Lanczos stays for a while and goes up and naive Lanczos's eigenvalue stays but eigenvector becomes oscillating. This shows Lanczos should be stopped around 20 iterations since we already obtain fairly good solutions. More iteration will cause solutions unstable

Since nave Lanczos is not stable, we can say this is mainly due to Lanczos, not due to error of fast Gauss method.

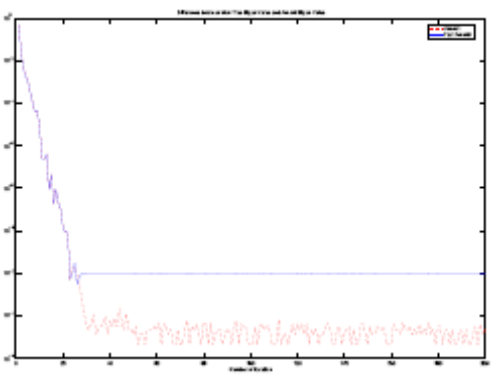


Fig3 Eigen value of Arnoldi

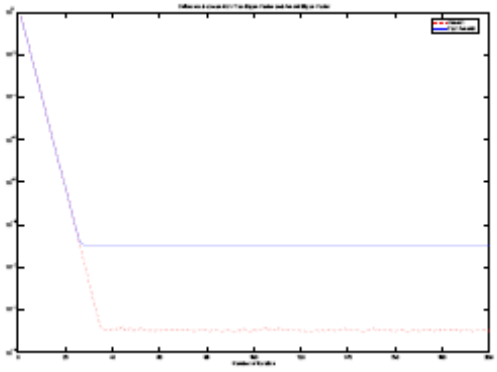


Fig 4 First Eigen vector of Arnoldi

For Arnoldi algorithms (See Figure 3 and Figure4), both methods are almost same in the first 20 iterations and differences between partial solutions and true solution drop below 10−10. After that, Fast Arnoldi stays around 10−11 and nave Arnoldi stays around 10−14. This shows Arnoldi is very stable. Fast Arnoldi cannot drop further because the specified error eps of fast Gauss method is 10−10. We believe if we allow eps to be smaller, then Fast Arnoldi should be able to get a better solution. In practice, error rate of 10−11 is already good enough.

3. EFFICIENT NCUT SEGMENTATION ALGORITHM

The summary of proposed new Ncut Algorithm is given below

1. Set up problem as G = (*V*, *E*) and define affinity matrix **W** and degree matrix **D**

2. Solve (**D** – **W**)**y** = λ**Dy** for the eigenvectors with the smallest eigenvalues

3. Find the 2nd smallest Eigen Vector and eigen value with O(n log n) time complexity using Fast combined approach

4. Let **x**2 = eigenvector with the 2nd smallest eigenvalue □2
5. Threshold **x**2 to obtain the binary-valued vector **x**´2 such that *ncut*(**x**´2) □ *ncut*(**x***t*2) for all possible thresholds *t*
6. For each of the two new regions, if *ncut* < threshold T, then recursive on the region.

The detailed steps are:

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:

$$W_{ij} = \exp(-\|F(i)-F(j)\|^2 / \sigma^2 I) * \begin{cases} \exp(-\|X(i)-X(j)\|^2 / \sigma^2 X) & \text{if } \|X(i)-X(j)\|_2 < r \quad \ldots\ldots (A) \\ 0 & \text{otherwise} \end{cases}$$

2. Solve for the eigenvectors with the smallest eigenvalues of the system

$$(D - W)y = \lambda Dy \quad \ldots. \quad (B)$$

As we saw above, the generalized eigensystem in (b) can be transformed into a standard eigenvalue problem of

$$D^{-1/2} (D - W) D^{-1/2} x = \lambda x \quad \text{---} \quad (C)$$

Solving a standard eigenvalue problem for all eigenvectors takes $O(n^3)$ operations, where n is the number of nodes in the graph. This becomes impractical for image segmentation applications where n is the number of pixels in an image.

Fortunately, our graph partitioning has the following properties: 1) The graphs are often only locally connected and the resulting eigensystems are very sparse, 2) only the top few eigenvectors are needed for graph partitioning, and 3) the precision requirement for the eigenvectors is low, often only the right sign bit is required. These special properties of our problem can be fully exploited by an eigensolver called the Lanczos method. The running time of a Lanczos algorithm is O(mn) + O(mM(n)), where m is the maximum number of matrix-vector computations required and M(n) is the cost of a matrix-vector.

,

3. For each iteration, the most time- consuming step is to calculate y = Lx. However, $L = D^{-1/2}W D^{-1/2}$ is not a standard Gaussian kernel matrix, we can't use fast method directly. If we use the fast combined approach, We can modify the above mentioned equation as follows

• Rewrite y = Lx as $y^1 = Wx^1$ where $y = D^{-1/2} y^1$ and $x^1 = D^{-1/2} x$. First, we calculate x^1 using O(N) time because D is a diagonal matrix; then we use fast method for y^1 using O(Nlog(N)) time, at last we get y.

