اصول پردازش تصویر

Principles of Image Processing

مصطفی کمالی تبریزی

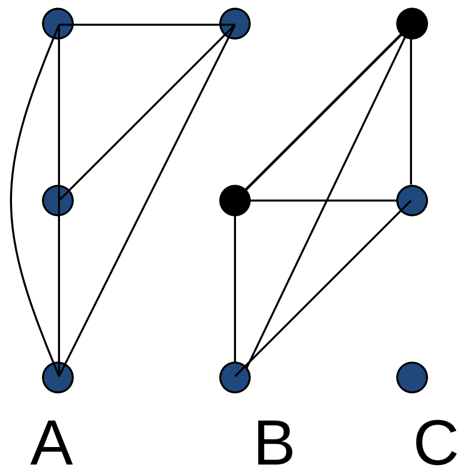
۱۷ آبان ۱۳۹۹

جلسه چهاردهم

Normalized Cuts for Image Segmentation

Jianbo Shi and Jitendra Malik
Normalized cuts and image segmentation
Pattern Analysis and Machine Intelligence (PAMI), 2000

Segmentation by Graph Partitioning



Graph: $G = (V, E)$

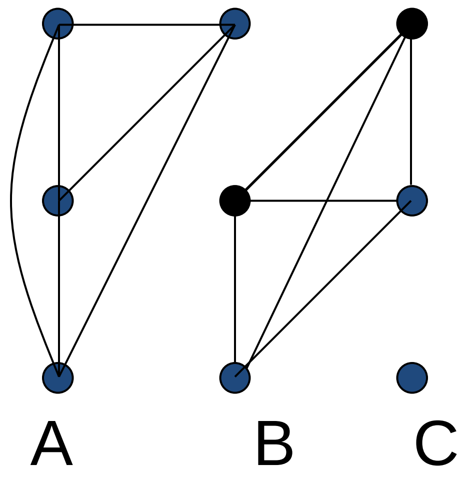
Nodes: All pixels of the image

Edges: Between every pair of nodes

Weights: Representing similarity between two nodes

Source: S. Seitz

Segmentation by Graph Partitioning



- Break Graph into Segments
  - Delete links that cross between segments
  - Easiest to break links that have low affinity
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

Source: S. Seitz

Measuring Affinity

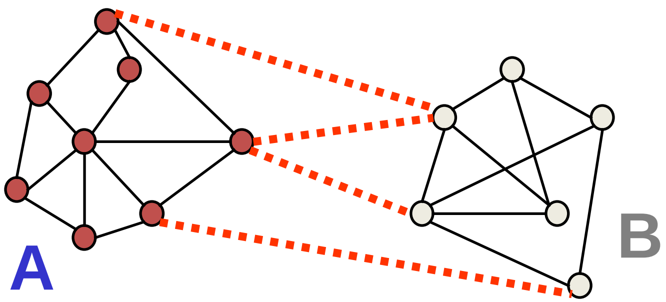
- Suppose we represent each pixel by a feature vector x, and define a distance function appropriate for this feature representation

- Then we can convert the distance between two feature vectors into an affinity with the help of a generalized Gaussian kernel:

$$e^{-\frac{dist(\mathrm{x}_i,\mathrm{x}_j)^2}{2\sigma^2}}$$

Slide Credit: S. Lazebnik

Graph Cut

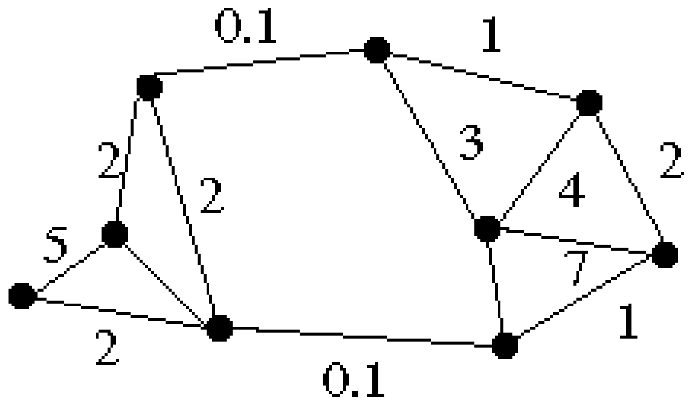


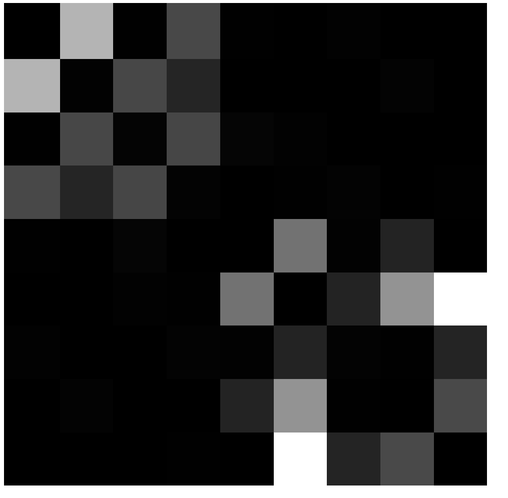
- Cut: Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a segmentation
  – What is a “good” graph cut and how do we find one?

Source: S. Seitz

Minimum Cut

- We can do segmentation by finding the *minimum cut* in a graph
  - Efficient algorithms exist for doing this

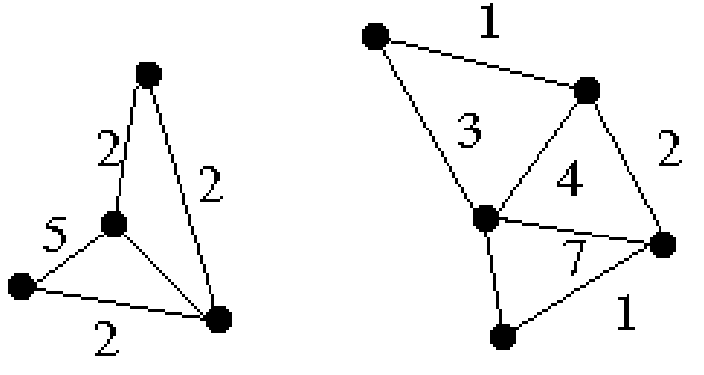


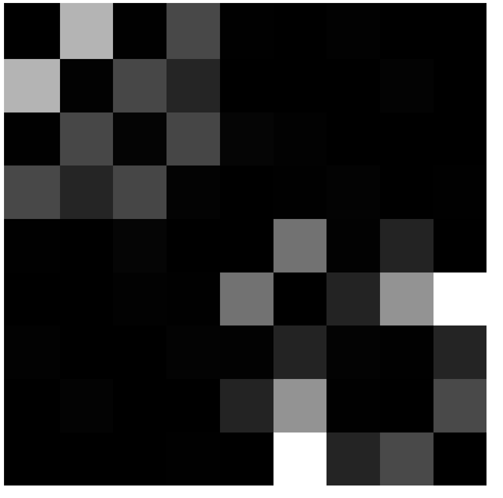


Slide Credit: S. Lazebnik

Minimum Cut

- We can do segmentation by finding the *minimum cut* in a graph
  - Efficient algorithms exist for doing this



