Lecture 7

A.L. Yuille

February 5, 2012

1 Spectral Clustering

Spectral clustering is a technique for segmenting data into non-overlapping subsets. It is used in many machine learning applications (e.g., von Luxberg 2007) and was introduced into computer vision by Shi and Malik (2000).

The data is defined by a graph with an *affinity*, or similarity measure, between graph nodes. The computation – to segment the data – can be performed by linear algebra followed by thresholding. Note that affinities relates to kernels (those that fall off with distance, like radial basis functions) used in machine learning. For some problems it is easier to define affinities between objects directly instead of obtaining them by the more standard method of specifying the objects by features and then calculating the distance between the features.

Spectral clustering is an alternative to probabilistic methods for segmentation. The main difference is that the probabilistic models define data at the nodes of the graph while spectral clustering defines data at the edges between nodes. (There are ways to relate the two approaches which will be discussed later).

For image segmentation a typical affinity between pixels i and j is defined by $w_{ij} = \exp\{-\gamma|I_i - I_j|\}\exp\{-\tau|x_i - x_j|\}$ where I_i, I_j are the intensities at pixels i, j and x_i, x_j are their spatial positions. Hence the affinity is high between neighboring pixels which have similar intensity values (small $x_i - x_j$ and small $|I_i - I_j|$) and the affinity is small between pixels which are far apart (large $x_i - x_j|$) or which have very different intensity values (large $|I_i - I_j|$). If this affinity is used, the spectral clustering will segment the data into subregions within which the intensity values changes slowly with position.

References

- U. von Luxburg, A tutorial on spectral clustering, Stat.Comput.,2007
- Shi and Malik, Normalized cuts and image segmentation, PAMI, 2000

2 Basic Concepts

Let $G = (V, E)$ be an undirected graph with nodes $V = v_1, v_2, ..., v_n$. The graph has weighted edges $w_{ij} = w_{ji} \geq 0$ which are called affinities and are measures of similarity between nodes. Large w_{ij} means strong affinities, or bonds, between node i and node j.

The degree of a vertex v_i is defined as: $d_i = \sum_{j=1}^{n} w_{ij}$. The degree matrix of the graph is defined by $\mathbf{D} = diag\{d_1, d_2, ..., d_n\}$.

For two subsets A, B (which do not need to be disjoint) of V, $w(A, B)$ is defined by:

$$w(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

The size of a subset $A \subseteq V$ has two definitions.

$$|A| = number\ of\ vertices\ in\ A \quad \text{(unweighted volume)}$$
$$vol(A) = \sum_{i \in A} d_i \quad \text{(weighted volumn)}$$

2.1 Examples of affinities

$$w_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Shi and Malik's definition of w_{ij} is given by:

$$w_{ij} = \begin{cases} e^{-\frac{\|I_i - I_j\|^2}{\sigma_I^2}} \times e^{-\frac{\|x_i - x_j\|^2}{\sigma_x^2}} & \text{if } \|I_i - I_j\| < r \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where r, σ_x, σ_I are parameters.

3 The Graph Laplacian Matrix (ref. Chung, Spectral graph theory, AMS-1997)

This section defines graph Laplacians on the graphs. In the special case where the affinities w_{ij} takes values $\{0, 1\}$ then the connected components of the graph can be found from the eigenvectors with zero eigenvalues. This enables us to segment the graph into its connected components by linear algebra. If we allow the w_{ij} to take continuous values, as will happen for computer vision applications, then we can estimate a segmentation of the data by using the eigenvectors of the Laplacian with sufficiently small eigenvalues. There are several different Laplacians (normalized and unnormalized) which will give different segmentations.

3.1 Unnormalized graph Laplacians

Given an undirected, weighted graph $G = (V, E)$, its Laplacian matrix is defined to be

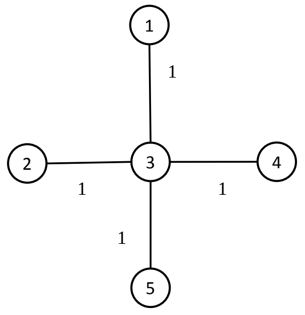
$$L := D - W \tag{3}$$

where $D = diag(d_i)$ and $W = (w_{ij})_{n \times n}$ with $n = |V|$. Note that L does not depend on w_{ii} (which cancels between D and W).