• However, here W is not a standard gaussian kernel matrix. The diagonal value is 0 rather than 1. We define $W = W^1 - E$, then W is standard Gaussian kernel matrix. We get $y = Wx^1 = (W^1 - E)x^1 = W^1x^1 - x^1$, where fast method can be used for W^1x.

4 Once the eigen vectors 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.

5 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.

6. The number of groups segmented by this method is controlled directly by the maximum allowed Ncut.

3.1 Simultaneous K-Way Cut with Multiple Eigenvectors

One drawback of the recursive 2-way cut is its treatment of the oscillatory eigenvectors. The stability criteria keeps us from cutting oscillatory eigenvectors, but it also prevents us cutting the subsequent eigenvectors which might be perfect partitioning vectors. Also, the approach is computationally wasteful; only the second eigenvector is used, whereas the next few small eigenvectors also contain useful partitioning information.

Instead of finding the partition using recursive 2-way cut as described above, one can use all of the top eigenvectors to simultaneously obtain a K-way partition. In this method, the n top eigenvectors are used as n dimensional indicator vectors for each pixel. In the first step, a simple clustering algorithm, such as the k-means algorithm, is used to obtain an oversegmentation of the image into k0 groups. No attempt is made to identify and exclude oscillatory eigenvectors-they exacerbate the over segmentation, but that will be dealt with subsequently.
In the second step, one can proceed in the following two ways:

1. Greedy pruning: Iteratively merge two segments at a time until only k segments are left. At each merge step, those two segments are merged that minimize the k-way Ncut criterion defined as:

Ncut k = cut(W1,V-W1) / assoc(W1,V) + cut(W2,V-W2) / assoc(W2,V) + ... + cut(Wn,V-Wn) / assoc(Wn,V)

where W_i is the i^{th} subset of whole set V.

This computation can be efficiently carried out by iteratively updating the compacted weight matrix W^c, with $W^c(i, j) = assoc(Wi,Wj)$

2. Global recursive cut. From the initial k′ segments, we can build a condensed graph $G^c = (V^c, E^c)$ where each segment Wi corresponds to a node V^c_i of the graph. The weight on each graph edge $W^c(i, j)$ is defined to be $assoc(W_i, W_j)$ the total edge weights from elements in W_i to elements in W_j From this condensed graph, we then recursively bipartition the graph according the Ncut criterion. This can be carried out either with the generalized eigenvalue system or with exhaustive search in the discrete domain. Exhaustive search is possible in this case since k′ is small, typically k′ ≤ 100.

4. EXPERIMENTS

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 each pixel as a node and define the edge weight wij between node i and j as the product of a feature similarity term and spatial proximity term:

Wij = $\exp(-\|F(i)-F(j)\|^2 / \sigma^2 I)$ *
{ $\exp(-\|X(i)-X(j)\|^2 / \sigma^2 X)$ if $\|X(i)-X(j)\|_2 < r$
0 otherwise

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:
*. F(i)=1, in the case of segmenting point sets,

* F(i)=I(i)., the intensity value, for segmenting brightness images,

*F(i) = [v, v.s.sin(h), v.s..cos(h)] (i), where h; s; v are the HSV values,
for color segmentation,
,
* F(i) = [| I * f1|, …,| I * fn|], where the fi are DOOG filters at various scales and orientations as used in [12], in the case of texture segmentation.

By applying new Ncut algorithm, the following Ncut values are obtained

Path From Root	Ncut Values
ROOT-A	0.000000
ROOT-B-A-A	0.000000
ROOT-B-A-B-A.	0.000000
ROOT-B-A-B-B-A-A-A-A-A-A.	0.000000
ROOT-B-A-B-B-A-A-A-A-A-B-A.	0.030964
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-A-A.	0.078106
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-A-B.	0.033421
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-A.	0.011684
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-A-	0.073347
A-A.	0.121786
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-A-	0.062340
A-B	-0.000000
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-A-B.	0.013397
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-A.	0.000000
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	0.000000
A-A-A-A	0.027056
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	0.000000
A-A-A-B	-0.000000
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	0.097920
A-A-B-A	0.072152
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	0.014628
A-A-B-B.	0.071980
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	-0.000000
A-B.	0.021055
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	0.050144
B-A.	0.000000
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-A-B-B-B-B-	0.033410
B-B	0.084656
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-B-A.	
ROOT-B-A-B-B-A-A-A-A-A-B-B-A-B-B.	
ROOT-B-A-B-B-A-A-A-A-A-B-B-B	
ROOT-B-A-B-B-A-A-A-A-B	
ROOT-B-A-B-B-A-A-A-B.	
ROOT-B-A-B-B-A-A-B.	
ROOT-B-A-B-B-A-B.	
ROOT-B-A-B-B-B	
ROOT-B-B	

