Unsupervised Segmentation/Supervised Segmentation

Unsupervised Segmentation and Grouping

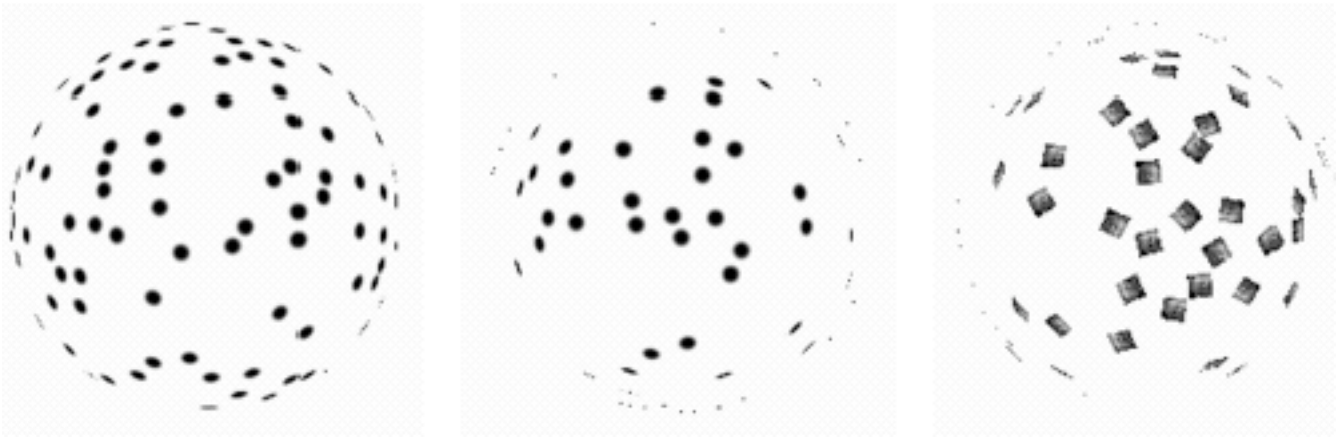
- Motivation: Many computer vision problems would be easy, except for background interference.
- Unsupervised Segmentation: no training data
- Use: Obtain a compact representation from an image/motion sequence/set of tokens
- Should support application
- Broad theory is absent at present

- Grouping (or clustering)
  - collect together tokens that "belong together"
- Fitting
  - associate a model with tokens
  - issues
    - which model?
    - which token goes to which element?
    - how many elements in the model?

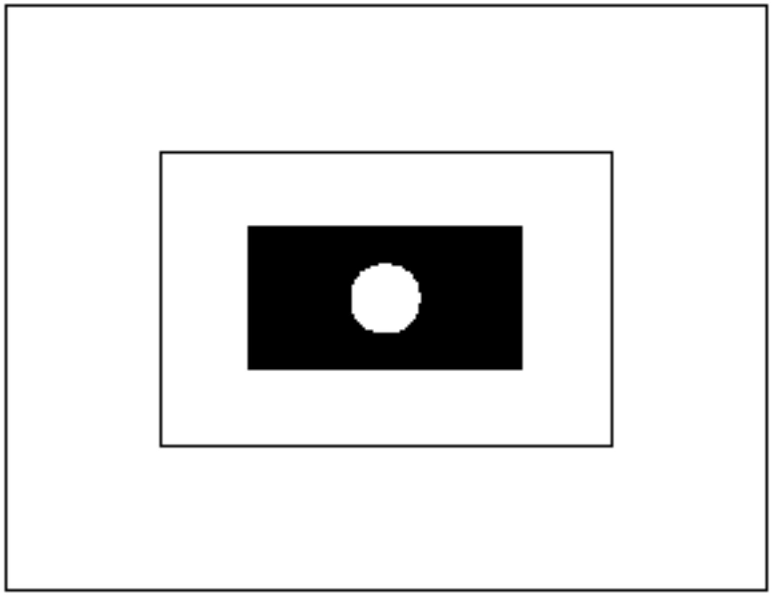
General ideas

- Features (tokens)
  - whatever we need to group (pixels, points, surface elements, etc., etc.)