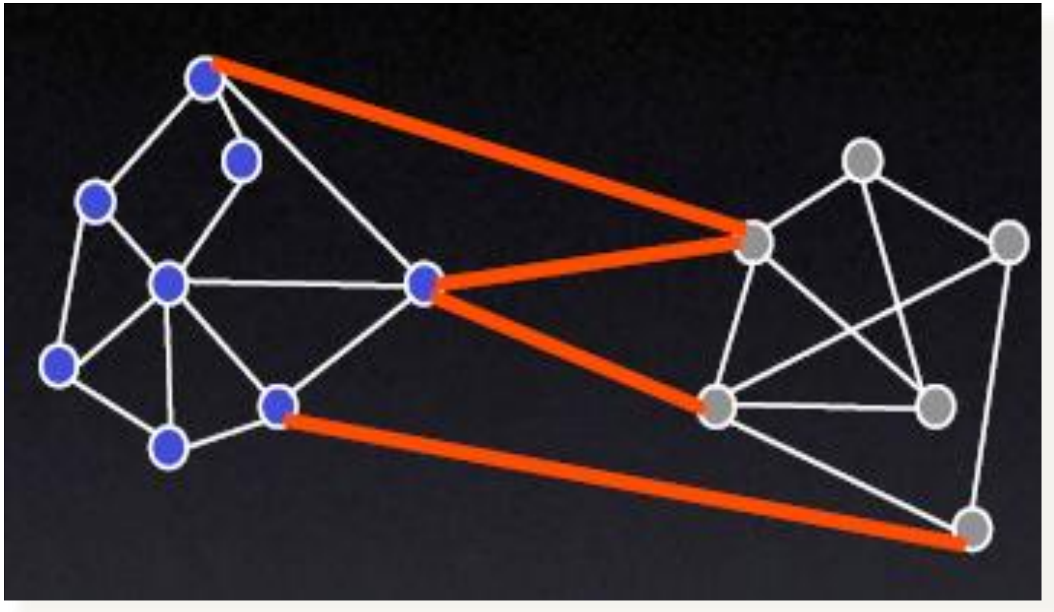


Slide Credit: S. Lazebnik

Minimum Cut Algorithm

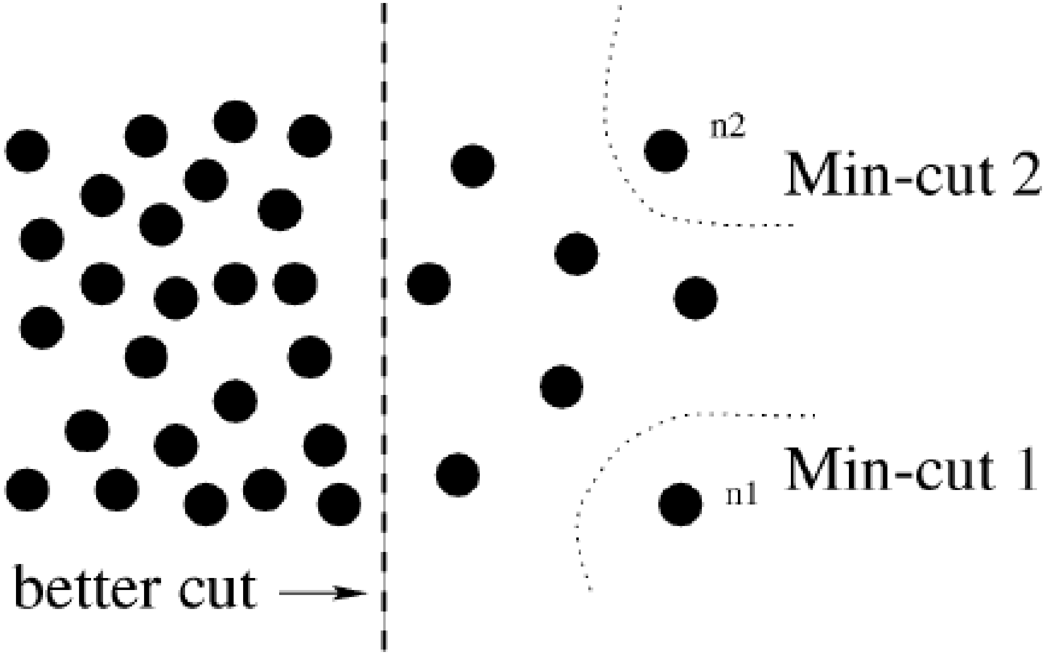
$$Cut(A,B) = \sum_{u \in A, v \in B} W(u,v)$$

$Cut(A,B)$ is a measure of similarity between the two groups.



The advantage of Min Cut:
Well-studied problem and efficient algorithms

The disadvantage of Min Cut:
Favors cutting small sets of isolated nodes in the graph



- Graph: $G = (V, E)$
  - Nodes: All pixels of the image
  - Edges: Between every pair of nodes
  - Weights: Representing similarity between two nodes

- Objective:
  - Partition V into two disjoint sets A and B.
  - Minimize similarity between two sets:

$$Cut(A, B) = \sum_{u \in A, v \in B} W(u, v)$$

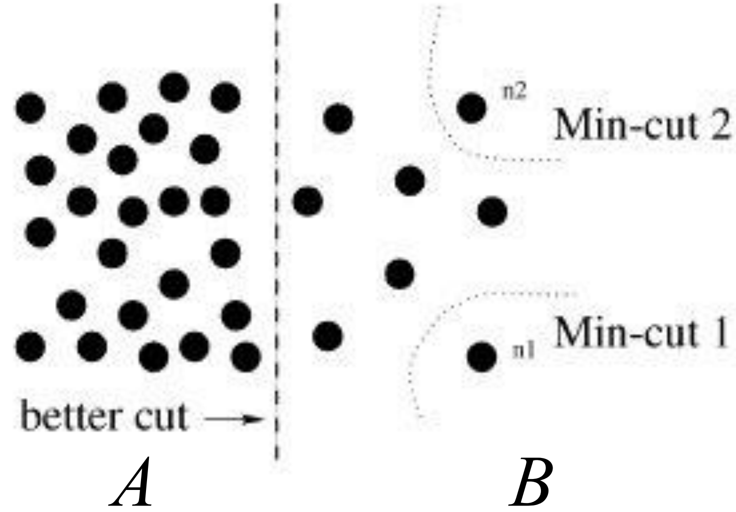
  - Maximize similarity within each set:

$$Assoc(A, A) = \sum_{u \in A, v \in A} W(u, v)$$

Normalized Cut

$$Cut(A,B) = \sum_{u \in A, v \in B} W(u,v)$$

$$Assoc(A,A) = \sum_{u \in A, v \in A} W(u,v)$$



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

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

Objective: Find A and B so that simultaneously $Ncut(A,B)$ is minimized and $Nassoc(A,B)$ is maximized.

$$Ncut(A,B) + Nassoc(A,B) = \frac{Cut(A,B) + Assoc(A,A)}{Assoc(A,V)} + \frac{Cut(A,B) + Assoc(B,B)}{Assoc(B,V)} = 2$$

Optional

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

NP-complete

$$\underset{y}{\operatorname{argmin}} \frac{y^t(D-W)y}{y^t D y}$$

$$y(i) \in \{1, -b\}$$

$$y^t D \mathbf{1} = \mathbf{0}$$

Generalized Rayleigh Quotient

$$W(i,j) = w_{i,j} \qquad D(i,i) = \sum_j w(i,j) \qquad d_i = D(i,i)$$

$$k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i} \qquad b = \frac{k}{1-k} \qquad y = (1+x) - b(1-x)$$

Optional

Rayleigh Quotient:

$$\frac{x^t A x}{x^t x} \Rightarrow \min\left(\frac{x^t A x}{x^t x}\right) = \min_{x^t x=1} x^t A x \Rightarrow \begin{array}{l} \min(x^t A x) = \lambda_{min} \\ A x_{min} = \lambda_{min} x_{min} \end{array}$$

Generalized Rayleigh Quotient:

$$\frac{x^t A x}{x^t B x} = \underbrace{\frac{x^t A x}{x^t L L^t x}}_{\text{Cholesky}} = \frac{x^t L L^{-1} A L^{-t} L^t x}{(L^t x)^t (L^t x)} = \frac{(L^t x)^t (L^{-1} A L^{-t})(L^t x)}{(L^t x)^t (L^t x)} = \frac{y^t C y}{y^t y}$$

$$\min\left(\frac{x^t A x}{x^t B x}\right) = \min\left(\frac{y^t C y}{y^t y}\right) = \min_{y^t y=1} y^t C y = \min(y^t C y) = \lambda_{min}$$

$$C y_{min} = \lambda_{min} y_{min}$$

$$L^{-1} A L^{-t} L^t x_{min} = \lambda_{min} L^t x_{min}$$

$$A x_{min} = \lambda_{min} L L^t x_{min}$$

$$A x_{min} = \lambda_{min} B x_{min}$$

Eigenvalue Problem: $Ax = \lambda x$

Generalized Eigenvalue Problem: $Ax = \lambda Bx$

Optional

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

NP-complete

$$\underset{y}{\operatorname{argmin}} \frac{y^t(D-W)y}{y^tDy}$$

$$y(i) \in \{1, -b\}$$

$$y^tD\mathbf{1} = \mathbf{0}$$

Generalized Rayleigh Quotient

y relaxed to take on real values

Generalized Eigenvalue Problem

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

$$W(i,j) = w_{i,j} \qquad D(i,i) = \sum_j w(i,j) \qquad d_i = D(i,i)$$

$$k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i} \qquad b = \frac{k}{1-k} \qquad y = (1+x) - b(1-x)$$

Optional

$$(D - W)y = \lambda D y$$
$$y^t D \mathbf{1} = 0$$

$$z \leftarrow D^{\frac{1}{2}} y$$
$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} z = \lambda z$$
$$z^t D^{\frac{1}{2}} \mathbf{1} = 0$$

$$z_0 = D^{\frac{1}{2}} \mathbf{1} \quad \lambda_0 = 0$$

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} D^{\frac{1}{2}} \mathbf{1} = D^{-\frac{1}{2}}(D - W)\mathbf{1} = \mathbf{0}$$

$\Downarrow$

$$\lambda_0 = eigenvalue$$
$$z_0 = eigenvector$$

$(D - W)$
Positive Semidefinite

$\Downarrow$

$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$

Symmetric Positive Semidefinite

$\Downarrow$

$\lambda_0 = 0$ is the smallest eigenvalue
&, eigenvectors are perpendicular

$\Downarrow$

$$\lambda_0 = \lambda_{min} = 0$$
$$z_0 = z_{min}$$

$$\mathbf{1}^t D \mathbf{1} \neq 0$$

$$z_{opt}^t z_{min} = 0$$

Optional

Let A be a real symmetric matrix.

Let x be orthogonal to $n-1$ smallest eigenvectors of A.

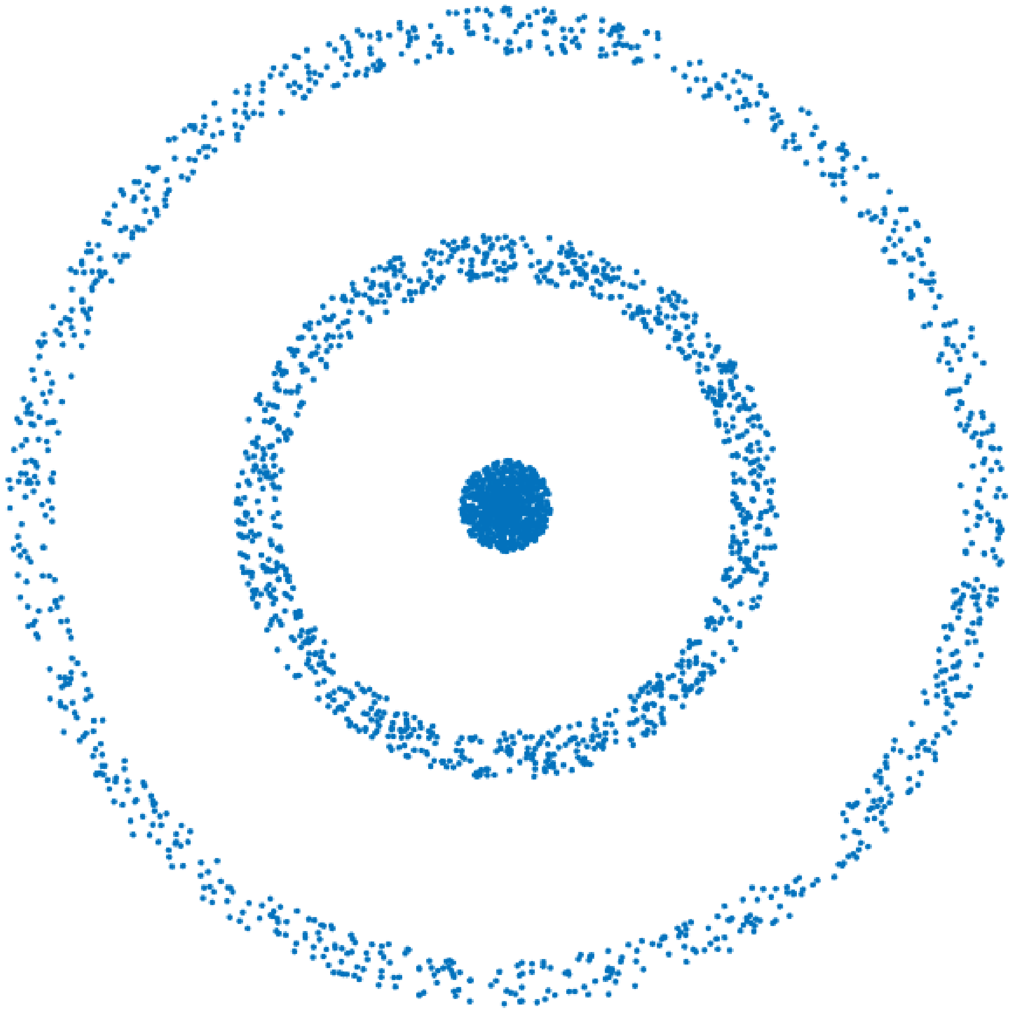
Quotient $\frac{x^t A x}{x^t x}$ is minimized by the n-th smallest eigenvector