Why do we call this matrix the "Laplacian"? From Figure 1, we see that it is similar to the standard discretization for the Laplacian differential operator $-\nabla^2 u = -(u_{xx} + u_{yy})$ in the special case where $w_{ij} = 1$ for nearest neighbor pixels and $w_{ij} = 0$ otherwise:

Here are some useful properties of L which will be useful for spectral clustering:

1. $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2, \forall f \in \mathbb{R}^n$
2. L is a symmetric positive semi-definite matrix (this follows from 1).
3. The smallest eigenvalue is 0, the corresponding eigenvector is the vector $\mathbb{1} = (1, 1, ..., 1)$.
4. L has n non-negative eigenvalues. $0 = \lambda_1 \le \lambda_2 \le \dots \le \lambda_k < \lambda_{k+1} \le \dots \le \lambda_n$.
5. If $0 = \lambda_1 = \lambda_2 = \dots = \lambda_k < \lambda_{k+1} \le \dots \le \lambda_n$, then G has k-connected components $A_1, \dots, A_k$ ($V = \bigcup_{i=1}^{k} A_i$). The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_k}$ of these components.



$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Figure 1: A simple graph and its corresponding Laplacian

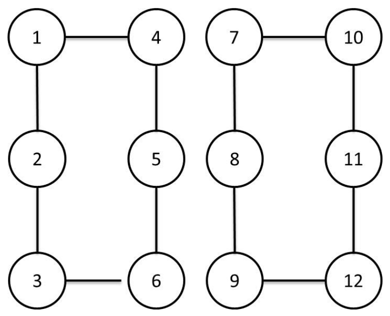


Figure 2: A simple graph G with two connected components

Here is a simple example of Property 5. The graph $G = (V, E)$ in Fig 2 has two connected components, A_1 and A_2. $W = \{w_{ij}\}$ of G is defined in Equation (6).

$$w_{ij} = \begin{cases} 1 & e_{ij} \in E \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Let $L = D{-}W$ and calculate the eigenvalues and eigenvectors of L. It follows that the first two eigenvalues λ_1, λ_2 are 0, and the corresponding eigenvectors (u_1 and u_2) correspond to the connected components A_1 and A_2.

$$A_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$
$$A_2 = \{v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$$
$$u_1 = (1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)$$
$$u_2 = (0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)$$

Note that the eigenvalues are degenerate and so that eigenvectors will be of form $cos\theta u_1 + sin\theta u_2$ and $sin\theta u_1 - cos\theta u_2$ where θ is any value. An additional algorithm is needed to find u_1 and u_2 as described in the next section.

3.2 Normalized graph Laplacians

There are two matrices which are called normalized graph Laplacians in the literature. Both matrices are closely related to each other and are defined by:

$$L_{sym} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{1/2} \quad \text{(Ng, Jordan, Weiss, 2002)} \tag{5}$$

$$L_{rw} := D^{-1}L = I - D^{-1}W. \quad \text{(Shi, Malik, 2000)} \tag{6}$$

The normalized Laplacian matrix satisfy the following properties:

1. $f^T L_{sym} f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}})^2, \forall f \in \mathbb{R}^n$
2. (λ, u) is an eigenpair of L_{rw} if and only if $(\lambda, w = D^{1/2}u)$ is an eigenpair of L_{sym}.
3. (λ, u) is an eigenpair of L_{rw} if and only if λ and u solve the generalized eigen-problem $Lu = \lambda Du$.
4. $(0, \mathbb{1})$ is an eigenpair of L_{rw}, and $(0, D^{1/2}\mathbb{1})$ is an eigenpair of L_{sym}.
5. L_{sym} and L_{rw} are positive semi-definite and have n non-negative eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$.
6. If $0 = \lambda_1 = \cdots = \lambda_k < \lambda_{k+1} \leq \cdots \leq \lambda_n$, then G has k-connected components $A_1, \ldots, A_k$. And the eigenspace of eigenvalue $\lambda = 0$ is spanned by the $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ for L_{rw} and $D^{1/2}\mathbb{1}_{A_1}, \ldots, D^{1/2}\mathbb{1}_{A_k}$ for L_{sym}.

