Normalized Cuts and Image Segmentation

Jianbo Shi and Jitendra Malik
Computer Science Division
University of California at Berkeley, Berkeley, CA 94720
{jshi,malik}@cs.berkeley.edu

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 eigenvalue problem can be used to optimize this criterion. We have applied this approach to segmenting static images and found results very encouraging.*

1 Introduction

Nearly 75 years ago, Wertheimer[17] launched the Gestalt approach which laid out the importance of perceptual grouping and organization in visual perception. For our purposes, the problem of grouping can be well motivated by considering the set of points shown in the figure (1).

Typically a human observer will perceive four objects in the image–a circular ring with a cloud of points inside it, and two loosely connected clumps of points on its right. However this is not the unique partitioning of the scene. One can argue that there are three objects–the two clumps on the right constitute one dumbbell shaped object. Or there are only two objects, a dumb bell shaped object on the right, and a circular galaxy like structure on the left. If one were perverse, one could argue that in fact every point was a distinct object.

This may seem to be an artificial example, but every attempt at image segmentation ultimately has to confront a similar question–there are many possible partitions of the domain D of an image into subsets D_i (including the extreme one of every pixel being a separate entity). How do we pick the "right" one? We believe the Bayesian view is appropriate– one wants to find the most probable interpretation 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.

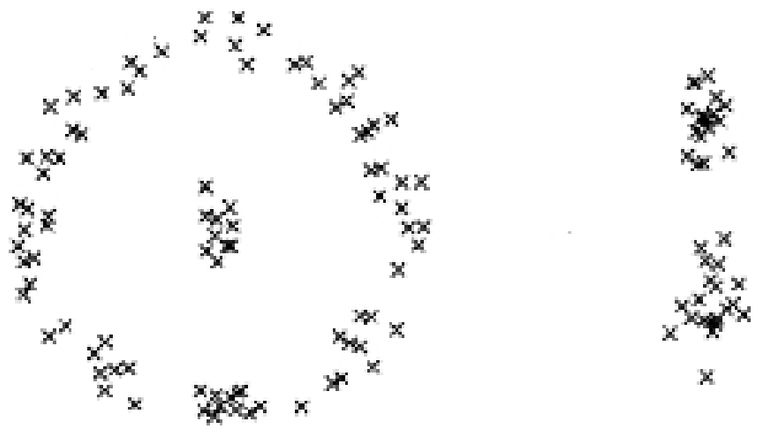

Figure 1: How many groups?

This suggests to us that image segmentation based on low level cues can not 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 candidate 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 downwards, rather like a painter first marking out the major areas and then filling in the details.

Prior literature on the related problems of clustering, grouping and image segmentation is huge. The clustering community[9] has offered us agglomerative and divisive algorithms; in image segmentation we have region-based merge and split algorithms. The hierarchical divisive approach that we are advocating produces a tree, the *dendrogram*. While most of these ideas go back to the 70s (and earlier), the 1980s brought in the use of Markov Random Fields[7] and variational formulations[13, 2, 11]. The MRF and variational formulations also exposed two basic questions (1) What is the criterion that one wants to optimize? and (2) Is there an efficient algorithm for carrying out the optimization? Many an attractive criterion has been doomed by the inability to find an effective algorithm to find its minimum–greedy or gradient descent type approaches fail to find global optima for these high dimensional, nonlinear problems.

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

1063-6919/97 $10.00

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 $\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_m$, where by some measure the similarity among the vertices in a set $\mathbf{V}_i$ is high and across different sets $\mathbf{V}_i, \mathbf{V}_j$ is low.

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.

In this paper we propose a new graph-theoretic criterion for measuring the goodness of an image partition– the *normalized cut*. We introduce and justify this criterion in section 2. The minimization of this criterion can be formulated as a generalized eigenvalue problem; the eigenvectors of this problem can be used to construct good partitions of the image and the process can be continued recursively as desired(section 3). In section 4 we show experimental results. The formulation and minimization of the normalized cut criterion draws on a body of results, theoretical and practical, from the numerical analysis and theoretical computer science communities–section 5 discusses previous work on the spectral partitioning problem. We conclude in section 6.

