CS 280, Computer Vision (Prof. Jitendra Malik) April 21, 1999
Lecture #24 (Grouping, Segmentation, and Normalized Cuts) **DRAFT** Notes by Steve Ross

---

- Administrivia

  May 10th Final Project Presentation

  May 20th Final (3 hours, 2 single sided pages of notes)

- Factors which lead to grouping

  Similarity of (brightness, color, texture, disparity, motion)

  Proximity

  Good continuation of boundary contours

  Closure

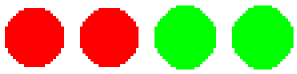
  Symmetry and parallelism

  Familiar Configuration

- Normalized Cuts

1 Factors which lead to grouping

1.1 Similarity of (brightness, color, texture, disparity, motion)



Most people will perceive two groups, a group of red dots, and a group of green dots. This is similarity in brightness. Similarity in motion, bird in the bush example for motion, group points that move together.

1.2 Proximity

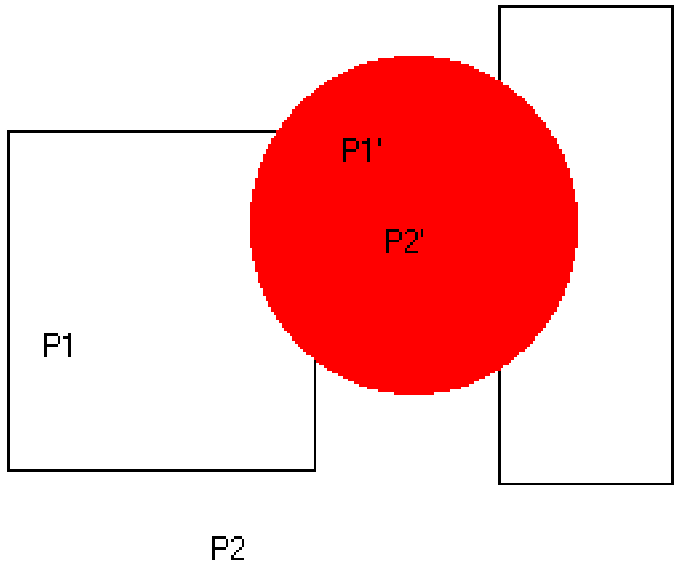
Why do we perceive this as two separate groups of black dots? The insight for this and other factors which, lead to grouping, comes from ecological statistics. In Computer Vision, we are given an image and trying to deduce the world that has highest probability of giving rise to that image. This can be considered as a Bayesian view, find the most probable interpretation in the context of prior world knowledge. Recall from statistics, the probability of dependent events. Applied to the problem of Computer Vision this is:

$$P(World|Image) = \frac{P(Image|World)P(World)}{P(Image)}$$

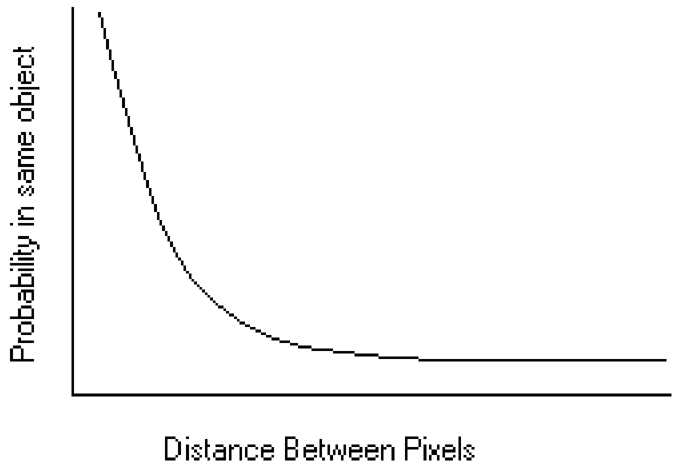
So given several probabilities:

$$\begin{aligned} P & (World_1|Image) \\ P & (World_2|Image) \\ & \vdots \\ P & (World_n|Image) \end{aligned}$$