- top down segmentation (model based)
  - features belong together because they lie on the same object.
  - Supervised segmentation
- bottom up segmentation (image based)
  - features belong together because they are locally coherent
  - Unsupervised segmentation
- These two are not mutually exclusive



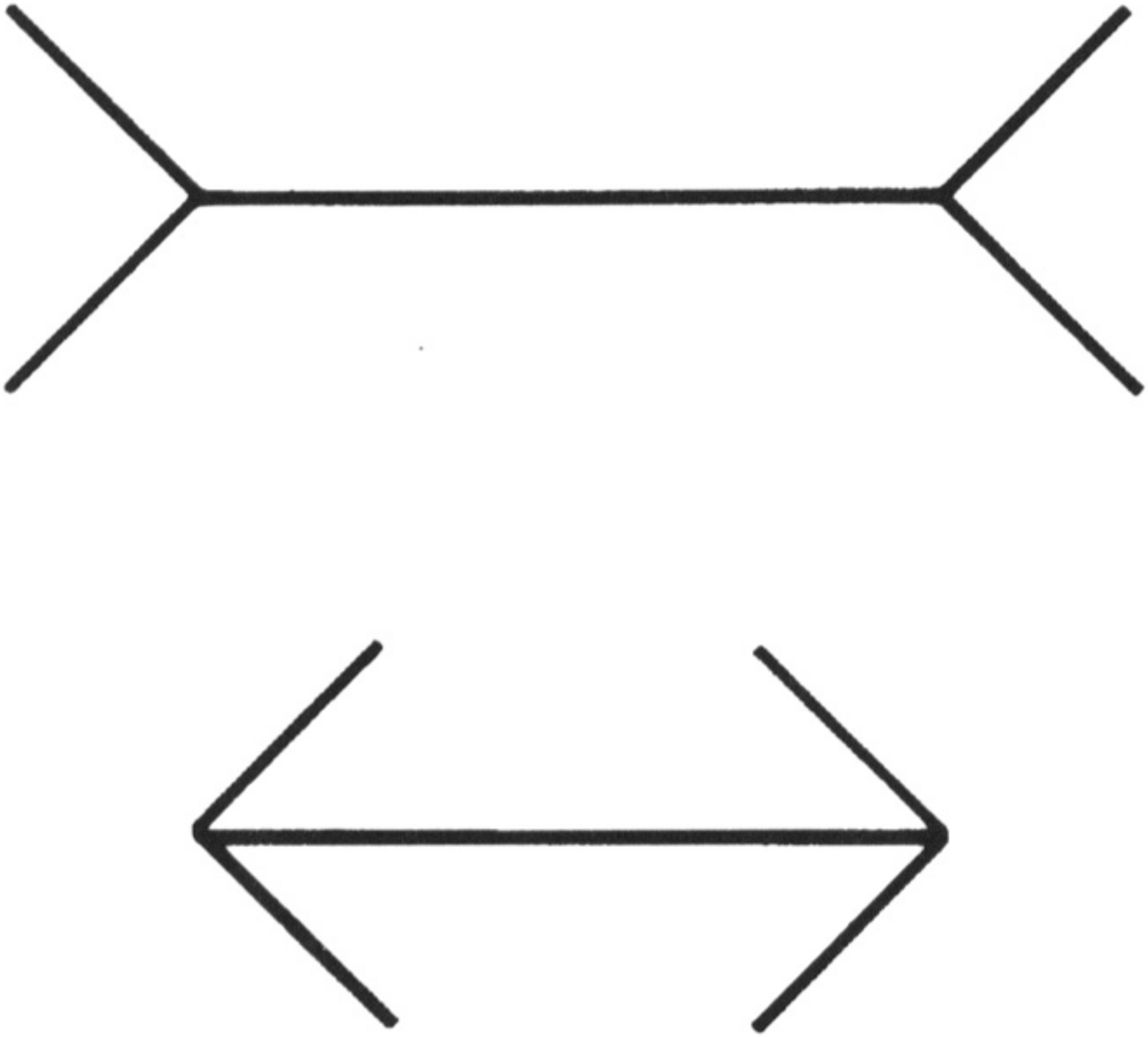
Why do these features belong together?

