Image Segmentation Using Normalized Cuts

A graph theoretical approach to image segmentation

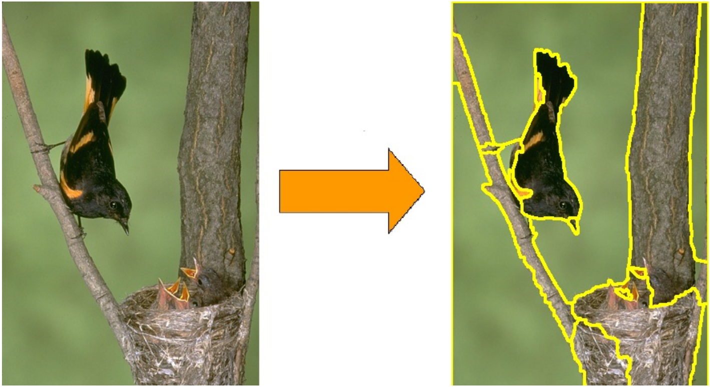
Manuel Pagliai, Lorenzo Sarti

May 23, 2013

Outline

The target

- Given an image I, the goal is to recognize the objects which compose the particular scene.
- This is a natural task for humans but it is not trivial for a computer.
- Image segmentation is the process of dividing an image into multiple parts and represents the basic step of the recognition process.

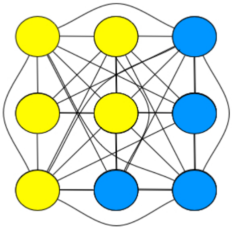


- Various approaches have been proposed and there is a wide literature on this subject.
- This work focuses on graph partitioning and spectral grouping methods (Jianbo Shi, Jitendra Malik 2000).
- We also made a comparison to other well known techniques.

- Partitioning algorithms model the images as graphs.
- The graph is then partitioned according to a criterion designed to achieve a good segmentation level.
- Each partition of the graph is considered as an object in the image.

- The image is represented as a completely connected weighted graph where each node represents a pixel or a connected patch in the image.

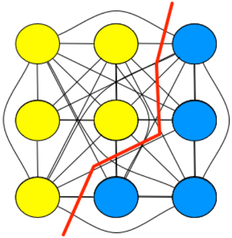
Weights on edges are assigned according to some measure of similarity.



- To achieve sparsity we can set the weight on an edge equal to zero if its value is beyond a given threshold.

- Given a graph $G = (V, E)$, V can be partitioned in two disjoint sets such that A, B: $A \cap B = \emptyset$, $A \cup B = V$ For this purpose, edges that connect nodes of A to nodes of B should be removed.

The degree of dissimilarity between A and B can be computed as a total weight of the edges that have been removed (cut):
$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$.



- The optimal bipartition of a graph is the one minimising this cut value.
- There are efficient algorithms for finding the minimum cut of a graph.
- However the minimum cut criteria is biased towards partitions containing few isolated nodes.

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

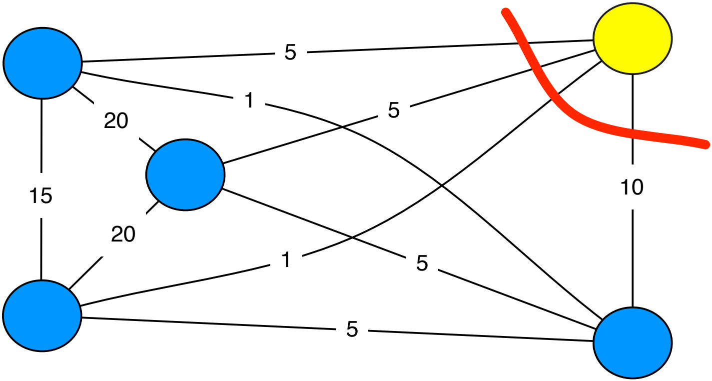


Figure : An example where the cut measure gives a biased result (edge weights are inversely proportional to the distance between the two nodes).

- Avoiding unnatural bias towards small partitions through normalized cuts .