4 Spectral Clustering Algorithms

Input: Affinity matrix $S \in \mathbb{R}^{n \times n}$, and number of clusters to construct k.

1. Compute the unnormalization Laplacian $L = D - W$.
2. Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.
   or Solve the general eigenvalue problem, i.e. $Lu = \lambda Du$, to get $u_1, \ldots, u_k$ (for L_{rw}).
   or Compute the first k eigenvectors $u_1, \ldots, u_k$ of $L_{sym} = D^{-1/2}LD^{-1/2}$(for L_{sym}).
3. Let

$$U = [u_1, \ldots, u_k] \in \mathbb{R}^{n \times k} \tag{7}$$

$$= \begin{pmatrix} y_1^T \\ \vdots \\ y_n^T \end{pmatrix}, y_i \in \mathbb{R}^k \tag{8}$$

   that is, y_i is the vector corresponding to the i-th row of U.

4. Cluster the k-dimension points $(y_i)_{i=1,\ldots,n}$ using e.g. k-means, into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$, with $A_i = \{j | y_j \in C_i\}$.
Main point: In the new representation y_i, clustering is much easier. For example, suppose $n = 5$ and there are two connected components — nodes $1, 2, 3$ and nodes $4, 5$. Then the zero eigenvectors will be of form $(cos\theta, cos\theta, cos\theta, sin\theta, sin\theta)$ and $(-sin\theta, -sin\theta, -sin\theta, cos\theta, cos\theta)$, where θ is an angle (this is because we know the zero eigenvectors must lie in the subspace spanned by $(1, 1, 1, 0, 0)$ and $(0, 0, 0, 1, 1)$, because of property 6 of the laplacian, and the eigenvectors must be orthogonal). Then if we set $k = 2$ we find that the clusters are $(cos\theta, -sin\theta)$ and $(sin\theta, cos\theta)$. Then the first three points are associated to the first cluster (i.e. the first connected component) and the last two points are associated to the second cluster (second connected component).

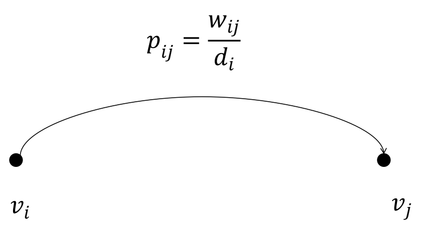


Figure 3: Random walk

5 The Graph cut point of view

We define several global measures of segmentation. These will be associated to different laplacians. They typically assume that there will be high affinity within subregions A_i and low affinities between subregions.

$$cut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i) \tag{9}$$

$$Ratiocut(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{|A_i|} \tag{10}$$

$$Ncut(A_i, \dots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{vol(A_i)} = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \tag{11}$$

These have the following relations to the three laplacians. Ratiocut $\sim$ unnormalized spectral clustering Normalized cut $\sim$ normalized spectral clustering of Shi & Malik.

6 Random walks point of view

Note: advanced topic.

A random walk on a graph is a stochastic process which randomly jumps from vertex to vertex. The transition probability of jumping in one step from vertex v_i to vertex v_j is proportional to the edge weight w_{ij} and is given by $p_{ij} := w_{ij}/d_i$, as shown in Figure 3. The transition matrix $P = (p_{ij})_{i,j=1,\dots,n}$ of the random walk is thus defined by

$$P = D^{-1}W. \tag{12}$$

If the graph is connected and non-bipartite, then the random walk always possesses a unique stationary distribution $\pi = (\pi_1, \dots, \pi_n)'$, with $\pi_i = d_i/vol(V)$.

Obviously, we have the relationship between L_{rw} and P, as $L_{rw} = I - P$. As a consequence, (λ, u) is an eigenpair of L_{rw} if and only if $(1 - \lambda, u)$ is an eigenpair of P, and the smallest eigenvectors of L_{rw} are the largest eigenvectors of P.