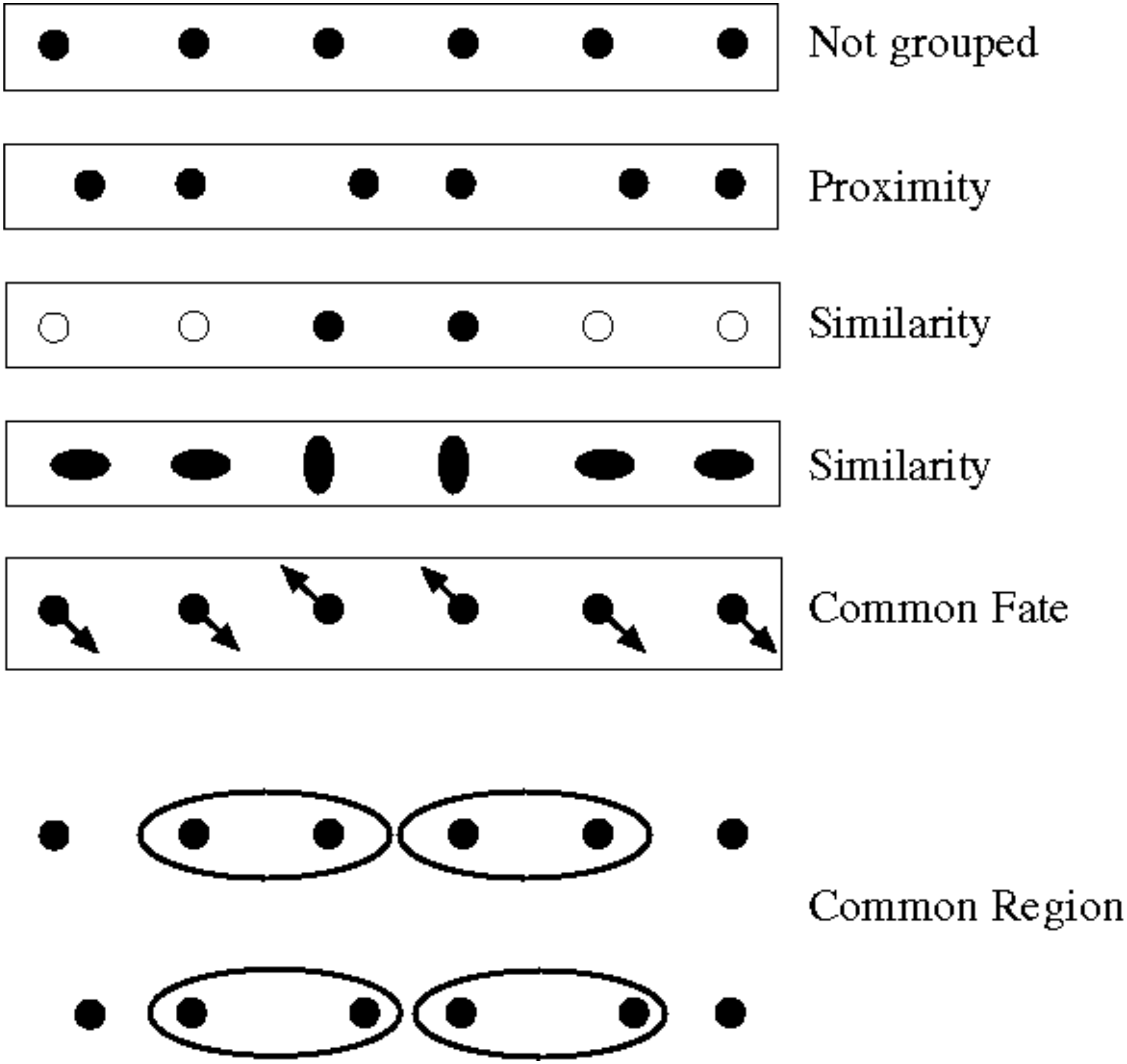


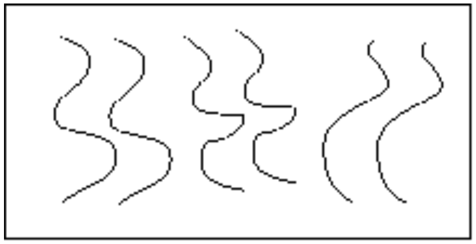
Basic ideas of grouping in humans

- Figure-ground discrimination
  - grouping can be seen in terms of allocating some elements to a figure, some to ground
  - impoverished theory

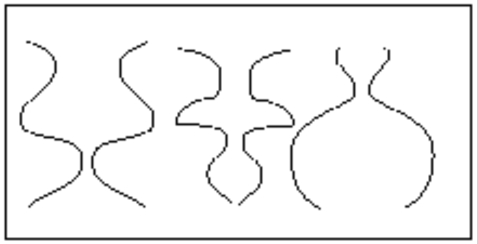
- Gestalt properties
  - elements in a collection of elements can have properties that result from relationships (Muller-Lyer effect)
    - gestaltqualitat
  - A series of factors affect whether elements should be grouped together
    - Gestalt factors