and, the minimum value is the corresponding eigenvalue.

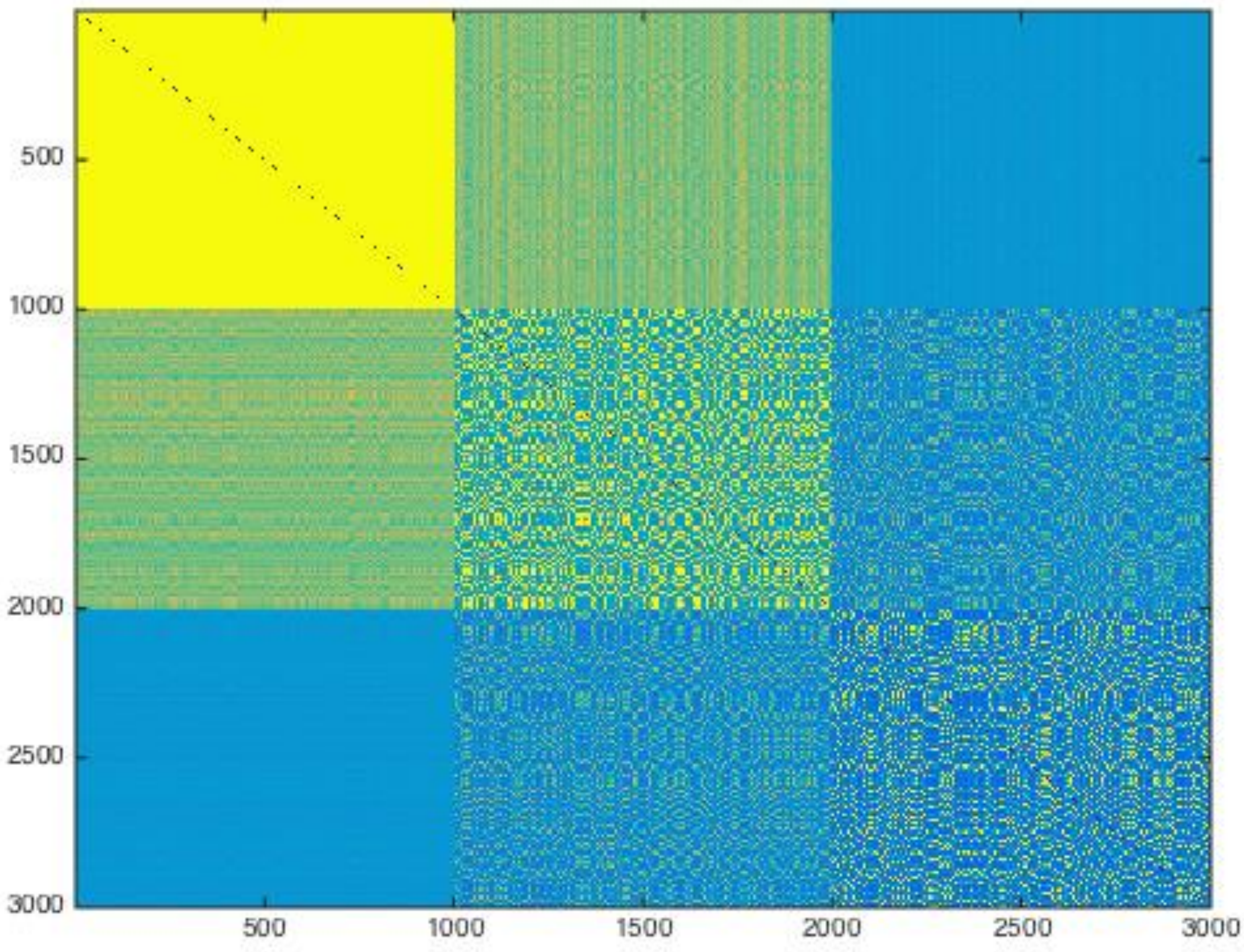
The answer of segmentation:
Second smallest eigenvector of
$(D-W)y = \lambda D y$

