

ECE 278 Final Report

Normalized Cuts and Image Segmentation

Baris Sumengen, Jelena Tešić

Abstract

Our project work includes the embedding of a normalized cuts algorithm and testing its application for image segmentation. Proposed solution minimizes a graph bi partitioning based on a modified cost function. Final result has a form of partition candidates based on a low-level image description. To increase the stability of the proposed method, algorithm has an iterative form. Computational complexity issues remain unsolved, even though the method performs well for a preprocessing.

1. Introduction

The importance and difficulties of perceptual grouping have been very well studied in vision research and image processing literature. How to partition image to subsets, when there are many possible ways of doing it? The intuitive answer would be a partitioning based on the available prior knowledge. The difficulty is specifying the knowledge in a compact and useful way i.e. defining the descriptors of an image. It is clear that the use of low-level descriptors could not lead to a complete and correct final segmentation. Instead, image attributes should be used for obtaining candidate partitions. Higher-level descriptors can be used in decision process and “finer tuning” of the proposed segments.

Over the decades, different scientific communities proposed various approaches to generalized image segmentation problem. Tree-based hierarchical clustering structures, region-based merge and split image processing algorithms and statistical approach to image segmentation introduced additional constraints on a possible solution. Before any algorithm implementation we have to determine what criterion we want to optimize and if there is an efficient algorithm for implementation method.

We separate the proposed method from the ones mentioned above because it has a novel approach to an image. It is taking a global image feature descriptor as a weighted graph and reduces segmentation to an optimal graph partitioning. New optimization criterion - normalized cut - is used for finding the optimal partitioning. The graph partitioning approach and related work is explained in the section 2. Normalized cuts criterion and its justification are presented in the section 3. The algorithm and its properties are presented in the section 4. Conducted experiments and results are outlined in the section 5. Section 6 discusses the proposed approach and derives the project conclusion.

2. Graph Partitioning

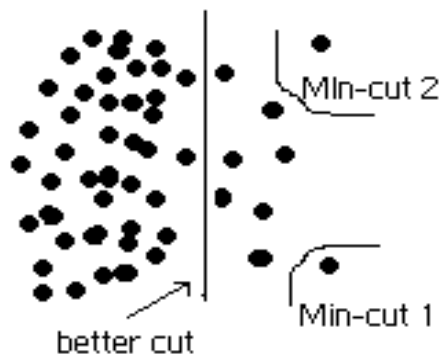
The set of points in an arbitrary feature space is presented as a weighted undirected graph $G=(V,E)$. Nodes of the graph are the points in the feature space. An edge is formed between every pair of nodes and the weight on each edge $W(i,j)$ is a function of the similarity between nodes i and j . A graph $G=(V,E)$ is partitioned into two disjoint complementary sets A and B, $B=V-A$, by removing the edges connecting two parts. The degree of dissimilarity between two sets can be computed as a total weight of removed edges. That closely relates to a mathematical formulation of a cut (1):

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

2.1. Related Approaches

Problem formulation is to find an optimal bi partitioning of a graph G . The problem reduces to finding a cost function that will give us perceptually good results (since we're dealing with feature descriptors of an image). Global optimum is achieved if we minimize the *cut* value. If N is cardinality of V , $N=|V|$, there is an exponential number of possible partitions 2^N . This optimization is well studied and there is an efficient algorithm for finding a *minimum cut* [3].

Wu and Leahy proposed a clustering method based on the minimum cut criterion [5]. They are trying to partition the graph into k sub graphs so that the maximum cut across the partitions is minimized. They introduced an efficient way to obtain the solution by recursively finding the minimum cut that bisects the existing elements. This method produced some good segmentation results. Authors noticed that the performance is highly influenced by the graph outliers. From the definition of global optimum (1), this is an expected outcome. The following example illustrates the bad partitioning case:



Cox [7] proposed a modified cost function (2), where $weight(A)$ is some function of set A, i.e. sum of elements in A.

$$WCut(A) = \frac{cut(A, V - A)}{weight(A)} \quad (2)$$

2.2. Normalized Cut

Shi and Malik [1,2] propose a modified cost function, *normalized cut*, to overcome outliers. Instead of looking at the value of total edge weight connecting the two partitions, the proposed measure computes the cut cost as a fraction of the total edge connections to all nodes (3):

$$Ncut(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)}, \quad asso(A, V) = \sum_{u \in A, t \in V} w(u, t) \quad (3)$$

The association value, $asso(A, V)$, is the total connection from nodes A to all nodes in the graph. $Ncut$ value won't be small for the cut that partitions isolating points, because the cut value will be a large percentage of the total connection from that set to the others.