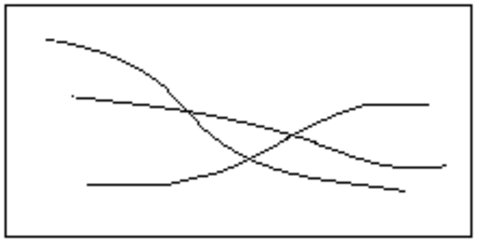

Not grouped
Proximity
Similarity
Similarity
Common Fate
Common Region



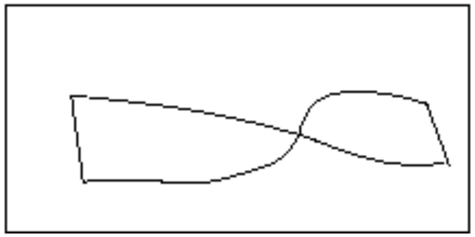
Parallelism



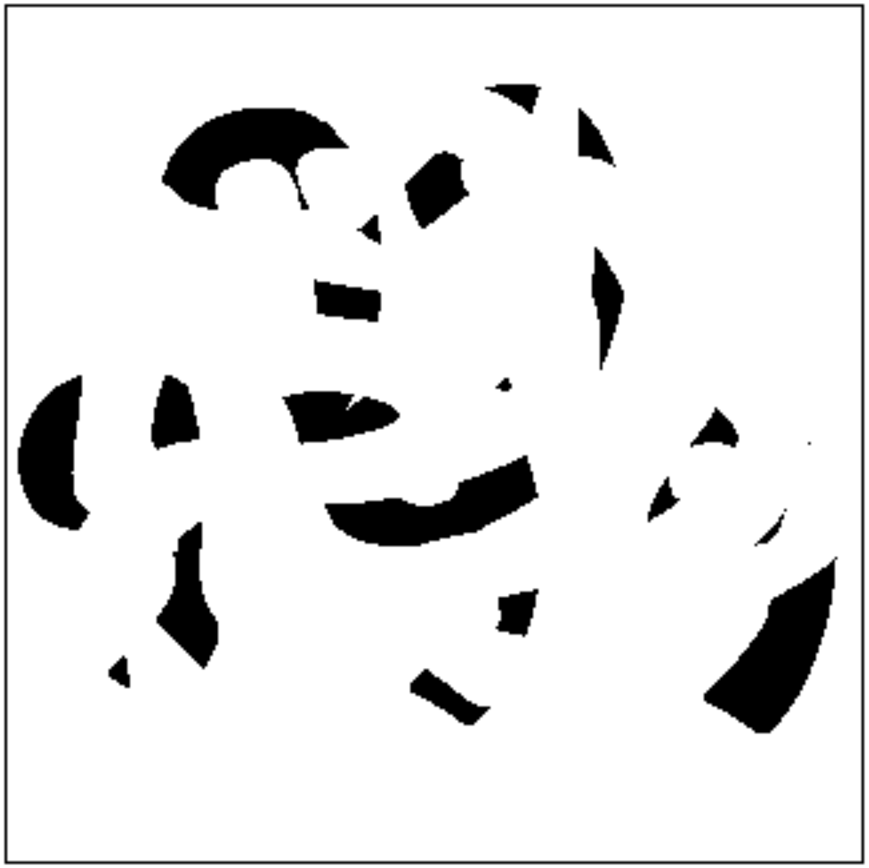
Symmetry