Efficient method to obtain the n-th eigenvector:
Lanczos Method

Example

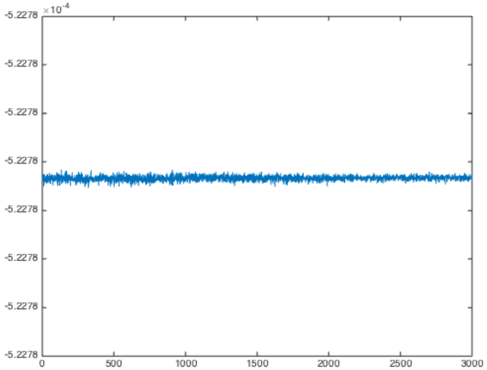


Affinity Matrix

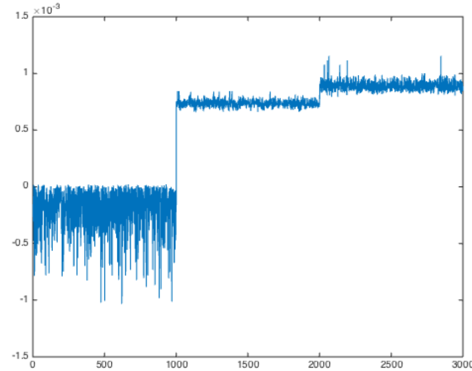


5 Smallest Eigenvalues & Eigenvectors

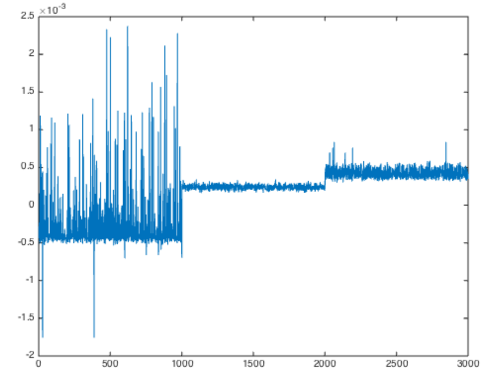
Smallest



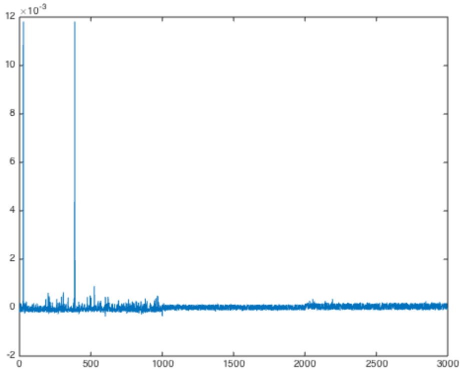
2nd Smallest



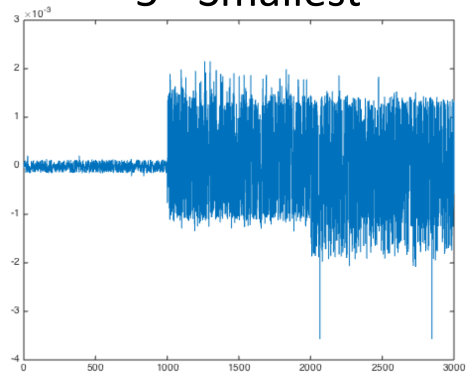
3rd Smallest



4th Smallest



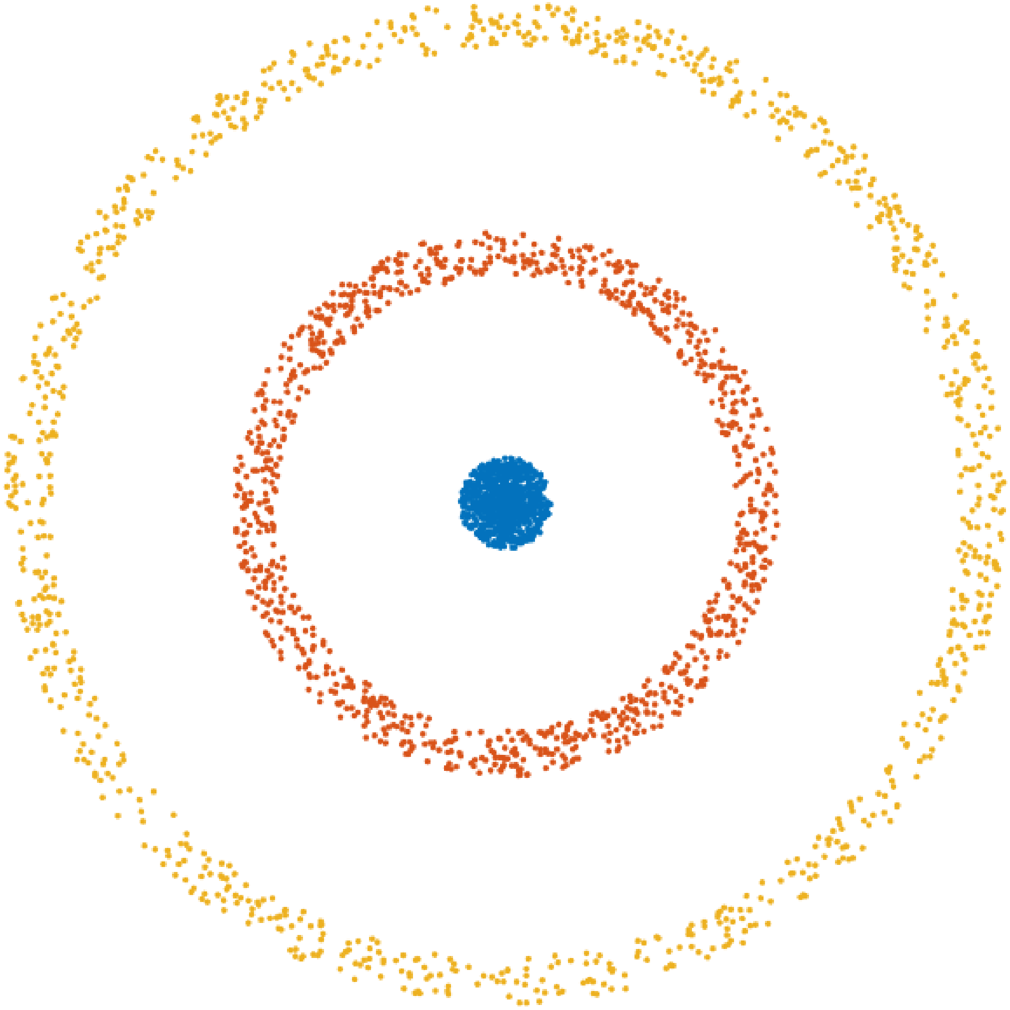
5th Smallest



Eigenvalues

0
0.3555
0.5330
0.6028
0.6160

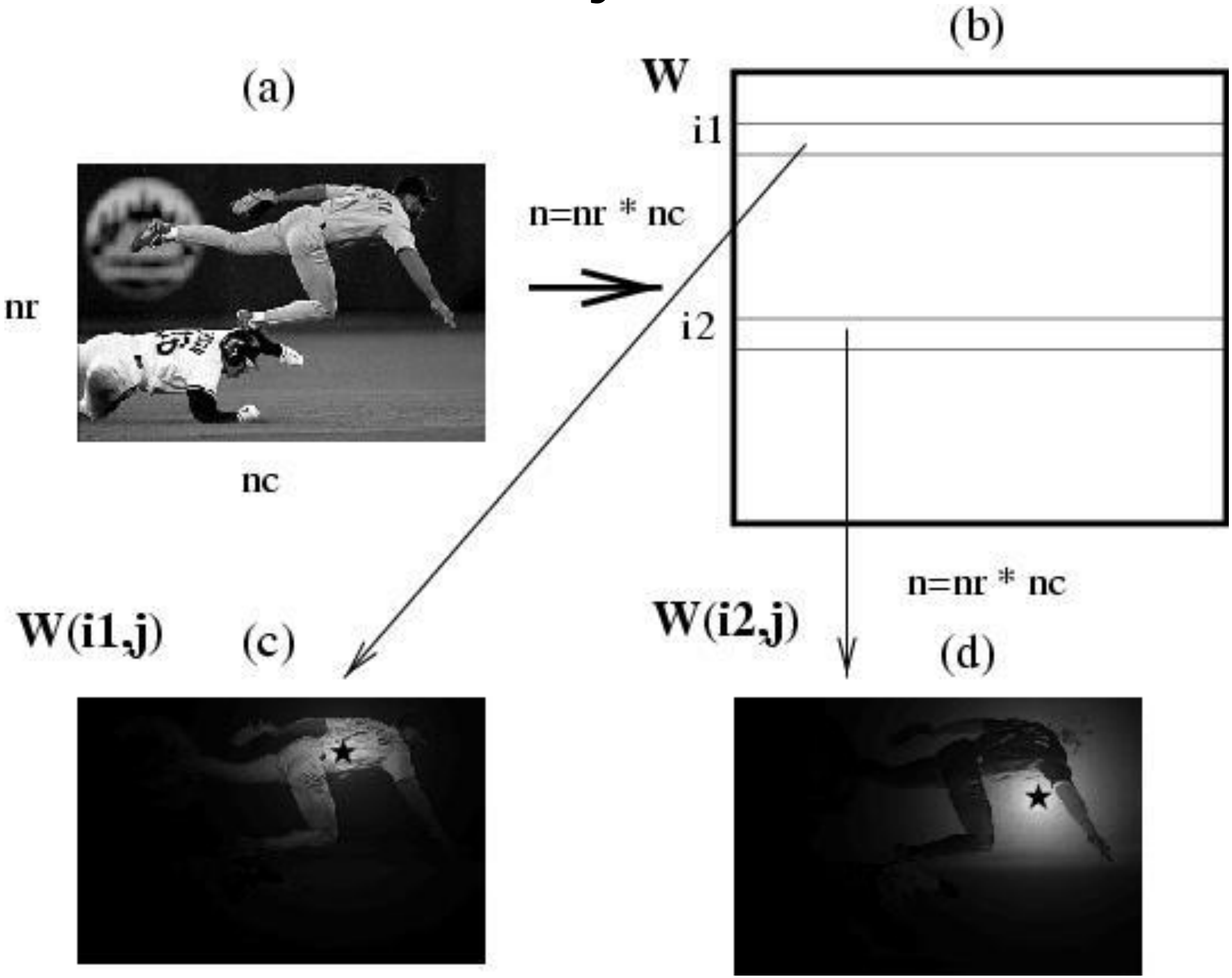
Result



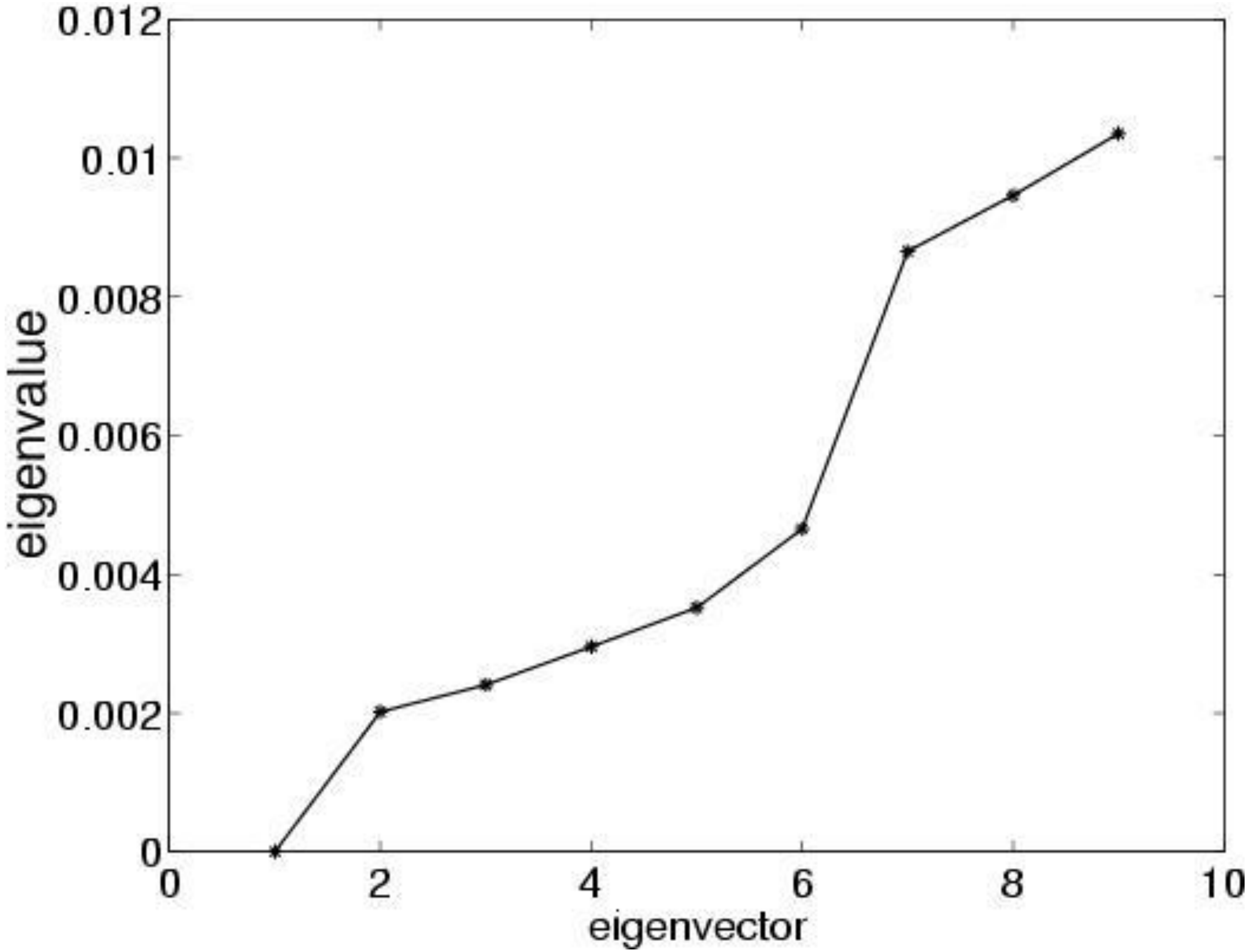
Example

$$W(i,j) = \exp\left(-\frac{\left\|F_i - F_j\right\|_2^2}{s_I}\right).\exp\left(-\frac{\left\|X_i - X_j\right\|_2^2}{s_X}\right) \qquad if \quad \left\|X_i - X_j\right\|_2^2 < r$$