We present an efficient algorithm for graph partitioning using the normalized cut as an optimal criterion. Given a partition of a graph V into two disjoint complementary sets A and B , let x be an $N=|V|$ dimensional indication vector, $x_i=1$ if node i is in A , -1 otherwise. Let $d_i = \sum_j W(i, j)$ be the total connection from node i to all other nodes. $Ncuts$ can be rewritten as (4).

$$Ncut(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)} = \frac{\sum_{x_i > 0, x_j < 0} -w_{ij} x_i x_j}{\sum_{x_i > 0} d_i} + \frac{\sum_{x_i < 0, x_j > 0} -w_{ij} x_i x_j}{\sum_{x_i < 0} d_i} \quad (4)$$

Let $D = diag(d_1, d_2, \dots, d_N)$ and $W(i, j) = w_{ij}$. Finding global optimum reduces to (5):

$$\min_x Ncut(x) = \min_y \frac{y^T (D - W) y}{y^T D y}, \quad y_i \in \left\{ 1, \frac{-\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i} \right\}, \quad y^T \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} = 0 \quad (5)$$

If y is relaxed to take on real values, we can minimize (5) by solving generalized eigenvalue system:

$$(D - W)y = \lambda D y \quad (6)$$

Constraints on y come from the condition on the corresponding indicator vector x . The second smallest eigenvector of the generalized system (6) satisfies the normality constraint. y_1 is a real valued solution to our normalized cut problem (7). The constraint, that y_1 takes on the two discrete values, is not satisfied. In the next section, we'll show how to obtain discrete values.

$$y_1 = \min_{y^T \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} = 0} \frac{y^T (D - W) y}{y^T D y} \quad (7)$$

The idea of using eigenvalue approach for graph partitioning originated in early 70s [4]. In fact, the second smallest eigenvalue is called the Fiedler value, by the author who suggested its use of it for splitting a graph[5]. Sarkar & Boyer [3] use the eigenvector with the largest eigenvalue of the system $Wx = \lambda x$ for finding the most coherent region in an edge map.

3. The grouping algorithm

The eigenvector corresponding to the second smallest eigenvalue is the real-valued solution that optimally sub-partitions the entire graph, the third smallest value is a solution that optimally partitions the first part into two, etc. One can subdivide the existing graphs each time using the eigenvector with the next smallest eigenvalue. However, since the approximation error accumulates with every eigenvector taken and we want eigenvectors with discrete elements to satisfy the orthogonal property, the solution based higher eigenvector become unreliable. We solve the partitioning problem for every sub graph separately and this is implemented as a recursive grouping algorithm. Eigenvalue computation is expensive. We are using the approximate Lanczos method for computation the eigenvector of a very sparse matrix where we need only couple of eigenvectors. Computational cost is typically less than $O(n^{3/2})$ where n is the number of nodes in the graph.

Graph is partitioned using the second smallest eigenvector. Eigenvector elements take on continuous values, so we need to define a splitting point for eigenvector. Authors adopt a heuristic method for splitting point search. Number of evenly spaced splitting points in the vector range is checked, and normalized cuts are computed for each point. The point that gives the minimum $Ncut$ value is chosen as a splitting point. The other approach could be to take 0 or a median value as a splitting point, but it wasn't really reliable because of the approximation error. Both methods are good enough if values are very well separated. We recursively apply the algorithm to every sub graph until the $Ncut$ exceeds certain limit.

Output of the one algorithm pass gives us partition candidates. Cut is unstable if $Ncut$ value does not change much when varying the set of graph edges forming the cut.