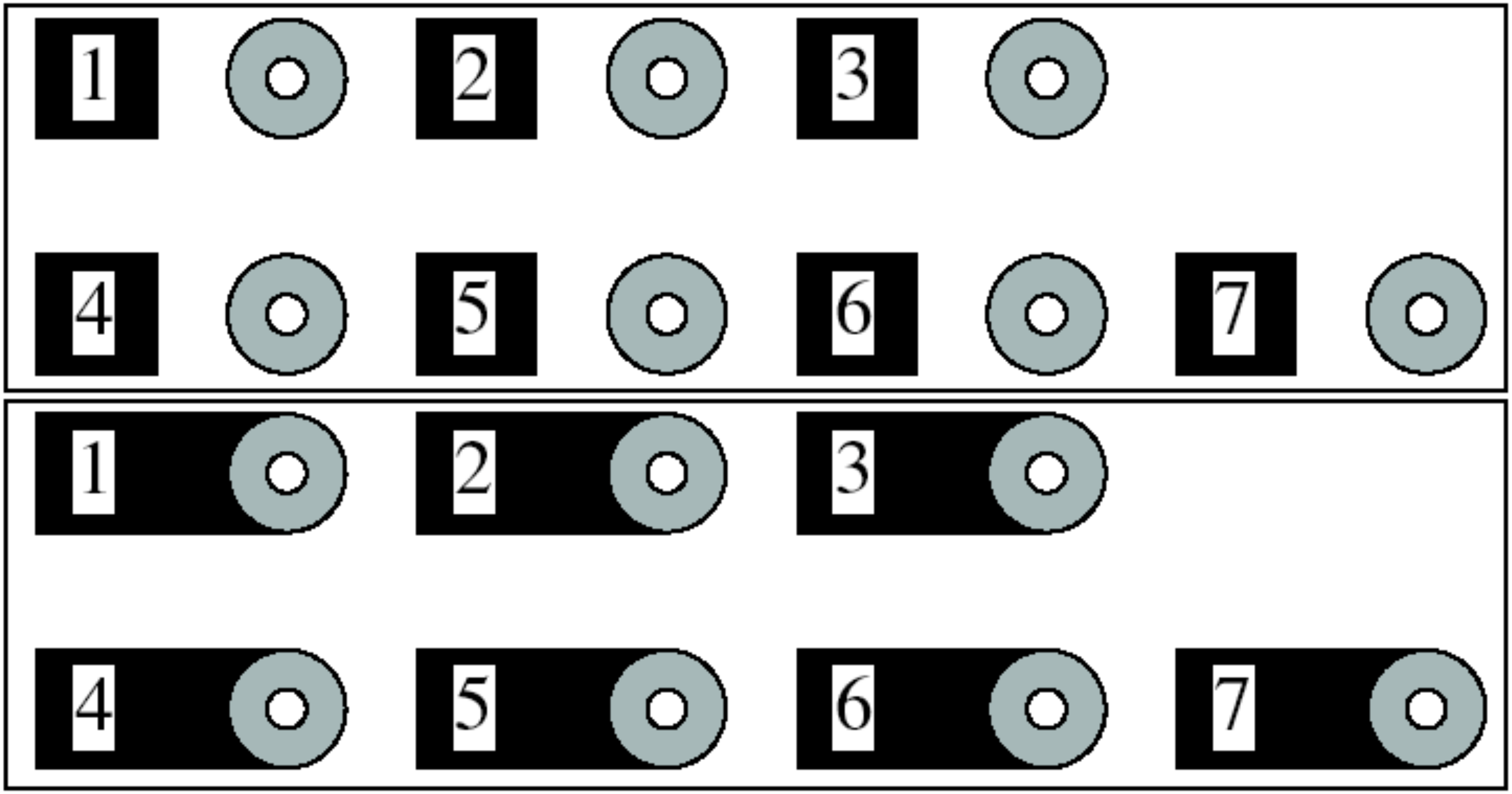


Continuity

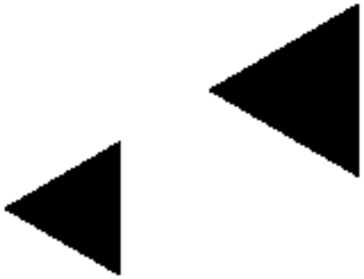


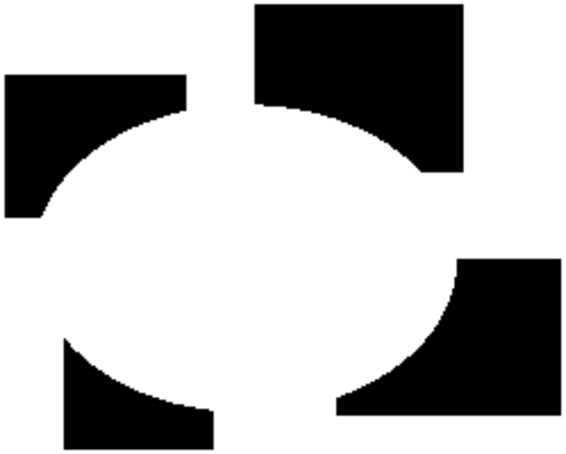
Closure





1
2
3
4
5
6
7
1
2
3
4
5
6
7





Are Gestalt laws the result of observed regularity in scenes?