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{1}$$

where:

$$assoc(A, V) = \sum_{a \in A, v \in V} w(a, v) \tag{2}$$

- A measure for association within groups for a given partition

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \qquad (3)$$

- Association and disassociation are naturally related:

$$Ncut(A,B) = 2 - Nassoc(A,B) \qquad (4)$$

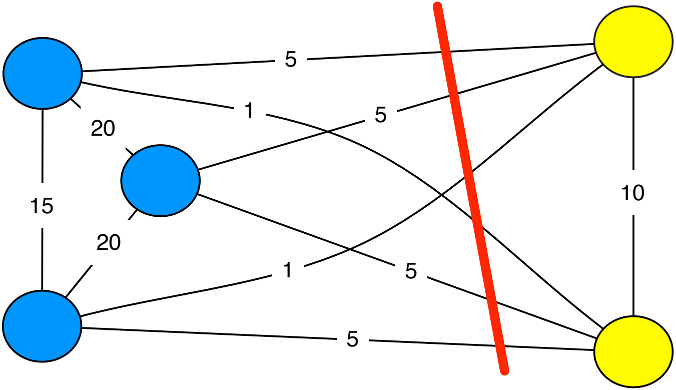
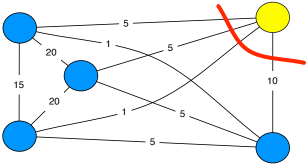


Figure : An example where the normalized cut measure gives the expected result (edge weights are inversely proportional to the distance between the two nodes).

A simple example

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

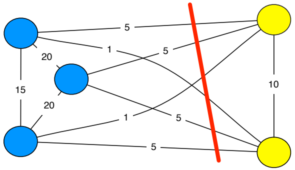
$$\begin{aligned} nCut &= \frac{5+5+10+1}{5+5+1+10} \\ &+ \frac{5+5+1+10}{5+5+1+10+(20+20+1+5+5+15)\cdot 2} = \\ &= 1 + 0.13 = 1.13 \end{aligned}$$



A simple example(Cont.)

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$\begin{aligned} nCut &= \frac{5+5+1+1+5+5}{5+5+1+1+5+(15+20+20)\cdot 2} \\ &+ \frac{5+5+1+1+5+5}{5+5+1+1+5+5+10+10} = \\ &= 0.17 + 0.52 = 0.69 \end{aligned}$$



- This time the cut is indeed gives the expected result.

- The algorithm uses the minimization of the normalized cut value as partition criterion.
- Unfortunately, solving the minimization problem is NP-complete (Papadimitrou '97)
- Need to introduce approximation for solving the problem efficiently.

- Let $d(i) = \sum_j w(i,j)$ and x be a $N = |V|$ dimensional vector with $x_i = 1$ if node i belongs to A, -1 otherwise. Then:

$$
\begin{aligned}
Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} = \\
&= \frac{\sum_{x_i>0,x_j<0} -w_{ij}x_ix_j}{\sum_{x_i>0} d(i)} + \frac{\sum_{x_i<0,x_j>0} -w_{ij}x_ix_j}{\sum_{x_i<0} d(i)}
\end{aligned} \tag{5}
$$

- Let W a $N \times N$ matrix with $W(i,j) = w_{ij}$, D a $N \times N$ diagonal matrix with d on its diagonal and $k = \frac{\sum_{x_i > 0} d_i}{\sum_i d_i}$. We can now rewrite the value of $Ncut(x)$ as:

$$\frac{(\mathbb{1} + x)^T (D - W)(\mathbb{1} + x)}{k \mathbb{1}^T D \mathbb{1}} + \frac{(\mathbb{1} - x)^T (D - W)(\mathbb{1} - x)}{(1 - k) \mathbb{1}^T D \mathbb{1}} \tag{6}$$

where $\mathbb{1}$ is the $N \times 1$ vector with all components set to 1

- By Setting $b = \frac{k}{1-k}$ and $y = (\mathbb{1} + x) - b(\mathbb{1} - x)$ after some mathematical calculations we obtain the following equation:

$$min_x \; Ncut(x) = min_y \frac{y^T(D-W)y}{y^T DY} \tag{7}$$

s.t. $y^T D \; \mathbb{1} \; = 0$ and $y(i) \in \{2, -2b\}$

- Given that the following properties hold

  1. Equation (7) can be seen as a Rayleigh ratio with respect to the matrix $\mathcal{A} = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$
  2. The matrix $(D - W)$ is singular

- If y is relaxed to take on real values the solution of the minimization problem is given by the eigenvector γ associated to the second smallest eigenvalue of the generalized eigenvalue system:

$$(D - W)y = \lambda Dy \tag{8}$$

- It can be shown that the constrain $y^T D \mathbb{1} = 0$ is automatically satisfied by γ.
- Relaxing y by letting it take real values (rather than two single values) is the approximation that makes the problem tractable.