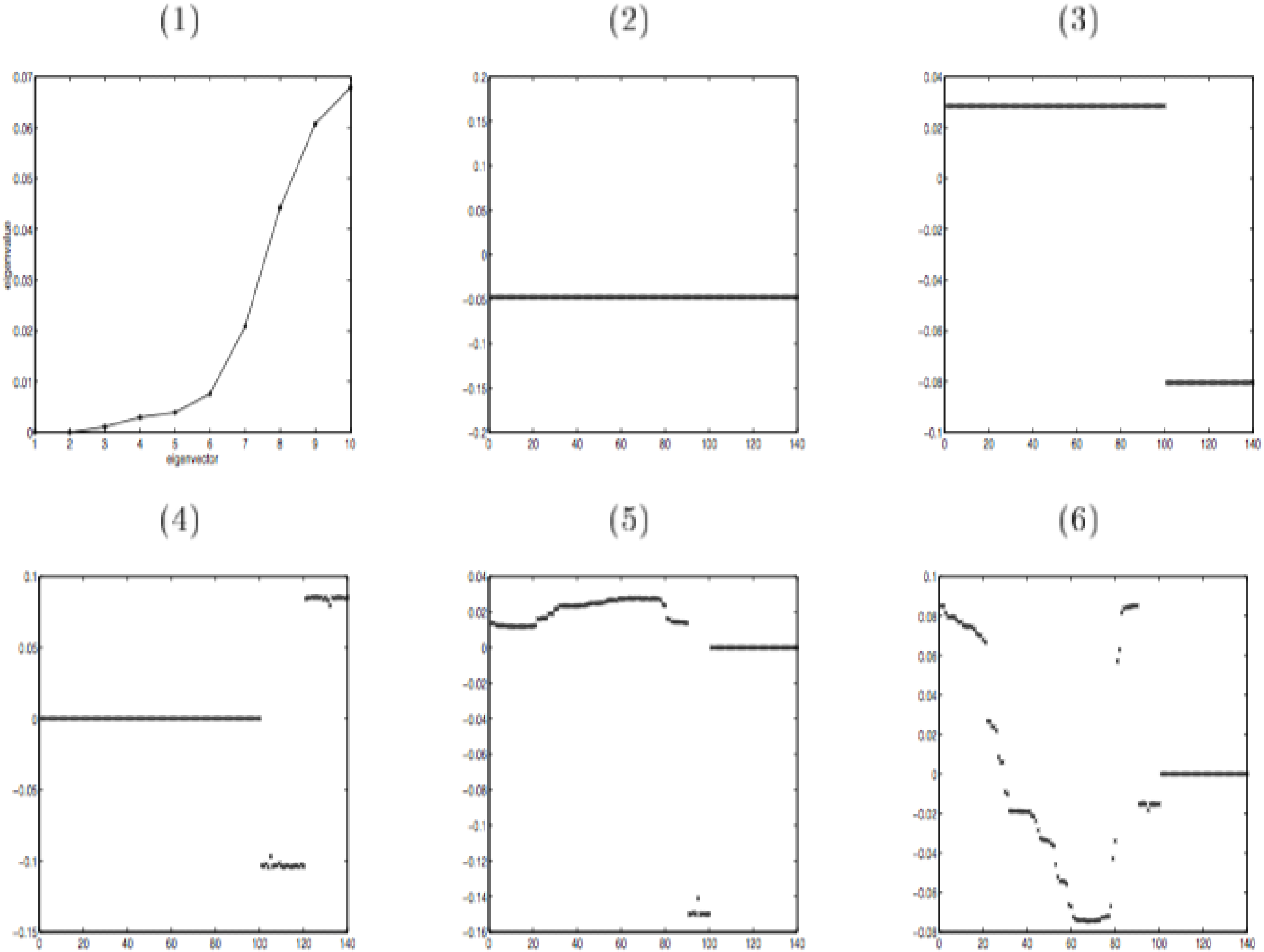
Affinity Matrix



Smallest Eigenvalues

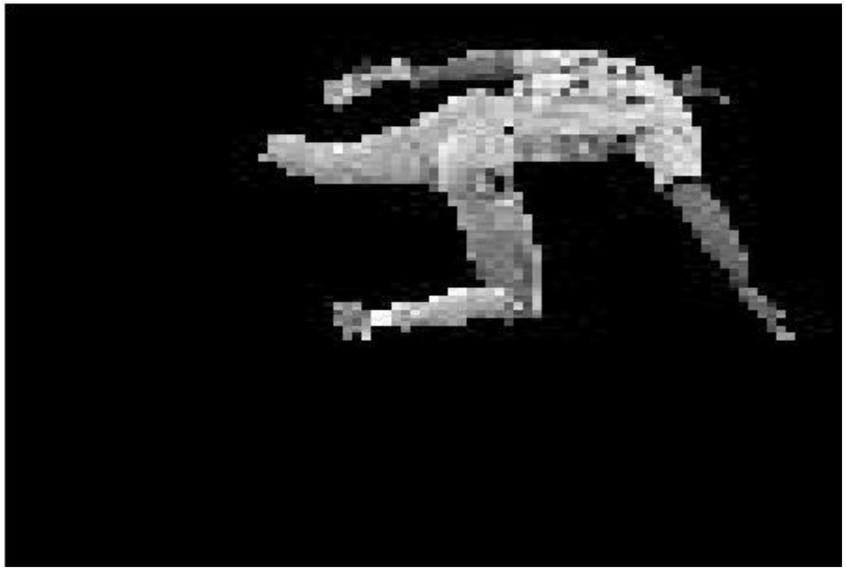


Smallest Eigenvalues & Eigenvectors

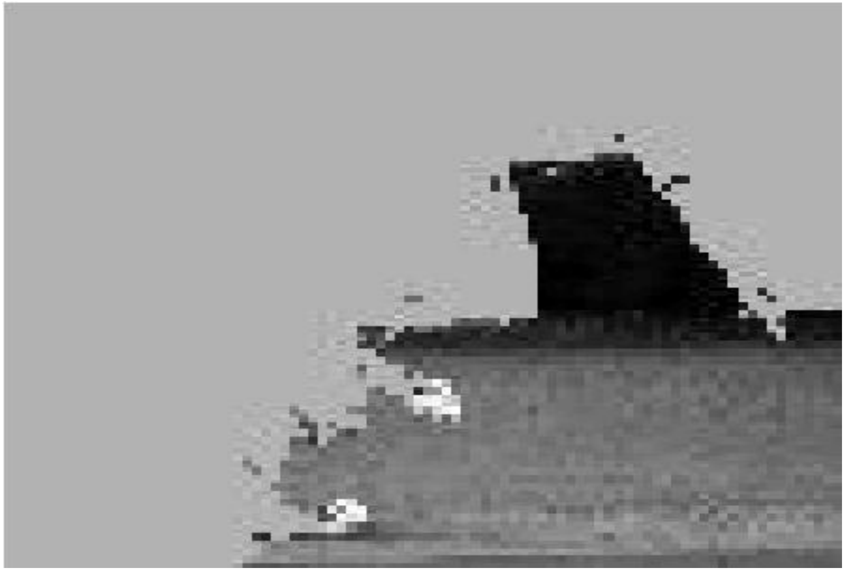


1st Segment

2nd Segment



3rd Segment



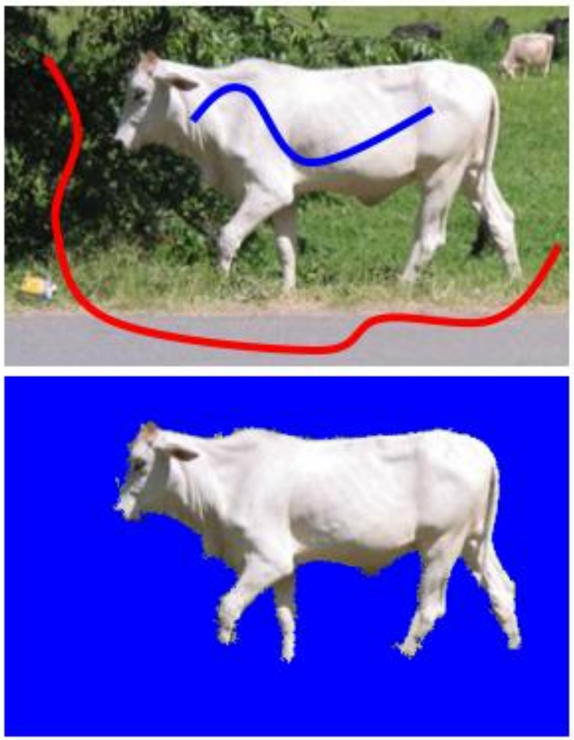
4th Segment

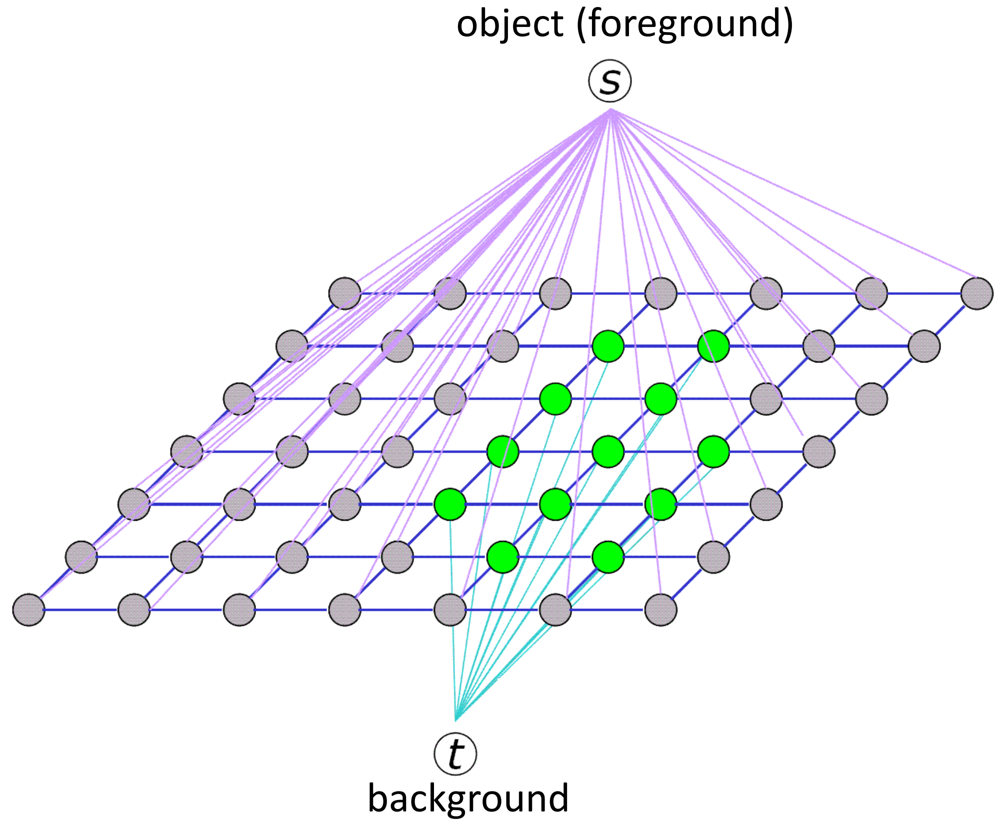
5th Segment

References

- Ncuts
  Szeliski, Section 5.4
- *Paper:*
  Jianbo Shi and Jitendra Malik
  Normalized cuts and image segmentation
  Pattern Analysis and Machine Intelligence (PAMI), 2000

Graph Cuts for Image Segmentation





Combinatorial Optimization

Yuri Boykov and Marie-Pierre Jolly, ***Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images***, ICCV 2001.

Weights

Edge	***Weight (Cost)***	***for***
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$
$\{p,S\}$	$\lambda\, R_p(\text{"}obj\text{"})$	$p \in P, p \notin O \cup B$
	K	$p \in O$
	0	$p \in B$
$\{p,T\}$	$\lambda\, R_p(\text{"}bkg\text{"})$	$p \in P, p \notin O \cup B$
	0	$p \in O$
	K	$p \in B$

$$K = 1 + \max_{p \in P} \sum_{q:\{p,q\} \in N} B_{\{p,q\}}$$

Boundary Term / Smoothness Term

Edge	***Weight (Cost)***	***for***
$\{p, q\}$	$B_{\{p,q\}}$	$\{p, q\} \in N$

Boundary Term / Smoothness Term

Edge	***Weight (Cost)***	***for***
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$

$B_{\{p,q\}}$ = discontinuity penalty.
$B_{\{p,q\}}$ is large when pixel p and q are similar.
$B_{\{p,q\}}$ is close to zero when the two are very different.

$$B_{\{p,q\}} \propto \exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$$

Boundary Term / Smoothness Term

Edge	***Weight (Cost)***	***for***
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$

$B_{\{p,q\}}$ = discontinuity penalty.
$B_{\{p,q\}}$ is large when pixel p and q are similar.
$B_{\{p,q\}}$ is close to zero when the two are very different.

$$B_{\{p,q\}} \propto \exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$$

Defining a segmentation:
$A = \left(A_1, \dots, A_p, \dots, A_{|P|}\right)$
Binary vector

Boundary property term:

$$B(A) = \sum_{\{p,q\} \epsilon N} B_{\{p,q\}}\, \delta\left(A_p, A_q\right)$$

$$\delta\left(A_p, A_q\right) = \begin{cases} 1 & A_p \neq A_q \\ 0 & A_p = A_q \end{cases}$$