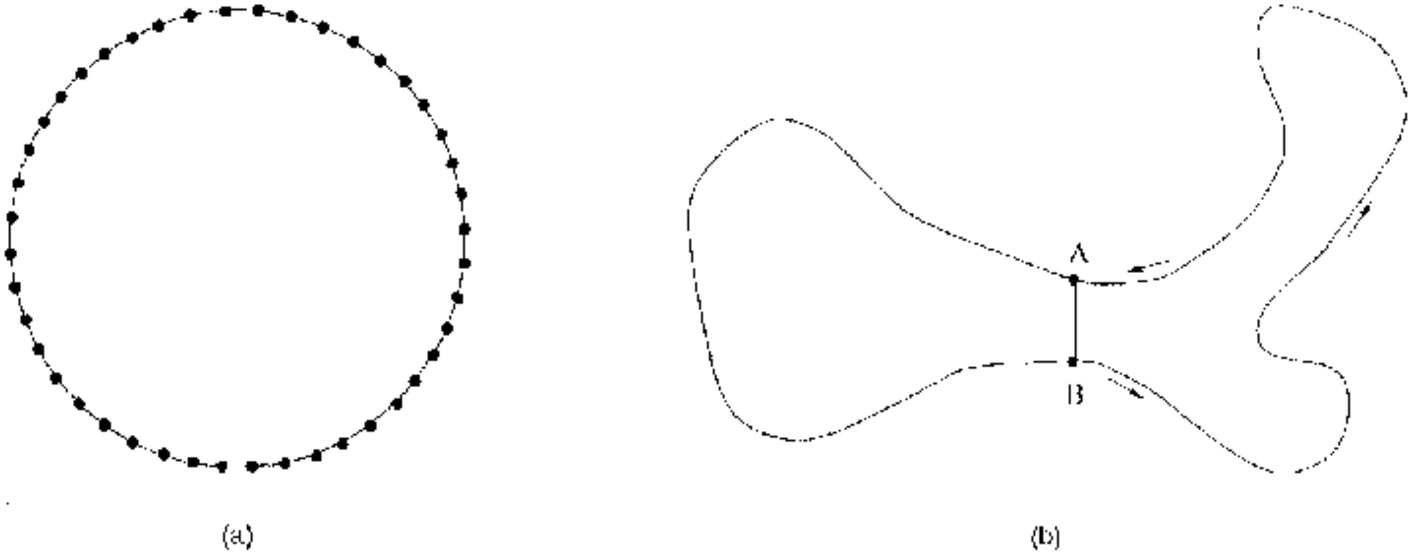
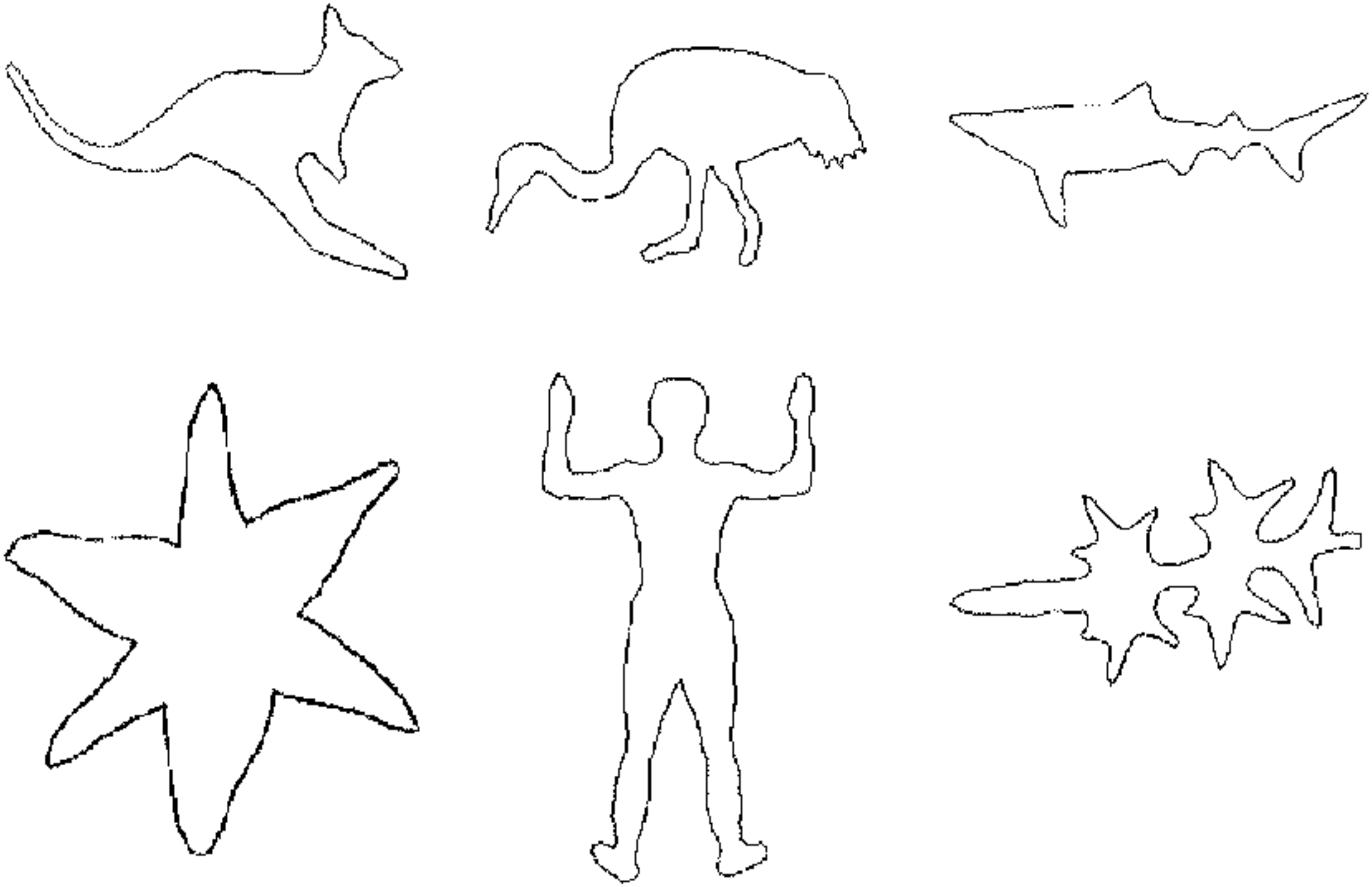