We would pick the one with the highest probability as the solution.
So, for proximity consider the following image:



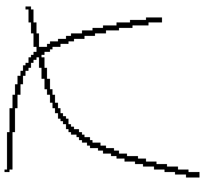
The question to ask is, given any two pixels in an image, how often are they pixels in the same object vs. being pixels in different objects. So we are trying to find P(P1 and P2 belong to the same object) as a function of the distance between P1 and P2. If you picked P1 and P2 separated by a distance of 1 pixel you would probably find they are part of the same object 95% of the time. If you increase the distance between them to 100 pixels you would probably find that they are part of different objects 95% of the time. In fact, the curve plotting the probability as a function of distance between pixels would probably look something like this:



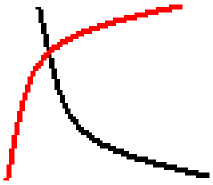
This is just a rule of thumb developed by psychologists to describe the phenomenon of proximity. This could be quantified by computing this on a large set of images, say 100,000. The reason this has yet to be done is that it is difficult to accurately segment images into separate objects. This is kind of a chicken and the egg problem as to develop more accurate segmentation algorithms researchers would probably need to make use of these statistics.

1.3 Good continuation of boundary contours

Given this image which two segments would we perceive ?

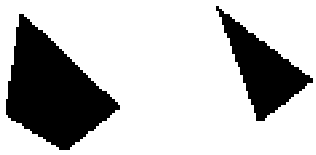


Most people would probably perceive the following two segments

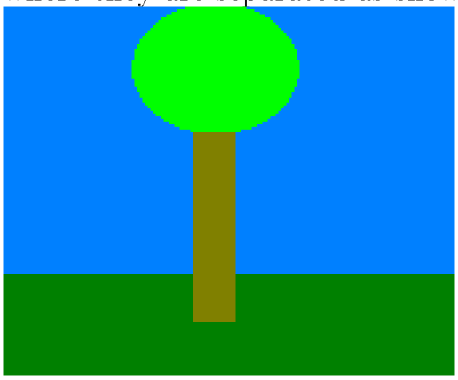


The justification for this is that most objects in the natural world have smooth continuous boundaries. Objects in our world do have sharp corners, but the probability of an object being smooth is much higher. The question to ask is how often is the tangent of a contour continuous vs. discontinuous. Again this is a statistical justification.

1.4 Closure

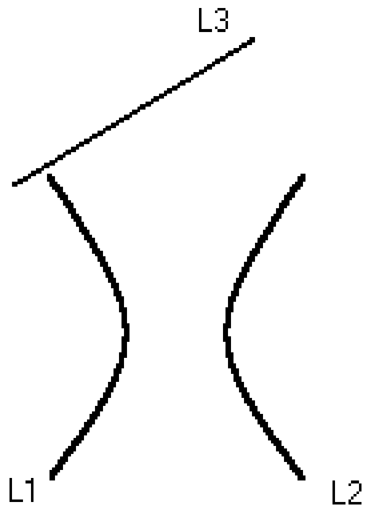


Most objects in the world appear to be enclosed in their contour. However, there are cases where they are separated as shown by the sky in the following image:



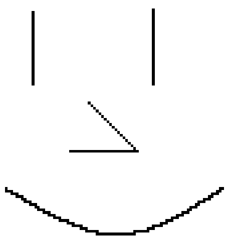
But again, the probability of the object being in enclosed by its contour is higher and hence we use closure as a factor for grouping.

1.5 Symmetry and parallelism



Most people would perceive that L1 and L2 are part of the same object due to symmetry. Again this is an ecological statistic as most objects in the natural world exhibit symmetry. The two types of symmetry seen most often are Rotational Symmetry (symmetry around an axis like a limb, i.e. your arm) and Bilateral Symmetry (i.e., the left and right sides of your body) Recall from previous lectures that we have seen that objects in the world that have rotational symmetry will project to objects in the image that have bilateral symmetry.