Fig 5. Ncut values for the image bear by applying new efficient ncut algorithm with SI =5; SX =6; r = 1.5; sNcut = 0.14; sArea = 220

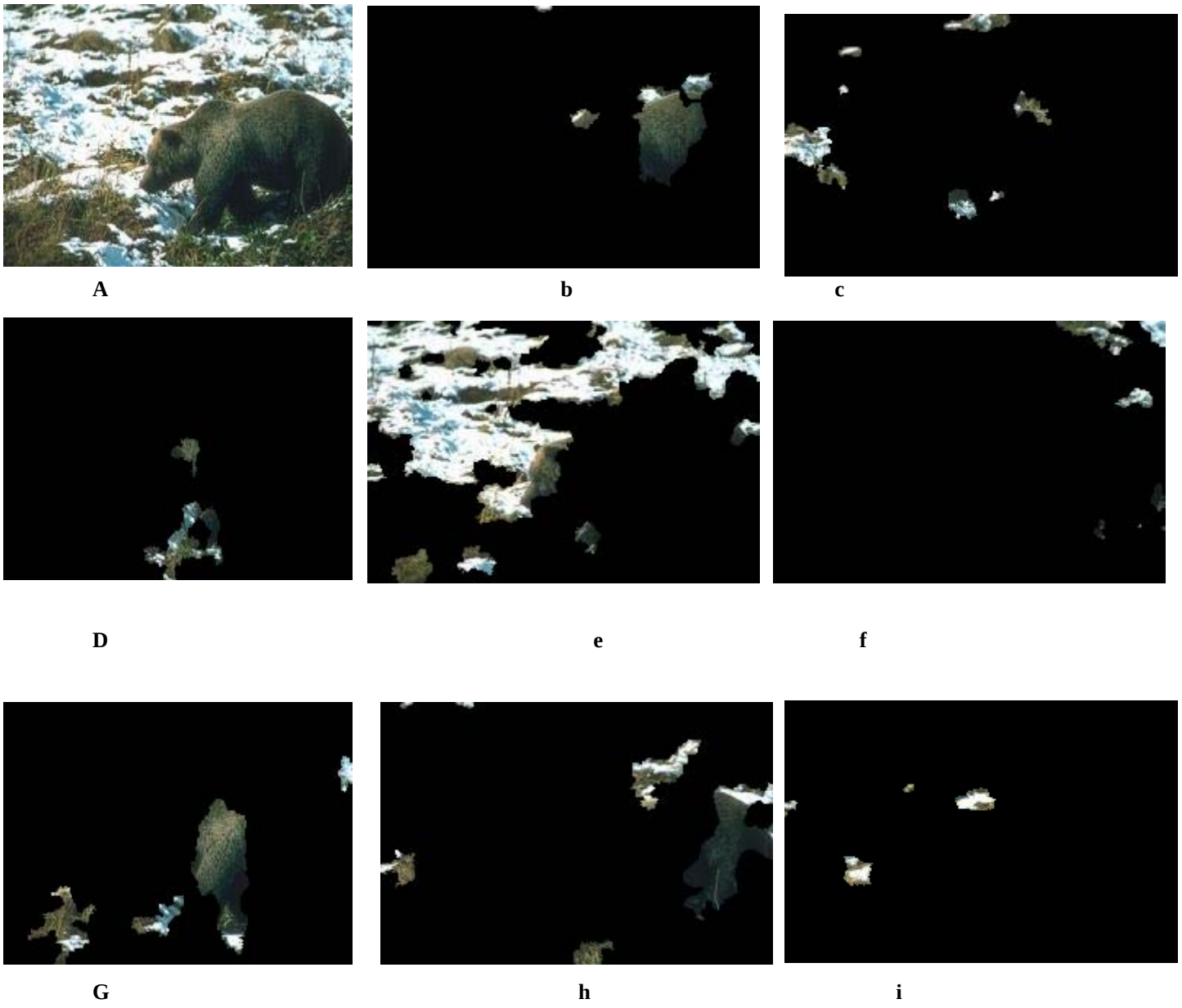


Fig 6: a) original image, b-i: segmented images by applying new method

Fig 5 shows ncut values and Fig 6 shows corresponding segmented images by applying new ncut method

4.1 Computation Time

With reference to [13] , old ncut method using general eigen value computation takes $O(n^3)$. The following figures shows new method reduce the time than old method