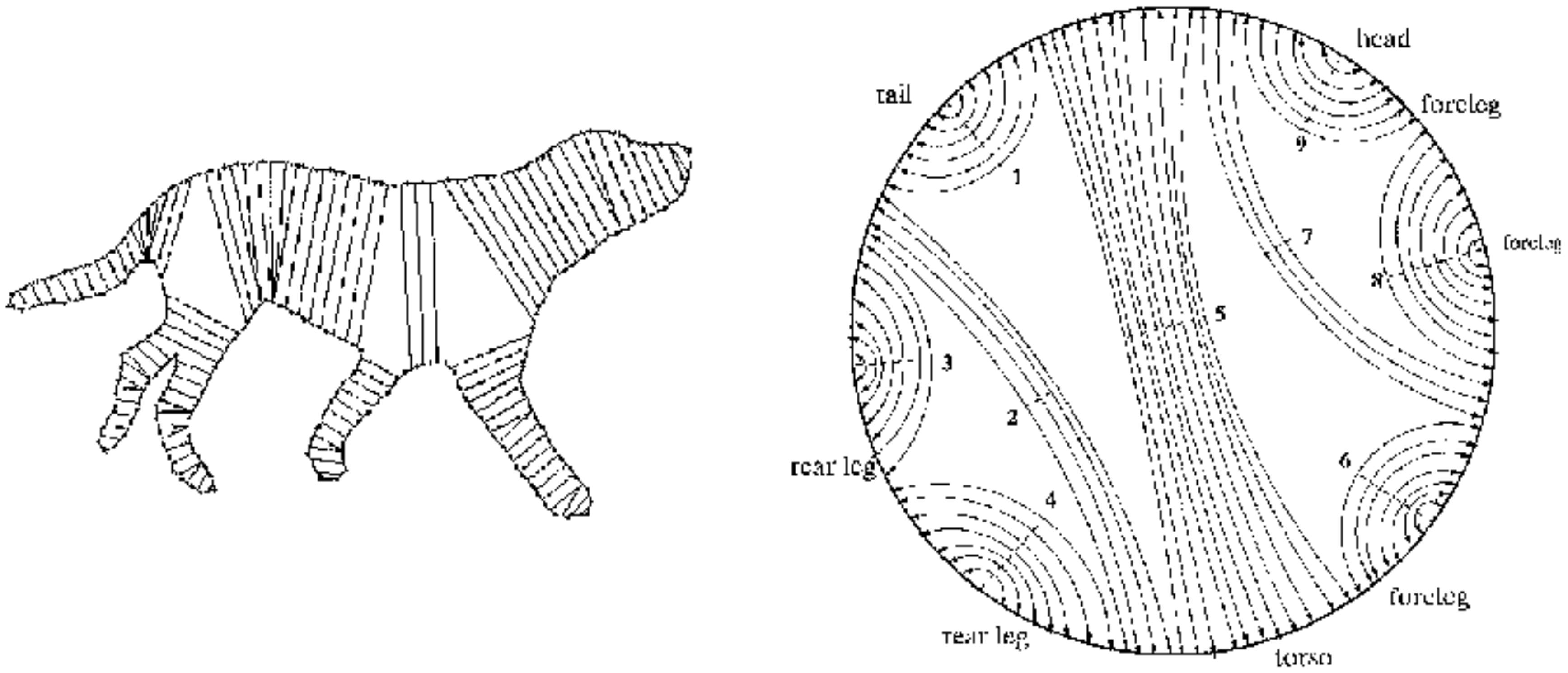