2 Grouping as graph partitioning

A graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ can be partitioned into two disjoint sets, A, B, $A \cup B = \mathbf{V}$, $A \cap B = \emptyset$, 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 the *cut*:

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

The optimal bi-partitioning of a graph is the one that minimizes this *cut* value. Although there are 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.

Wu and Leahy[18] 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 & Leahy's work, this

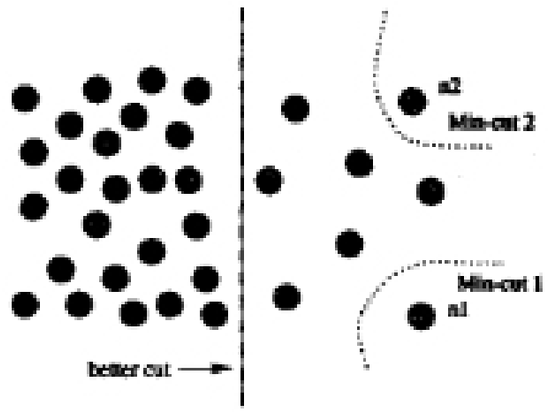

Figure 2: A case where minimum cut gives a bad partition.

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. Figure (2) 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 n_1 or n_2 will have a very small value. In fact, any cut that partitions 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.

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*):

$$Ncut(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)} \qquad (2)$$

where $asso(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in A to all nodes in the graph, and $asso(B, V)$ 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 figure 2, we see that the cut_1 value across node n_1 will be 100% of the total connection from that node.

In the same spirit, we can define a measure for total normalized association within groups for a given partition:

$$Nasso(A, B) = \frac{asso(A, A)}{asso(A, V)} + \frac{asso(B, B)}{asso(B, V)} \qquad (3)$$

where $asso(A, A)$ and $asso(B, B)$ 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:

$$\begin{aligned} Ncut(A,B) &= \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)} \\ &= \frac{asso(A,V) - asso(A,A)}{asso(A,V)} \\ &\quad + \frac{asso(B,V) - asso(B,B)}{asso(B,V)} \\ &= 2 - \left(\frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)}\right) \\ &= 2 - Nasso(A,B) \end{aligned}$$

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

Having defined the graph partition criterion that we want to optimize, we will show how such an optimal partition can be computed efficiently.

2.1 Computing the optimal partition

Given a partition of nodes of a graph, V, into two sets A and B, let $\boldsymbol{x}$ be an $N = |\boldsymbol{V}|$ dimensional indicator vector, $\boldsymbol{x}_i = 1$ if node i is in A, and -1 otherwise. Let $\boldsymbol{d}(\mathrm{i}) = \sum_j w(i,j)$, be the total connection from node i to all other nodes. With the definitions $\boldsymbol{x}$ and $\boldsymbol{d}$ we can rewrite $Ncut(A,B)$ as:

$$\begin{aligned} Ncut(A,B) &= \frac{cut(A,B)}{asso(A,V)} + \frac{cut(B,A)}{asso(B,V)} \\ &= \frac{\sum_{(\boldsymbol{x}_i>0,\boldsymbol{x}_j<0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i>0}\boldsymbol{d}_i} \\ &\quad + \frac{\sum_{(\boldsymbol{x}_i<0,\boldsymbol{x}_j>0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i} \end{aligned}$$

Let $\mathbf{D}$ be an $N \times N$ diagonal matrix with $\boldsymbol{d}$ on its diagonal, $\mathbf{W}$ be an $N \times N$ symmetrical matrix with $\mathrm{W(i\,j)} = w_{ij}$, $k = \frac{\sum_{\boldsymbol{x}_i>0}\boldsymbol{d}_i}{\sum_i \boldsymbol{d}_i}$, and $\mathbf{1}$ be an $N \times 1$ vector of all ones. Using the fact $\frac{\mathbf{1}+\boldsymbol{x}}{2}$ and $\frac{\mathbf{1}-\boldsymbol{x}}{2}$ are indicator vectors for $\boldsymbol{x}_i > 0$ and $\boldsymbol{x}_i < 0$ respectively, we can rewrite $4[Ncut(\boldsymbol{x})]$ as:

$$\begin{aligned} &= \frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{k\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ &= \frac{(\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}+\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1})}{k(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} + \frac{2(1-2k)\mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{k(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} \end{aligned}$$