We obtain the following conclusion: Let G be connected and non-bipartite. Assume that we run the random walk $(X_t)_{t \in \mathbb{N}}$ starting with X_0 in the stationary distribution π. For disjoint subsets $A, B \subset V$, denote by $P(B|A) := P(X_1 \in B|X_0 \in A)$. Then:

$$Ncut(A, \bar{A}) = P(\bar{A}|A) + P(A|\bar{A}). \tag{13}$$

A second connection between random walks and graph Laplacians can be made via the commute distance on the graph. The commute distance(also called resistance distance) c_{ij} between two vertices v_i and v_j is the expected time it takes the random walk to travel from vertex v_i to vertex v_j and back. As opposed to the shortest path distance on a graph, the commute distance between two vertices decreases if there are many different short ways to get from vertex v_i to vertex v_j. So instead of just looking for the one shortest path, the commute distance looks at the set of short paths. (Ref. L.Grady, Random walks for image segmentation, PAMI-06).

We obtain the following conclusion: Let $G = (V, E)$ a connected, undirected graph. Denote by c_{ij} the commute distance between vertex v_i and vertex v_j, and by $L^\dagger = (l^\dagger_{ij})_{i,j=1,\dots,n}$ the generalized inverse of L. Then we have:

$$c_{ij} = vol(V)(l^\dagger_{ii} - 2l^\dagger_{ij} + l^\dagger_{jj}) = vol(V)(e_i - e_j)' L^\dagger (e_i - e_j). \tag{14}$$

This conclusion leads to an important consequence. It shows that $\sqrt{c_{ij}}$ can be considered as a Euclidean distance function on the vertices of the graph. This means that we can construct an embedding which maps the vertices v_i of the graph on points $z_i \in \mathbb{R}^n$ such that the Euclidean distances between the points z_i coincide with the commute distances on the graph.

7 Berkeley Edge Detector

In this section we describe the Berkeley edge detector which has two stages. The first stage is local and couples multi-scale local brightness, color, and texture cues. The second stage is global and uses spectral clustering. This is based on the paper "Contour Detection and Hierarchical Image Segmentation"[1].

7.1 Building Blocks

The basic ideas of the contour detector is that the intensity properties are different on the two sides of an edge. To detect an edge at (x, y) at orientation θ we split a circular neighborhood of radius σ centered at (x, y) into two semi-circles, then we compute the histograms g and h of the filter responses (see below) in the semi-circles, and then compute a measure of difference by the χ^2 distance:

$$\chi^2(g, h) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)} \tag{15}$$

Figure 4 shows an example.

The features used are texture and CIE Lab colorspace (brightness, color a and color b). For texture channel, first the image is converted to grayscale and convolved with 8×2 Gabor filters. Each pixel is associated with a 17-dimensional vector of responses. These vectors are then clustered using K-means and each pixel is assigned to the closest cluster center. The histograms for texture channel is the frequency of the occurrence of each clusters. For color channel, each color component is quantized into 25 and three histograms are computed.

7.2 Multiscale Cue Combiniation

In order to detect fine as well as coarse structures, gradients at three scales are considered $[\sigma/2, \sigma, 2\sigma]$ for each of the brightness, color, and texture channels. Figure 5 shows an example of the oriented gradients