- In order to perform a bipartition of the image, the values of the eigenvector γ need to be discretised.

- The vector γ can be discretised according to a pivot value. Following are some of the methods available:

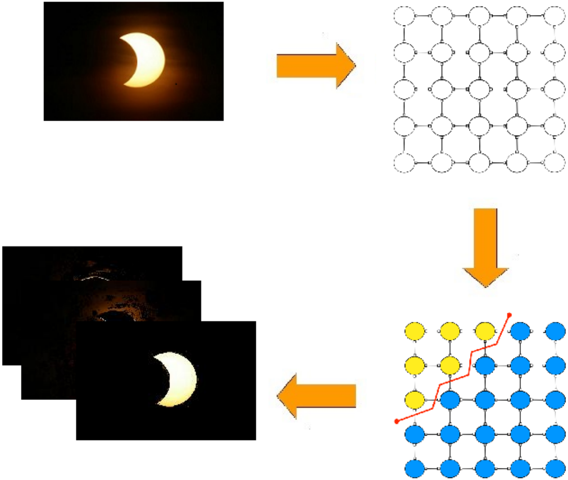
  - Use value 0

  - Use the median

  - Minimise the cost function

- The above algorithm can be applied to image segmentation through the following steps:

1. Given an image I, set up a weighted graph $G = (V, E)$ using as weights some measure of similarity between pixels
2. Find the eigenvector associated to the second smallest eigenvalue solving the system $(D - W)y = \lambda Dy$
3. Discretize the value of the computed solution in order to split the graph.
4. Use a stopping criterion to decide whether or not the segment should be recursively partitioned.



- Weight function should reflect the likelihood of two pixels belonging to the same object
- There are many affinity measures that can be considered:
  - Affinity by distance
  - Affinity by brightness
  - Affinity by colour
  - Affinity by texture
- This work focuses on affinity by distance and colour.

- By defining $F(i)$ and $X(i)$ the brightness and the position of pixel i respectively, the following formula accounts for pixel similarity:

$$w_{ij} = e^{-\frac{\|F(i)-F(j)\|^2}{\sigma_I^2}} \cdot \begin{cases} e^{-\frac{\|X(i)-X(j)\|^2}{\sigma_X^2}} & \text{if } \|X(i)-X(j)\|_2 < r \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Given σ_I and σ_X parameters controlling the weight of brightness and proximity respectively.

- W is a matrix of $N \times N$ where
  $N = \textit{image width} \cdot \textit{ image height}$

- W is sparse as well as the matrices $D - W$ and D and thus the matrix computation require $O(N)$.

- We can take advantage of well known techniques for representing sparse matrices as well as methods for computing eigenvectors (e.g. Lanczos method).

- In this work we choose the pivot as the value minimizing the value of the cut (5)
- After the graph is partitioned in two regions we can either recursively run the algorithm on the two parts, or we can take advantage of the properties of the other computed eigenvectors.
- It can be shown that the eigenvector associated to the third smallest eigenvalue is the real valued solution that optimally subpartitions the first two parts.

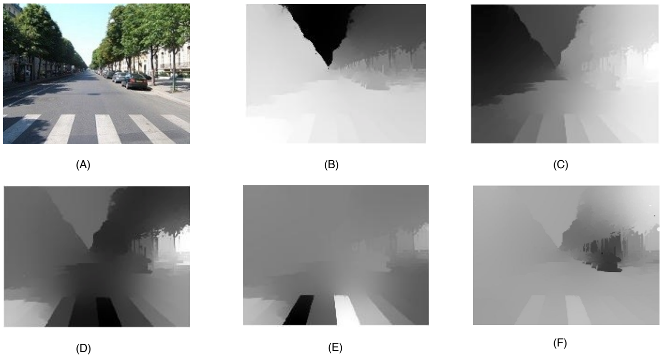


Figure : (A) is the original image, (B)-(F) are the first five eigenvectors displayed as grayscale images

- This observation can be extended to show that it is possible to recursively partition the resulting graph , each time using the eigenvector with the next smallest eigenvalue.