1.6 Familiar Configuration



Most people would interpret this as a face. This is due to high level knowledge that humans use. An interesting psychological study showed that face recognition is probably genetically encoded, as babies prefer to look at faces as early as 15 minutes after birth. Recognition of other objects in the world, like a 99 Ford Mustang, is a learned trait. The interesting question is how much is learned, and how much is hard coded at birth. Thus it is the nature vs. nurture debate seen again.

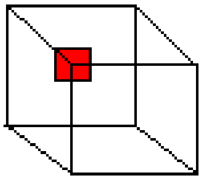
1.7 Summary

Ecological statistics provide a correlation that is true probabilistically and are a useful tool in grouping and segmentation. Ecological Statistics will probably be as useful a tool in Computer Vision as Ecological Optics, which we studied in the first part of this course, and led to the geometric basis and mathematical algorithms.

2 Segmentation using Normalized Cuts

2.1 Segmentation

Edge detection is a local technique for segmentation that identifies changes in brightness that constitutes edges. As you probably noticed from implementing the Canny edge detector it makes mistakes/has its limitations. The basic problem is that it is a local approach. It looks at the image through a small hole and it is much harder to interpret images locally. For example, if you just look at a corner, its hard to tell if its concave or convex through a small window.



Similarly, its hard to tell the direction an object is moving in a small window, this is the aperture problem. For segmentation using local algorithms, its hard to tell if a change in brightness is due to a texture element, a real edge, or noise. Locally a small window doesn't have enough information. Normalized cuts takes a global approach to segmentation, looking at the entire image.

2.2 Normalized Cuts

In the Normalized Cuts algorithm, image segmentation is treated as a graph-partitioning problem. Build a weighted Graph:

$$\begin{aligned} G &= (V, E) \\ V &: \text{pixels in the image} \\ E &: \text{connections between pairs of neighboring pixels} \\ W(i,j) &: \text{probability that pixels i and j belong to the same region} \end{aligned}$$

One way to think of it is as the pixels being point masses with either strong or weak connections and see what shakes together. The difference between this and local edge detection is that it is a global approach looking at the entire image. By using all image information the algorithm is able to produce better results. W(i,j) are calculated from cues based on local measurements such as change in brightness. Adding continuity of contour to brightness information is a low-level cue, which helps to improve performance.

- If connection is weak - there is an edge
- If connection is strong - no edge

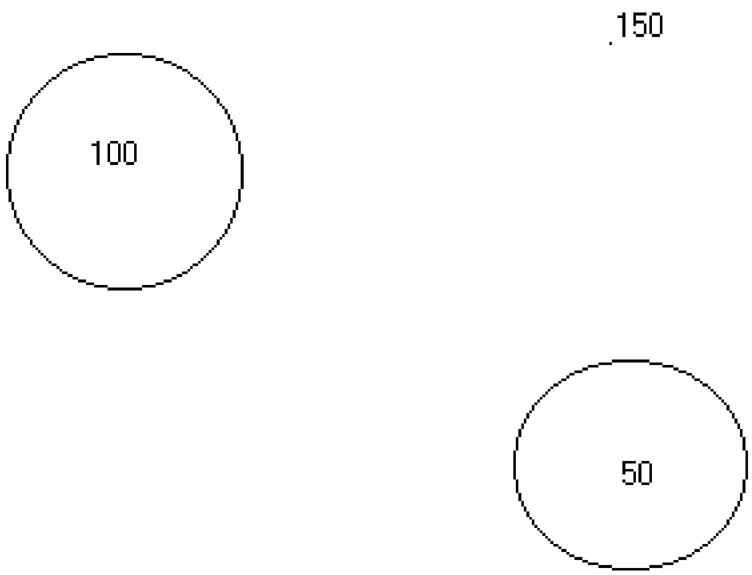
A threshold is used to determine how many parts to partition the graph, which represents the image. Consider this room and grouping at different levels of hierarchy.