[1] `http://www.cs.berkeley.edu/~malik/papers/arbelaezMFM-pami2010.pdf`

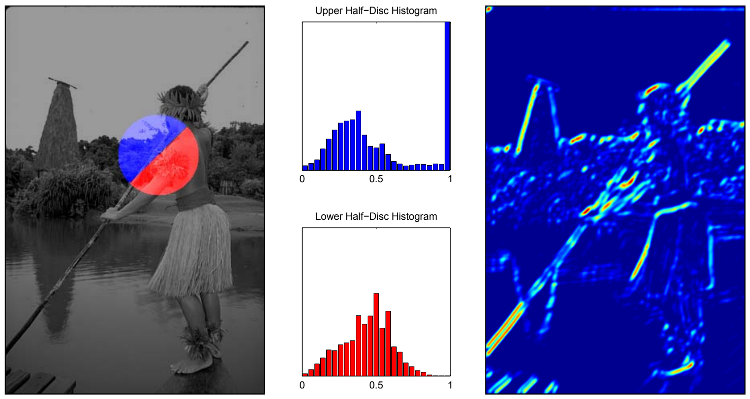


Figure 4: **Oriented gradient of histograms**. Given an intensity image, consider a circular disc centered at each pixel and split by a diameter at angle θ. We compute histograms of intensity values in each half-disc and output the χ^2 distance between them as the gradient magnitude. The blue and red distributions shown in the middle panel are the histograms of the pixel brightness values in the blue and red regions, respectively, in the left image. The right panel shows an example result for a disc of radius 5 pixels at orientation $\theta = \pi$. Note that the left panel displays a larger disc (radius 50 pixels) for illustrative purposes.

obtained for each channel. For the brightness channel, we use $\sigma = 5$ pixels, while for color and texture we use $\sigma = 10$ pixels. We then linearly combine these local cues into a single multiscale oriented signal:

$$mPb(x, y, \theta) = \sum_s \sum_i \alpha_{i,s} G_{i,\sigma(i,s)}(x, y, \theta) \tag{16}$$

where s indexes scales, i indexes feature channels and $G_{i,\sigma(i,s)}(x, y, \theta)$ measures the histogram difference in channel i between two halves of a disc of radius $\sigma(i, s)$ centered at (x, y) and divided by a diameter at angle θ. The parameters $\alpha_{i,s}$ weight the relative contribution of each gradient signal. Taking the maximum response over orientations yields a measure of boundary strength at each pixel:

$$mPb(x, y) = \max_\theta \{mPb(x, y, \theta)\} \tag{17}$$

7.3 Globalization

As input to the spectral clustering stage, a sparse symmetric affinity matrix W is constructed using the maximal value of mPb along a line connecting two pixels. All pixels i and j within a fixed radius r are connected with affinity:

$$W_{ij} = \exp\left(-\max_{p \in \overline{ij}} \{mPb(p)\}/\rho\right) \tag{18}$$

where $\overline{ij}$ is the line segment connecting i and j and ρ is a constant.

In order to introduce global information, we define $D_{ii} = \sum_j W_{ij}$ and solve for the generalized eigenvectors $\{v_0, \ldots, v_n\}$ for $(D - W)v = \lambda Dv$, where $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_n$. Figure 6 displays an example with four eigenvectors.

The eigenvectors themselves carry contour information. Treating each eigenvector v_k as an image, we convolve with Gaussian directional derivative filters at multiple orientations θ, obtaining oriented signals $\nabla_\theta v_k(x, y)$. Taking derivatives in this manner ignores the smooth variations that previously lead to errors (middle right in Figure 6). The information from different eigenvectors is then combined to provide the