- This approach has a stability problem and it is better to recursively run the algorithm using the second smallest eigenvector .

- Another possible approach, belonging to the spectral grouping algorithm family, consists in computing the first m solution with the smallest eigenvalues of the system $(D - W)y = \lambda Dy$ and using their values as features of a $m - dimensional$ space.

- As a second step we perform a $K - means$ clustering algorithm on the feature space.

- The clusters provided by $K - means$ algorithm give the image segments.

- We implemented the normalized cuts algorithm and the spectral k-means using matlab.

- Since we experienced numerical problems, the matrix D is perturbed by adding a small quantity on the diagonal before computing the eigenvectors.
- Rather then computing the second eigenvector only, we compute the first m eigenvectors.
    - One single eigenvector γ is used for partitioning
    - γ is the one minimising the cut value among the first m eigenvectors.

- The algorithm follows these steps:
  1. The image is optionally represented using a YUV colour space
  2. The $N \times N$ matrix W is computed where $w_{i,j}$ represents the affinity between pixel i and pixel j
  3. The $N \times N$ diagonal matrix D is computed where $D_{i,i} = \sum_j W_{i,j}$
  4. The generalised eigensystem $(D - W)y = \lambda Dy$ is solved to compute the first m eigenvectors.
  5. The splitting vector is chosen among the first m discretised eigenvectors, as the one that minimises the normalized cut value.
  6. The procedure is called recursively on both sub-partitions according to a stopping criterion (*e.g.* the cut value is beyond a given threshold).

- By carefully tuning the parameters our methods show good performances.
- There is a small partitioning issue caused by the approximation of the eigenvector, removable by a post-processing operation.
- Our methods are computational expensive, mainly due to the computation of the matrix W.

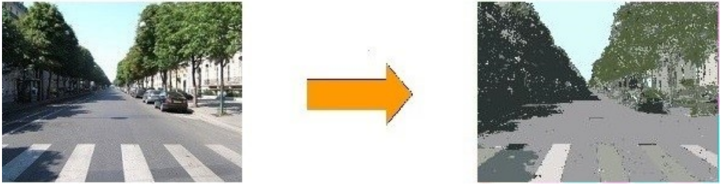


Figure : Segmentation using normalized cuts

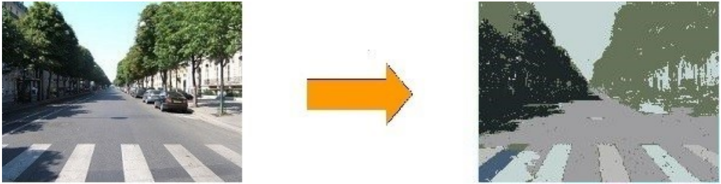


Figure : Segmentation using the spectral grouping algorithm

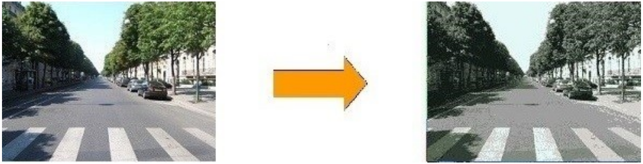


Figure : Segmentation using k-means clustering

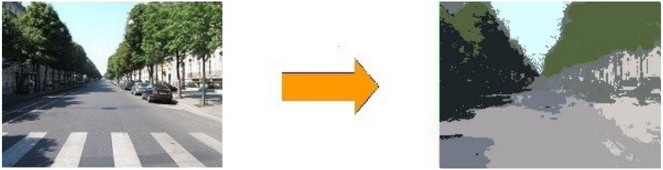


Figure : Segmentation using gaussian mixture model

- We achieve good segmentation level on several test images.
- Future developments:
  - Speed up the process using approximate eigenvectors evaluation
  - Remove noise from extracted regions using a filtering operation.