Weights

Edge	***Weight (Cost)***	***for***
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$
$\{p,S\}$	$\lambda\, R_p(\text{"}obj\text{"})$	$p \in P, p \notin O \cup B$
	K	$p \in O$
	0	$p \in B$
$\{p,T\}$	$\lambda\, R_p(\text{"}bkg\text{"})$	$p \in P, p \notin O \cup B$
	0	$p \in O$
	K	$p \in B$

$$K = 1 + \max_{p \in P} \sum_{q:\{p,q\} \in N} B_{\{p,q\}}$$

Region Term / Data Term

$R_p(A_p) =$ Individual penalties for assigning pixel p to "object" and "background".

$R_p(A_p)$ may reflect on how the intensity of pixel p fits into a known intensity model (histogram) of object and background.

Region Term / Data Term

$R_p(A_p) =$ Individual penalties for assigning pixel p to "object" and "background".

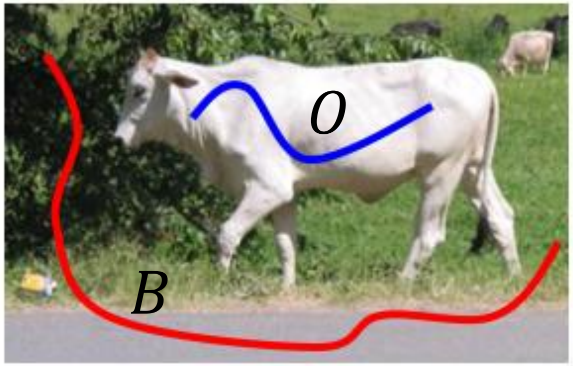
$R_p(A_p)$ may reflect on how the intensity of pixel p fits into a known intensity model (histogram) of object and background.

Similarity to foreground (object):

$$R_p(\text{"}obj\text{"}) = -\ln\big(\Pr\big(I_p|\text{"}bkg\text{"}\big)\big)$$

Similarity to background:

$$R_p(\text{"}bkg\text{"}) = -\ln\big(\Pr\big(I_p|\text{"}obj\text{"}\big)\big)$$



Region Term / Data Term

$R_p(A_p) =$ Individual penalties for assigning pixel p to "object" and "background".

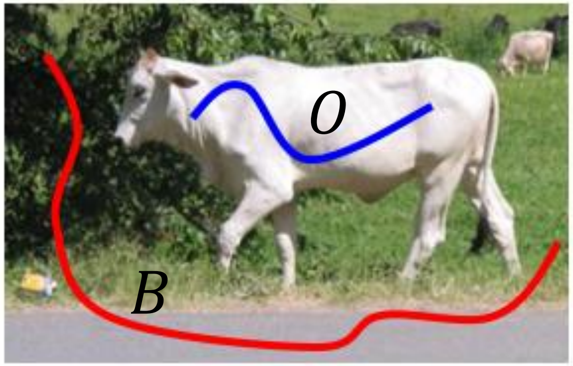
$R_p(A_p)$ may reflect on how the intensity of pixel p fits into a known intensity model (histogram) of object and background.

Similarity to foreground (object):

$$R_p(\text{"}obj\text{"}) = -\ln\left(\Pr\left(I_p | \text{"}bkg\text{"}\right)\right)$$

Similarity to background:

$$R_p(\text{"}bkg\text{"}) = -\ln\left(\Pr\left(I_p | \text{"}obj\text{"}\right)\right)$$



Region property term:

$$R(A) = \sum_{p \epsilon P} R_p(I_p)$$

Defining a segmentation:
$A = (A_1, \dots, A_p, \dots, A_{|P|})$
Binary vector

Cost Function

$$E(A) = \lambda\, R(A) + B(A)$$

Defining a segmentation:
$A = \left(A_1, \dots, A_p, \dots, A_{|P|}\right)$
Binary vector

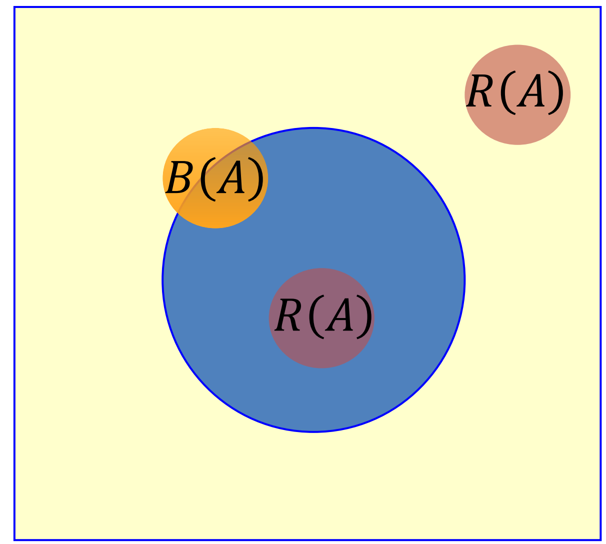
Cost Function

$$E(A) = \boxed{\lambda}\,\boxed{R(A)} + \boxed{B(A)}$$

Boundary properties term

Region properties term

Coefficient : relative importance



Defining a segmentation:

$A = (A_1, \dots, A_p, \dots, A_{|P|})$

Binary vector

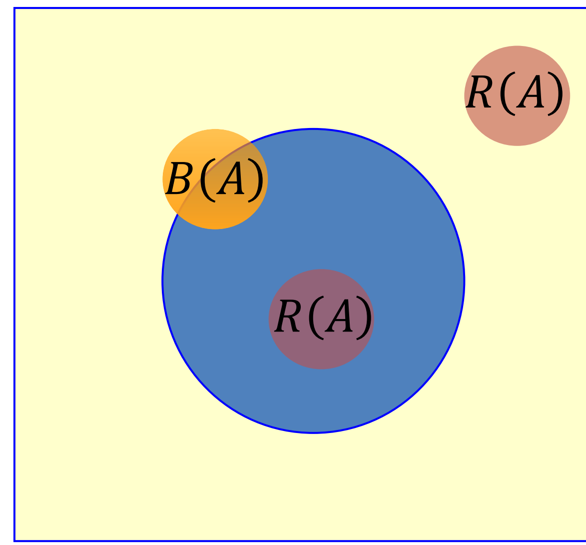
Cost Function

$$E(A) = \boxed{\lambda}\,\boxed{R(A)} + \boxed{B(A)}$$

Boundary properties term

Region properties term

Coefficient : relative importance



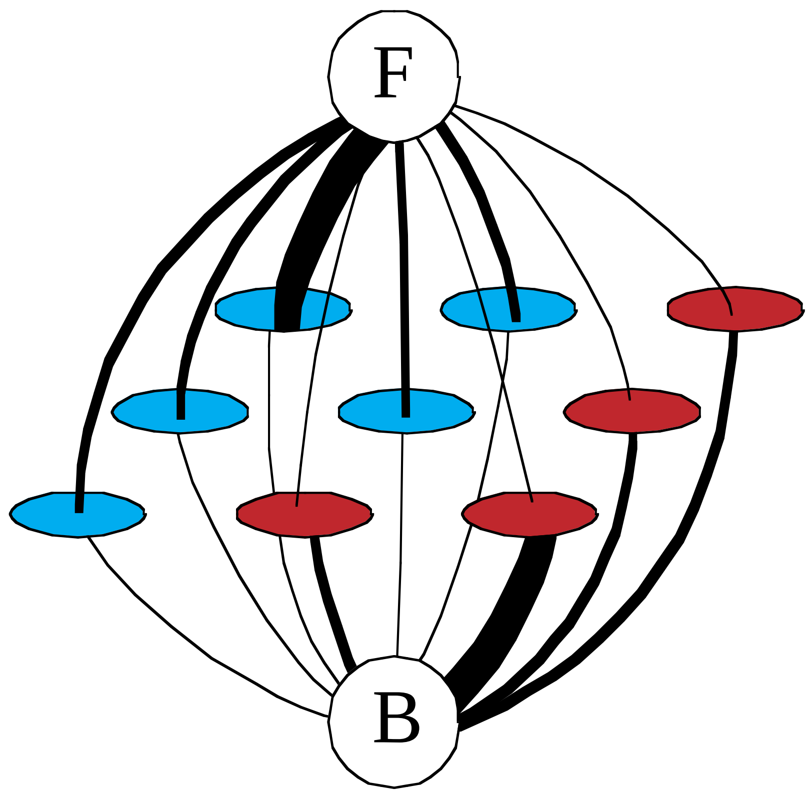
$$R(A) = \sum_{p \epsilon P} R_p(I_p)$$

$$B(A) = \textstyle\sum_{\{p,q\} \epsilon N} B_{\{p,q\}}\, \delta(A_p, A_q) \qquad \delta(A_p, A_q) = \begin{cases} 1 & A_p \neq A_q \\ 0 & A_p = A_q \end{cases}$$

Region Term / Data Term

- Put one edge between each pixel and both F & B

- Weight of edge: $R(A) = \sum_{p \epsilon P} R_p(I_p)$
  - Don't forget huge weight for hard constraints

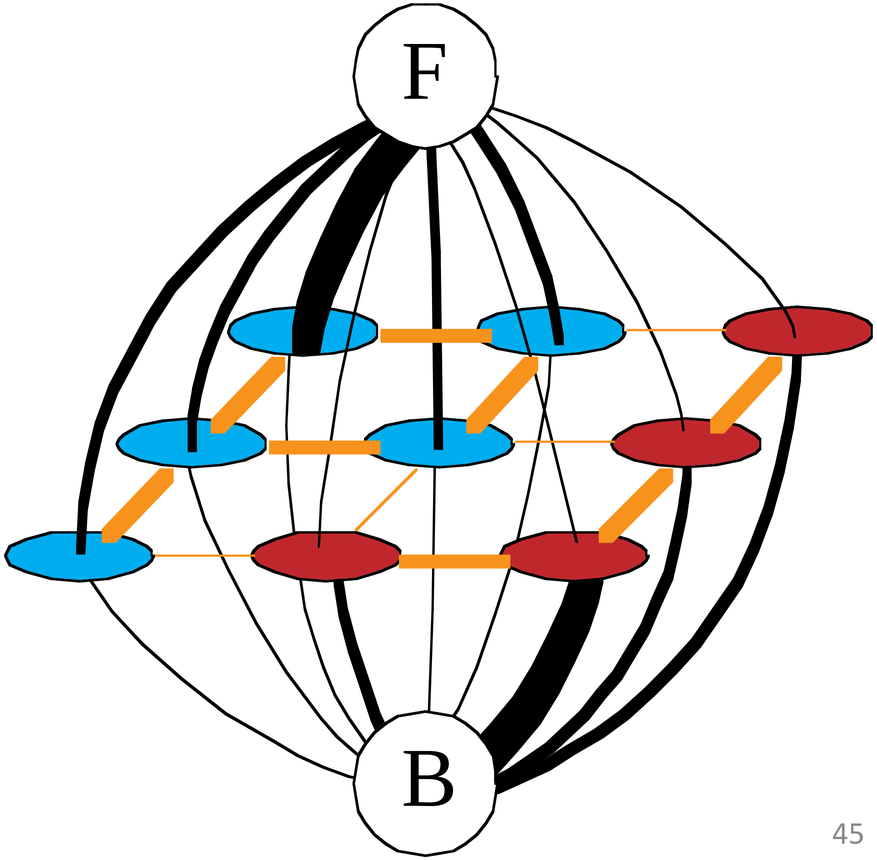


Boundary Term / Smoothness Term

- Add an edge between each neighbor pair
- Weight = smoothness term

$$B(A) = \sum_{\{p,q\}\epsilon N} B_{\{p,q\}}\, \delta(A_p, A_q)$$

$$\delta(A_p, A_q) = \begin{cases} 1 & A_p \neq A_q \\ 0 & A_p = A_q \end{cases}$$

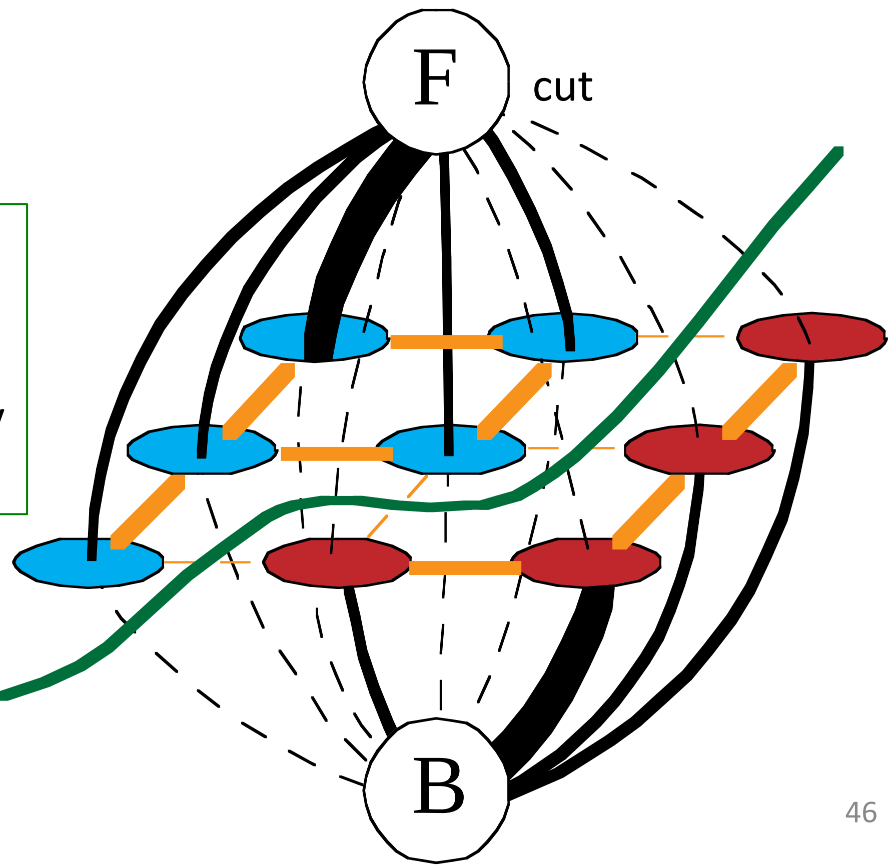


Min Cut

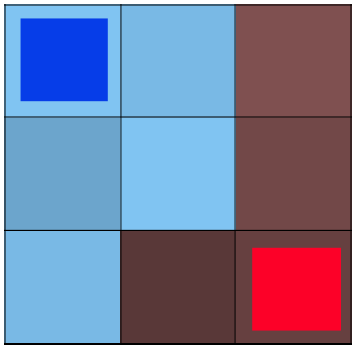
- Energy optimization equivalent to graph min cut
- Cut: remove edges to disconnect F from B
- Minimum: minimize sum of cut edge weight

$$E(A) = \lambda\, R(A) + B(A)$$

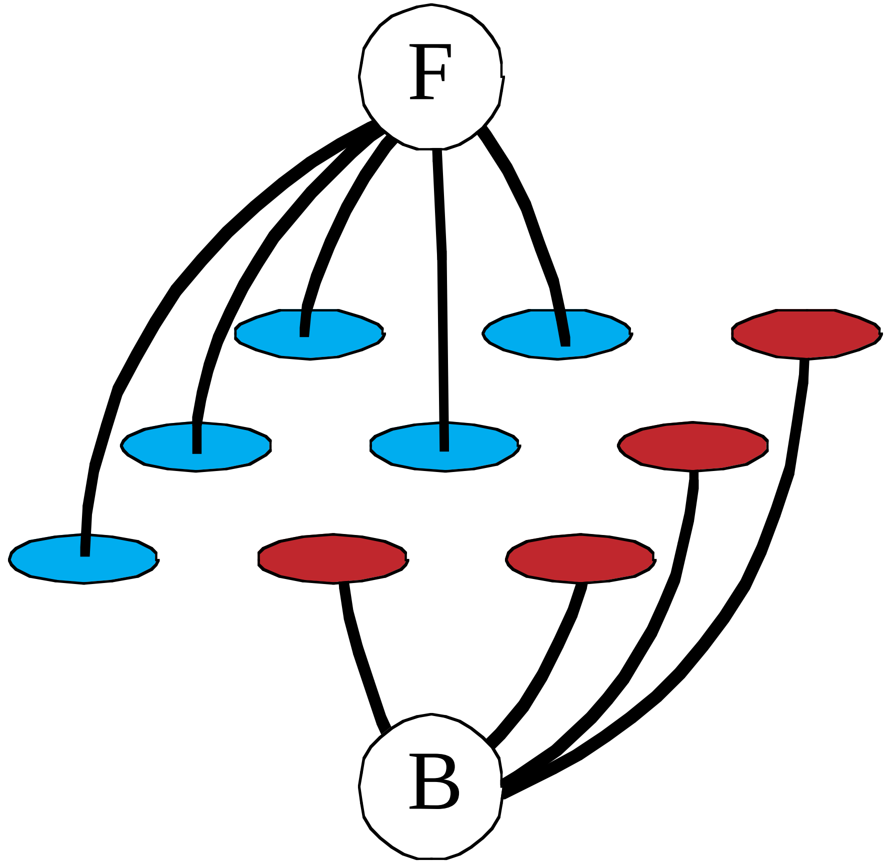
A well-known combinatorial optimization fact:
A globally minimum cut of a graph with two terminals can be computed efficiently in low order polynomial time.