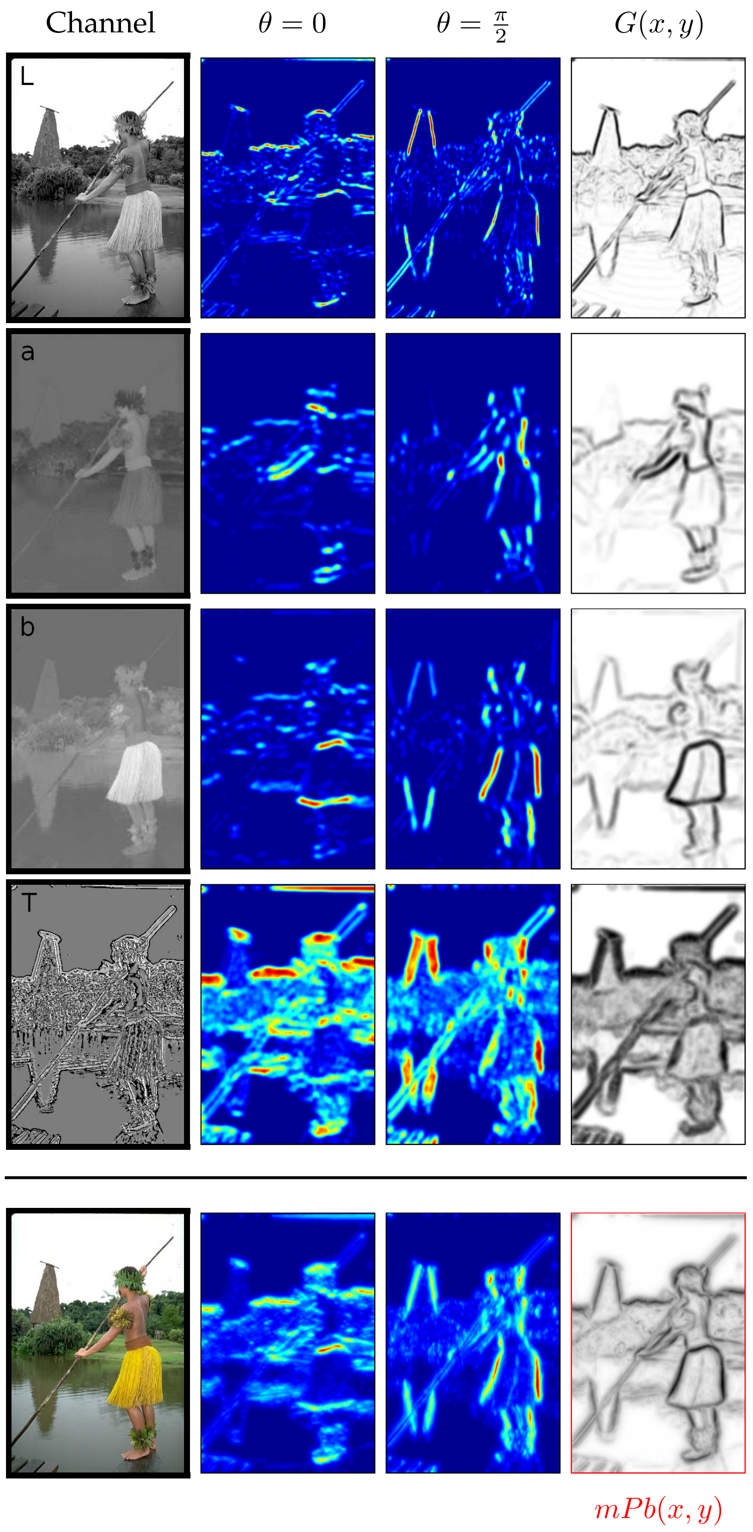


Figure 5: **Multiscale Pb. Left Column, Top to Bottom:** The brightness and color a and b channels of Lab color space, and the texton channel computed using image-specific textons, followed by the input image. Rows: Next to each channel, we display the oriented gradient of histograms (as outlined in Figure 4) for $\theta = 0$ and $\theta = \pi/2$ (horizontal and vertical), and the maximum response over eight orientations in $[0, \pi)$ (right column). Beside the original image, we display the combination of oriented gradients across all four channels and across three scales. The lower right panel (outlined in red) shows mPb, the final output of the multiscale contour detector.