Summarized algorithm flow looks like:

1. Define the feature description matrix for a given image and a weighting function
2. Set up a weighted graph $G=(V,E)$, compute the edge weights and summarize information into W and D
3. Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues.
4. Use the eigenvector with second smallest eigenvalue to bipartition the graph by finding the splitting points such that $Ncut$ is minimized.
5. Decide if the current partition is stable and check the value of the resulting $Ncut$
6. Recursively repartition the segmented parts (go to step2)
7. Exit if $Ncut$ for every segment is over some specified value – maximum allowed $Ncut$

4. Experiments and Results

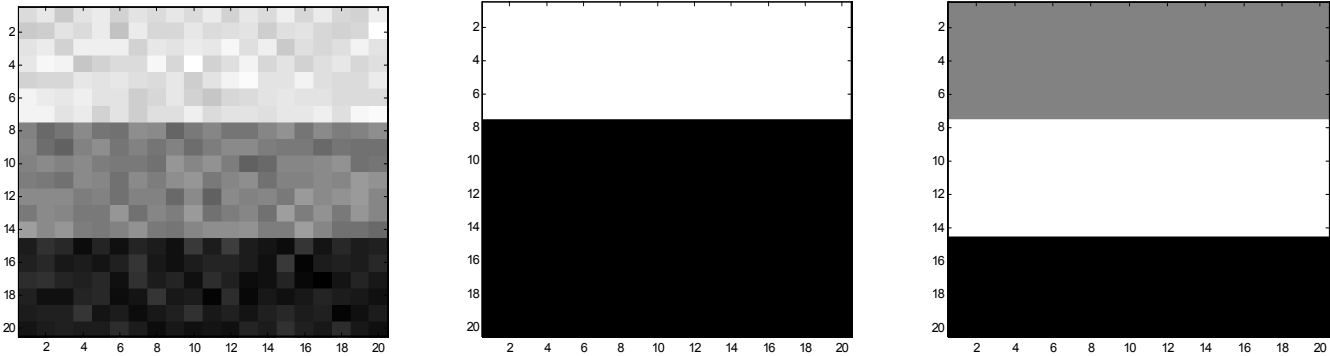
We applied the partitioning algorithm to gray-scaled and color images. We constructed the graph $G=(V,E)$ by taking each pixel as a node and defining the edge weight function as following:

$$W(i, j) = e^{-\frac{\|F_i - F_j\|_2}{\sigma_I}} * e^{-\frac{\|X_i - X_j\|_2}{\sigma_X}}, \quad \left\| X_i - X_j \right\|_2 < r \quad F(i) = \begin{cases} I(i), & \text{intensity} \\ [v, v \cdot s \cdot \sin(h), v \cdot s \cdot \cos(h)](i), & \text{color} \\ [|I * f_1|, \dots, |I * f_n|](i), & \text{texture} \end{cases}$$

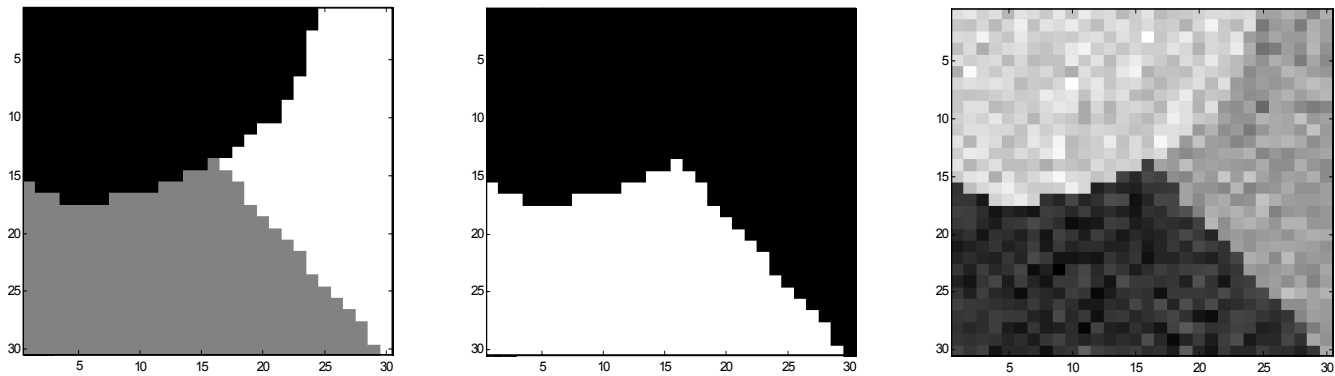
4.1. Texture Images - descriptors formed based on intensities

The weight $w_{ij}=0$ for any pair of nodes i and j that are more than r pixels apart.