- Students can appear as one group, a crowd separated from the walls
- Or students can be considered individually
- Or each student's face can be considered separate from his or her body

Thus, segmentation is Hierarchical. How do we bipartition a graph. Some terms from graph partitioning.

$$Cut(A,B) = \sum W(u,v) u \in A, v \in B with A \cap B = \{\}$$

One intuition is to find the minimum cut, it turns out that this a bad idea. Consider this image:



The ideal cut would segment the image separating the circle with brightness of 100 from the circle with brightness of 50. The minimum cut that bisects the graph is just the noise pixel with the value of 150 and the rest of the image. This leads to the normalized cut. Adding a normalizing factor that penalizes partitions that cut groups normalizes the cut. The normalizing factor is defined as the association between pixels.

$$Association(A, A') = \sum W(u, v) u \in A, v \in A'$$

The normalized cut is:

$$Ncut(A, B) = \frac{Cut(A, B)}{Assoc(A, V)} + \frac{Cut(A, B)}{Assoc(B, V)}$$

Normalizing the cut helps to solve the problem of segmenting out small groups of points that are probably due to noise. To minimize a normalized cut is a NP complete problem, so instead we construct an approximation. Let W be the association Matrix:

$$W(i, j) = Wi, j$$

Let D be the diag Matrix:

$$D(i, i) = \sum jW(i, j)$$

Let x be a vector in $\{1, -1\}^N$
R^N where N is the number of Pixels
x is an indicator vector, solution is encoded in this vector x

$$\begin{array}{rcl} 1 & : & \text{for all pixels in A} \\ -1 & : & \text{for all pixels in B} \end{array}$$

Spectral graph bisection, connects graphs to eigenvalues. To partition the graph you just need to find the eigenvalues. But even for small images, the matrices are big. For a 100x100 image i is 10,000 and j is 10,000. So W(i,j) is a 10,000 x 10,000 matrix. Luckily it turns out that the matrix is sparse. So with lots of algebra normalized cuts can elegantly be rewritten as a generalized eigenvalue problem. After simplification, the normalized cut formula becomes

$$min_x Ncut(x) = min_y \frac{y^T(D - W)y}{y^T D y} \; with \; y_i \in \{1, -b\}, y^T D_1 = 0$$

We know D and W so we just need to solve for the y vector that minimized the normalized cut. This is a Rayleigh Quotient and can be minimized by solving the generalized eigenvalue system. The eigenvectors in the solution y are real valued, so eigenvectors are just an approximation and need to be discretized.
The analog here is between Linear Programming and Integer Programming. Minimizing/Maximizing an objective function using Linear Programming can be done in polynomial

time. To do it in Integer Programming is NP hard, so we use the heuristic of solving it as a Linear Programming problem and discretizing the solution. Recall that an eigenvector is:

$$Ax = \lambda x$$
$$\text{x: is the eigenvector}$$
$$\lambda : \text{is the associated eigenvalue}$$

So for normalized cuts the eigenvalue problem is:

$$(D - W)y = \lambda Dy$$

The solution is the eigenvector with the second smallest eigenvalue of the generalized eigensystem.
Another way to interpret normalized cuts is as a Dynamical System. You could think of it as a spring mass system.

$$LaplacianMatrix : (D - W) => \text{stiffness matrix k}$$

$$DiagMatrix : D => \text{mass Matrix M}$$

So the equivalent spring-mass system equation for the problem can be written as:

$$Mx''(t) = -kx(t)$$

The generalized eigenvectors are the fundamental modes of oscillation. Objects with tight connections oscillate together as the system is shaken.

- For more detail on normalized cuts see: http://www.cs.berkeley.edu/~jshi/Grouping/
- For more detail on graph bisection see: http://www.cs.berkeley.edu/~demmel/cs267_Spr99/ lecs:14-16