Result
min cut/max flow algorithm

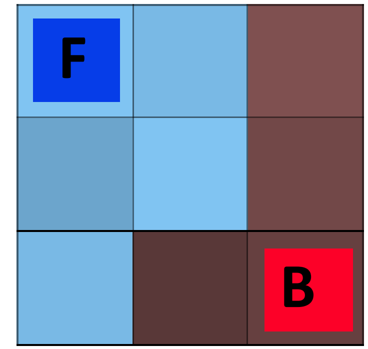


F	F	B
F	F	B
F	B	B



Overview

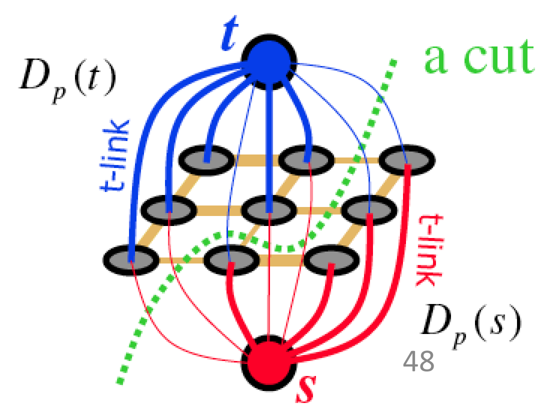
- Interactive image segmentation using graph cut
- Binary label: foreground vs. background
- User labels some pixels (hard constraints)
- Exploit
  - Statistics of known Fg & Bg
  - Smoothness of label
- Turn into discrete graph optimization
  - Graph cut (min cut / max flow)



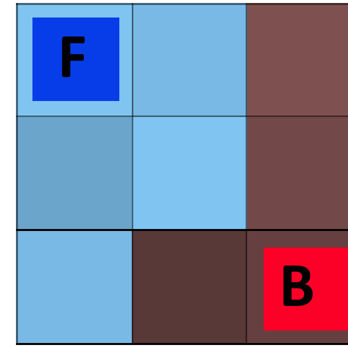
F	F	B
F	F	B
F	B	B

Defining a segmentation:
$A = (A_1, \dots, A_p, \dots, A_{|P|})$
Binary vector

Minimizing cost function
$E(A) = \lambda\, R(A) + B(A)$



Energy Function: $E(A) = \lambda\, R(A) + B(A)$

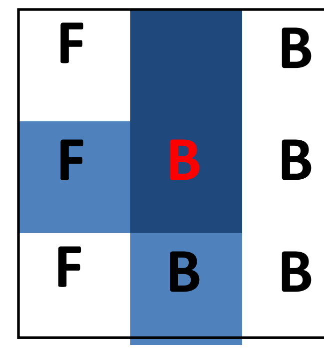


- Labeling: one value per pixel, F or B
- Energy (labeling) = data + smoothness
  - Very general situation
  - Will be minimized
- Data: for each pixel
  - Probability that this color belongs to F (resp. B)
- Smoothness (aka regularization): per neighboring pixel pair
  - Penalty for having different labels
  - Penalty is downweighted if the two pixel colors are very different
  - Similar in spirit to bilateral filter

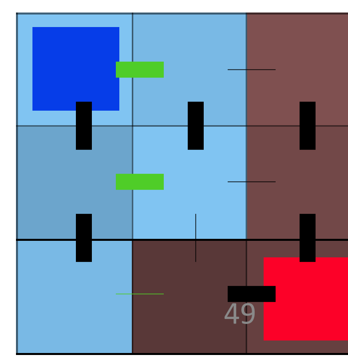
One labeling (ok, not best)

F	B	B
F	B	B
F	B	B

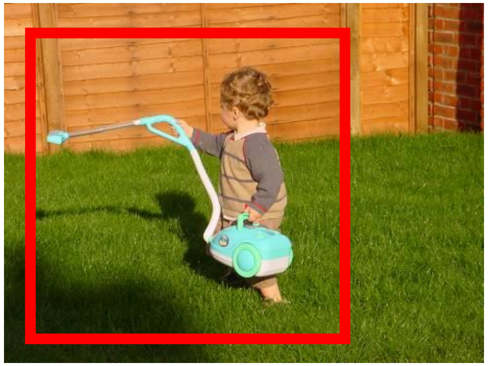
Data



Smoothness



What is easy or hard about these cases for graphcut-based segmentation?



Slide: Grab-Cut presentation

Easier examples

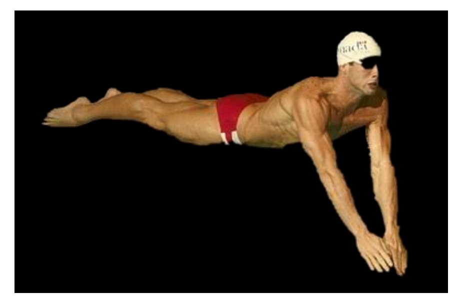
Slide: Grab-Cut presentation

More difficult Examples

Camouflage & Low Contrast | **Fine structure** | **Harder Case**

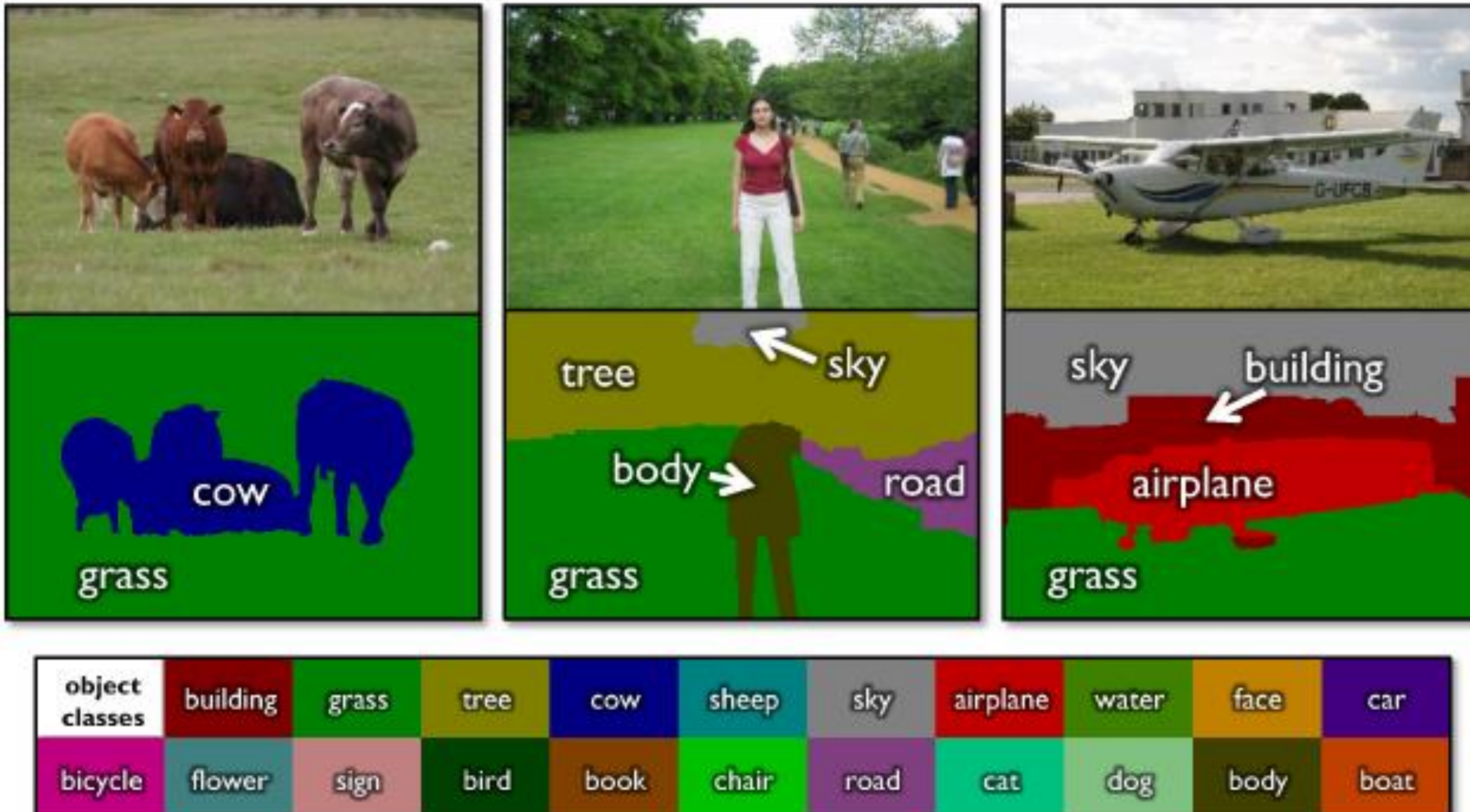
Initial Rectangle

Initial Result



Slide: Grab-Cut presentation

Using graph cuts for recognition



TextonBoost (Shotton et al. 2009 IJCV)