$$\sigma = 0.07, r = 3.0, \sigma_I = 0.02, \sigma_X = 3.0$$



$$\sigma = 0.1, r = 5.0, \sigma_I = 0.01, \sigma_X = 4.0$$

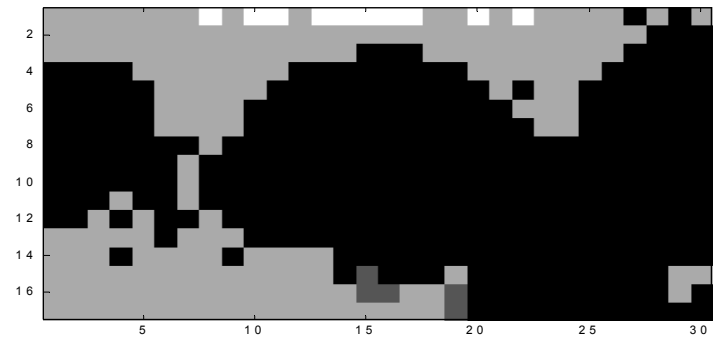
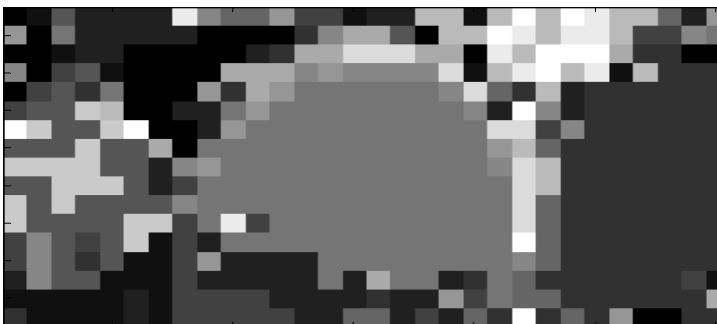
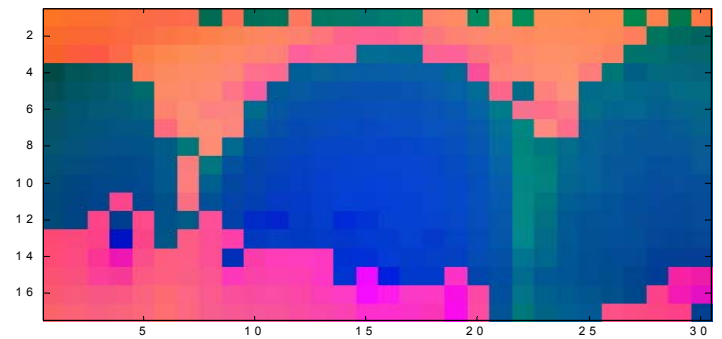
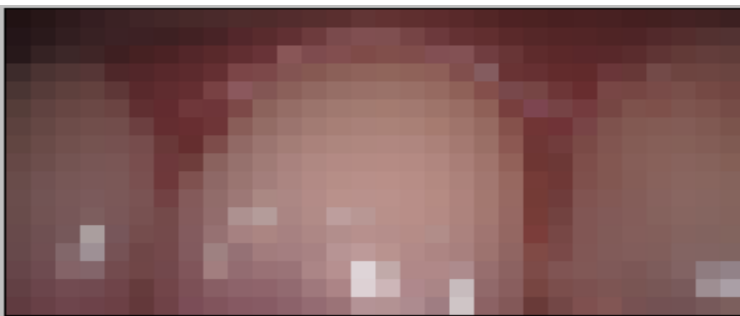
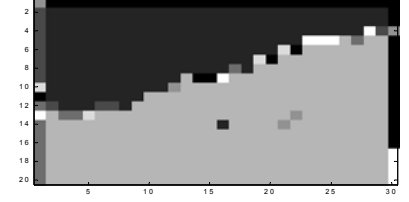
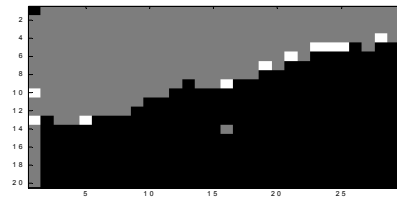
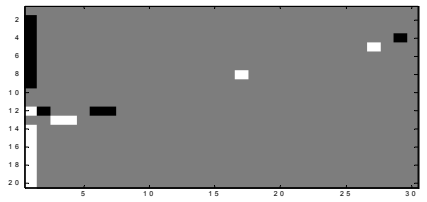
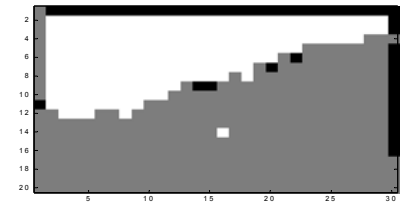
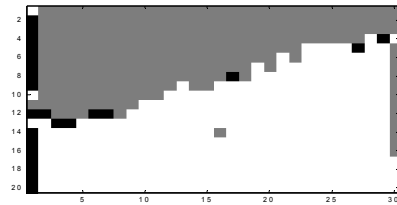
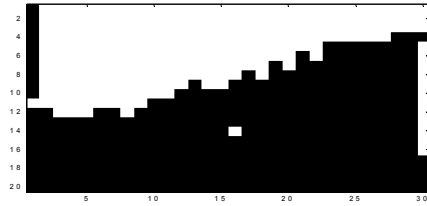
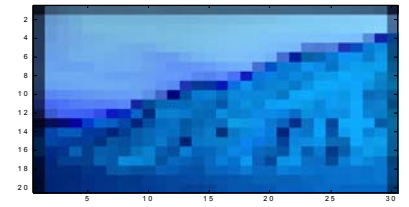
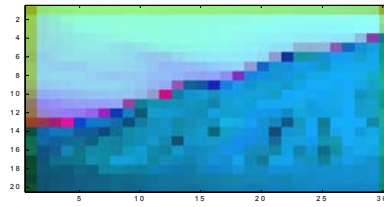