Let $\alpha(\boldsymbol{x}) = \boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}$, $\beta(\boldsymbol{x}) = \mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}$, $\gamma = \mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1}$, and $M = \mathbf{1}^T\mathbf{D}\mathbf{1}$, we can then further expand the above equation as:

$$\begin{aligned} &= \frac{(\alpha(\boldsymbol{x})+\gamma)+2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} \\ &= \frac{(\alpha(\boldsymbol{x})+\gamma)+2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} - \frac{2(\alpha(\boldsymbol{x})+\gamma)}{M} \\ &\quad + \frac{2\alpha(\boldsymbol{x})}{M} + \frac{2\gamma}{M} \end{aligned}$$

dropping the last constant term, which in this case equals 0, we get

$$\begin{aligned} &= \frac{(1-2k+2k^2)(\alpha(\boldsymbol{x})+\gamma)+2(1-2k)\beta(\boldsymbol{x})}{k(1-k)M} + \frac{2\alpha(\boldsymbol{x})}{M} \\ &= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(\boldsymbol{x})+\gamma) + \frac{2(1-2k)}{(1-k)^2}\beta(\boldsymbol{x})}{\frac{k}{1-k}M} + \frac{2\alpha(\boldsymbol{x})}{M} \end{aligned}$$

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

$$\begin{aligned} &= \frac{(1+b^2)(\alpha(\boldsymbol{x})+\gamma)+2(1-b^2)\beta(\boldsymbol{x})}{bM} + \frac{2b\alpha(\boldsymbol{x})}{bM} \\ &= \frac{(1+b^2)(\alpha(\boldsymbol{x})+\gamma)}{bM} + \frac{2(1-b^2)\beta(\boldsymbol{x})}{bM} + \frac{2b\alpha(\boldsymbol{x})}{bM} - \frac{2b\gamma}{bM} \\ &= \frac{(1+b^2)(\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}+\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1})}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \\ &\quad + \frac{2(1-b^2)\mathbf{1}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \\ &\quad + \frac{2b\boldsymbol{x}^T(\mathbf{D}-\mathbf{W})\boldsymbol{x}}{b\mathbf{1}^T\mathbf{D}\mathbf{1}} - \frac{2b\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1}}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \\ &= \frac{(\mathbf{1}+\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \\ &\quad + \frac{b^2(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\boldsymbol{x})}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \\ &\quad - \frac{2b(\mathbf{1}-\boldsymbol{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\boldsymbol{x})}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \\ &= \frac{[(\mathbf{1}+\boldsymbol{x})-b(\mathbf{1}-\boldsymbol{x})]^T(\mathbf{D}-\mathbf{W})[(\mathbf{1}+\boldsymbol{x})-b(\mathbf{1}-\boldsymbol{x})]}{b\mathbf{1}^T\boldsymbol{D}\mathbf{1}} \end{aligned}$$

Setting $\boldsymbol{y} = (\mathbf{1}+\boldsymbol{x}) - b(\mathbf{1}-\boldsymbol{x})$, it is easy to see that

$$\boldsymbol{y}^T\boldsymbol{D}\mathbf{1} = \sum_{\boldsymbol{x}_i>0}\boldsymbol{d}_i - b\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i = 0 \qquad (4)$$

since $b = \frac{k}{1-k} = \frac{\sum_{\boldsymbol{x}_i>0}\boldsymbol{d}_i}{\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i}$, and

$$\begin{aligned} \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} &= \sum_{\boldsymbol{x}_i>0}\boldsymbol{d}_i + b^2\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i \\ &= b\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i + b^2\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i \\ &= b(\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i + b\sum_{\boldsymbol{x}_i<0}\boldsymbol{d}_i) \\ &= b\mathbf{1}^T\boldsymbol{D}\mathbf{1}. \end{aligned}$$