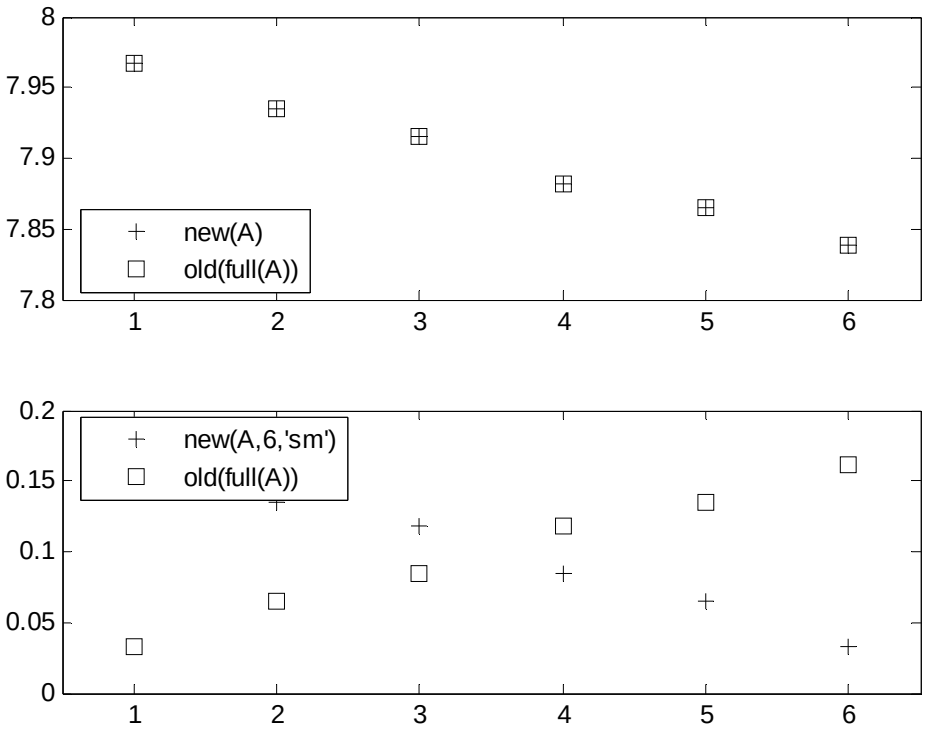


Fig 7: a) old and new method perform well for producing limited eigen values and b) new method perform well for producing large eigen values with less computation time

In Fig 7, Legend + indicates the performance of new method with running time 0.156 seconds and square indicates the worst behavior of the old method with running time 0.266 seconds

5. CONCLUSION

We Conclude that average time complexity of the above algorithm is O(n log n). This is more reduced time complexity than using of standard eigen value problem method. In Future work, we may introduce new eigen value method to reduce the time as O(n).

REFERENCES

[1] M. Wertheimer, ªLaws of Organization in Perceptual Forms (partial translation),º A Sourcebook of Gestalt Psycychology, W.B. Ellis, ed.,p. 71-88, Harcourt, Brace, 1938.

[2]S.GemanandD.Geman, ªStochastic Relaxation, Gibbs Distributions, an the Bayesian Restoration of Images, ºIEEE Trans. Pattern Analysis and Machine Intelligence, vol. 6, pp. 721-741, Nov. 1984.

[3] A.K. Jain and R.C. Dubes,Algorithms for Clustering Data. Prentice Hall, 1988.

[4] A .blake. and A. Zisserman, Visual Reconstruction. MIT Press, 1987.

[5] Y.G. Leclerc, ªConstructing Simple Stable Descriptions for Image Partitioning,º Int'l J. Computer Vision, vol. 3, pp. 73-102, 1989.

[6] D. Mumford and J. Shah, ªOptimal Approximations by Piecewise Smooth Functions, and Associated Variational Problems,º Comm. Pure Math., pp. 577-684, 1989.

[7] Charless, F. & Serge, B. & Fan C. & Jitendra M. (2004) Spectral Grouping Using the Nystrom Method. IEEE Transactions on

Pattern Analysis an Machine Intelligence. pp. 214-225.

[8] Marina, M. & Susan, S. & Liang X. (2004) Regularized spectral learning UW- Stat Dept TR # 465.

[9] Timothee, C. & Nicolas, G. & Jianbo S. (2005) Learning Spectral Graph Segmentation AISTATS 2005[

[10] Francis, R.B. & Michael, I.J. (2003) Learning Spectral Clustering. NIPS 2003

[11] Z. Wu and R. Leahy, ªAn Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentationº IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15,no. 11, pp. 101-1,113, Nov. 1993.

[12] J. Malik and P.PeronaªPreattentive Texture Discrimination with Early Vision Mechanisms,º J. Optical Soc. Am., vol. 7, no. 2, pp. 923- 932, May 1990.

[13] S.V Kasmir Raja A.Shaik Abdul Khadir and S.S. Riaz Ahamed studying Feasibility and importance of Graph based image segmentation Techniques "Journal of Theoretical and applied information Technologies" vol 5, no. 1 ,pp 056-063, 2009