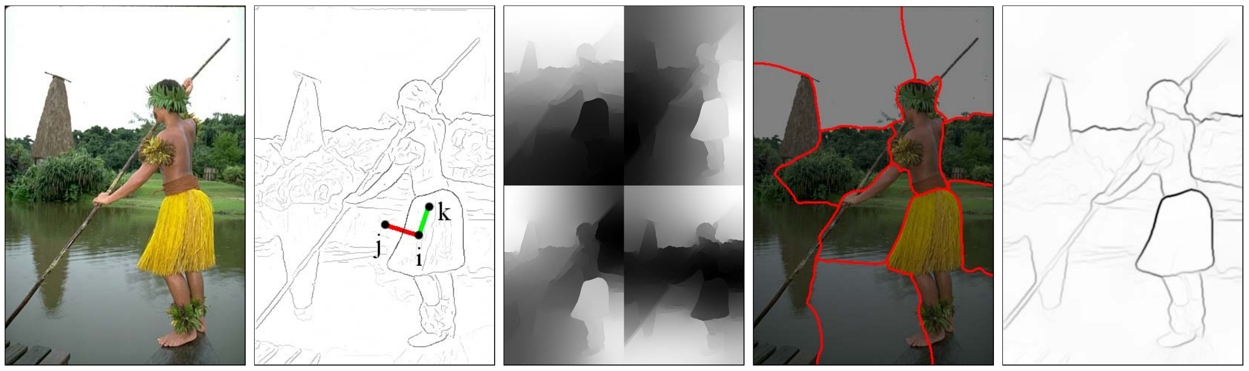


Figure 6: **Spectral Pb. Left:** Image. **Middle Left:** The thinned non-max suppressed multiscale Pb signal defines a sparse affinity matrix connecting pixels within a fixed radius. Pixels i and j have a low affinity as a strong boundary separates them, whereas i and k have high affinity. **Middle:** First four generalized eigenvectors resulting from spectral clustering. **Middle Right:** Partitioning the image by running K-means clustering on the eigenvectors erroneously breaks smooth regions. **Right:** Instead, we compute gradients of the eigenvectors, transforming them back into a contour signal.

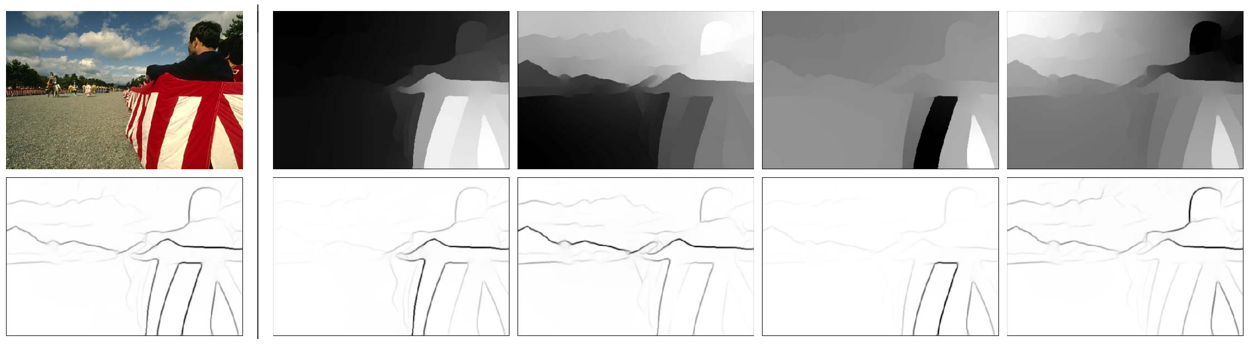


Figure 7: Eigenvectors carry contour information. Left: Image and maximum response of spectral Pb over orientations, $sPb(x,y) = max_\theta sPb(x,y,\theta)$. Right Top: First four generalized eigenvectors, $v1, ..., v4$, used in creating sPb. Right Bottom: Maximum gradient response over orientations, $max_\theta \nabla_\theta v_k(x,y)$, for each eigenvector.

"spectral" component of the boundary detector:

$$sPb(x, y, \theta) = \sum_{k=1}^{n} \frac{1}{\sqrt{\lambda_k}} |\nabla_\theta v_k(x, y)| \tag{19}$$

Figures 6 and 7 present examples of the eigenvectors, their directional derivatives, and the resulting sPb signal.

Note that this differs from how the eigenvectors v_k are used in the standard spectral clustering algorithm, described in the earlier sections, which uses a threshold to determine which eigenvalues to use and then uses their eigenvectors as indicator functions to determine the segmented regions. In practice, finding a good threshold is difficult. Instead observe that if the v_k act as indicator functions for the regions then their gradients will be large at the edges (e.g., the indicator function takes value 1 in one region and is 0 is all the others). To avoid picking a threshold then instead they sum the gradients of the eigenvectors and weight them inversely by the square root of the eigenvalue, so that small eigenvalues contribute most.

Our final globalized probability of boundary is then written as a weighted sum of local and spectral signals:

$$gPb(x, y, \theta) = \sum_s \sum_i \beta_{i,s} G_{i,\sigma(i,s)}(x, y, \theta) + \gamma \cdot sPb(x, y, \theta) \tag{20}$$