Normalized cuts algorithm is slow. Main reason is that, if the image dimension is $n \times n$, we have to keep $n^2 \times n^2$ matrix in the main memory. Based on that reasoning, we have decided to use 20×20 gray scaled images. If our original image is larger, we subsampled it. Since this algorithm is based on low-level descriptors, use of a coarse image representation is justified. We're looking for partition candidates, not trying to produce a fine segmentation.

The algorithm is implemented in MATLAB. Few eigenvector are found using approximate iterative MATLAB function EIGS. Split point is found among 128 equally spaced points in the vector element range, as the one that gives the smallest N_{cut} value.

Gray-scaled image segmentation is an example of image enhancement. We introduced a Gaussian noise to the image. Segmentation results highly depend on a chosen parameters. These simple examples were the best starting point since the format of the weighted function gives the best results for Gaussian noise.

4.2. Gaussian noise, Color Images - descriptors formed based on HSV values



For color based segmentation, feature vector is defined as $F(i) = [v, v \cdot s \cdot \sin(h), v \cdot s \cdot \cos(h)](i)$, where h, s, v are the HSV values. Again, depending on the parameters, for the teeth image we get small number of well-defined segments or a big number of small, not perceptually different segments.

5. Conclusion

We implemented a normalized cut algorithm and used it for the intensity and color segmentation. Even though the approximate eigenvalue method and the algorithm construction optimize implementation, computational complexity remains unsolved for the full-scale image. The performance and stability of the partitioning highly depends on the choice of parameters.

We have noticed that the choice of parameters is data-dependent in some sense, but authors don't offer the solution and do not justify their choice of parameters. Heuristic method for finding the splitting point is questionable. Smallest, non-zero eigenvalues, for given eigensystem with sparse matrices, often have really small magnitude and the choice of splitting point is influenced by computation precision.

Stability is controlled by many factors: choice of splitting point, eigenvalue computations and relative segment position. Authors [2] didn't suggest how to deal with border segments. We modify matrix D and W depending on the relative segment position. Otherwise, algorithm becomes unstable and starts partitioning homogeneous regions (since neighboring regions are not taken into account). The other approach is to choose maximum $Ncut$ for every segment but that would be difficult to implement, if not impossible.

Consequently, Normalized cut application is highly constrained by the previous facts.

6. References

- [1] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *Int. Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997.
- [2] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *Technical Report CSD-97-940*, UC Berkley, 1997.
- [3] A. Pothen and H. Simon, "Partitioning Sparse Matrices with Eigenvectors of Graphs", *IBM Journal of Research and Development*, pp. 420-425, 1973.
- [4] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory", *Czech. Math. J.* 25, pp.619--637, 1975.
- [5] Wu, Z., and Leahy, R., "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation", *PAMI (15)*, No. 11, pp. 1101-1113, 1993.