Fig. 1. (a) A 1D Markov random field where the nodes represent random variables for positions of contour points. (b) Node A is spatially adjacent to point B, but it is far away from B in the circular neighborhood of (a).




head
tail
foreleg
9
1
7
foreleg
8
5
3
2
rear leg
6
4
foreleg
rear leg
torso

(a)

(b)

Collect statistics on local curvature and adjacency for natural contours.

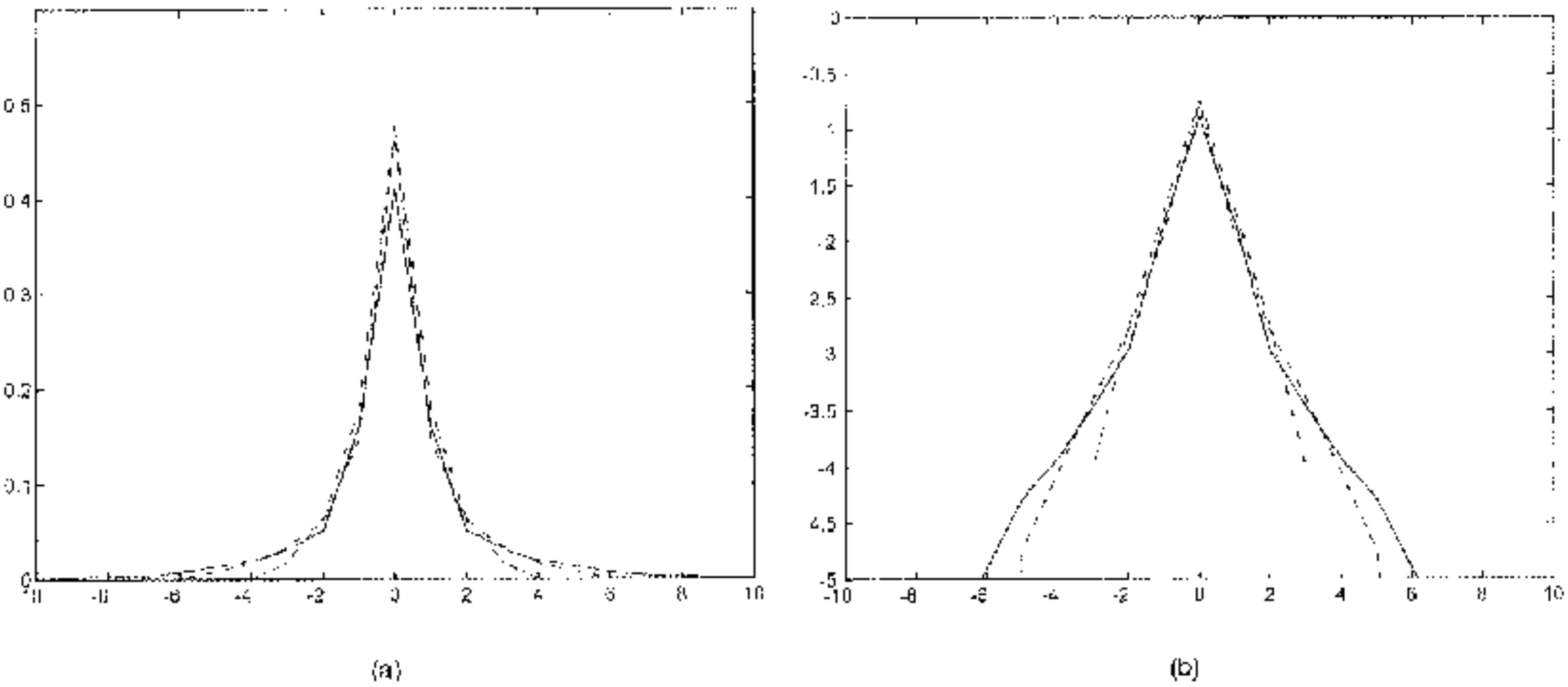


Fig. 5. (a) The histograms of $\kappa(s)$ averaged over 22 animate objects at scale 0 (solid curve), scale 1 (dashed curve), and scalo 2 (dash-dotted curve), the horizontal axis Is $\kappa(s)$ with unit $d\kappa =$ [illegible]. (b) The logarithm of curves in (a).

Sampled boundaries from learning probability model.

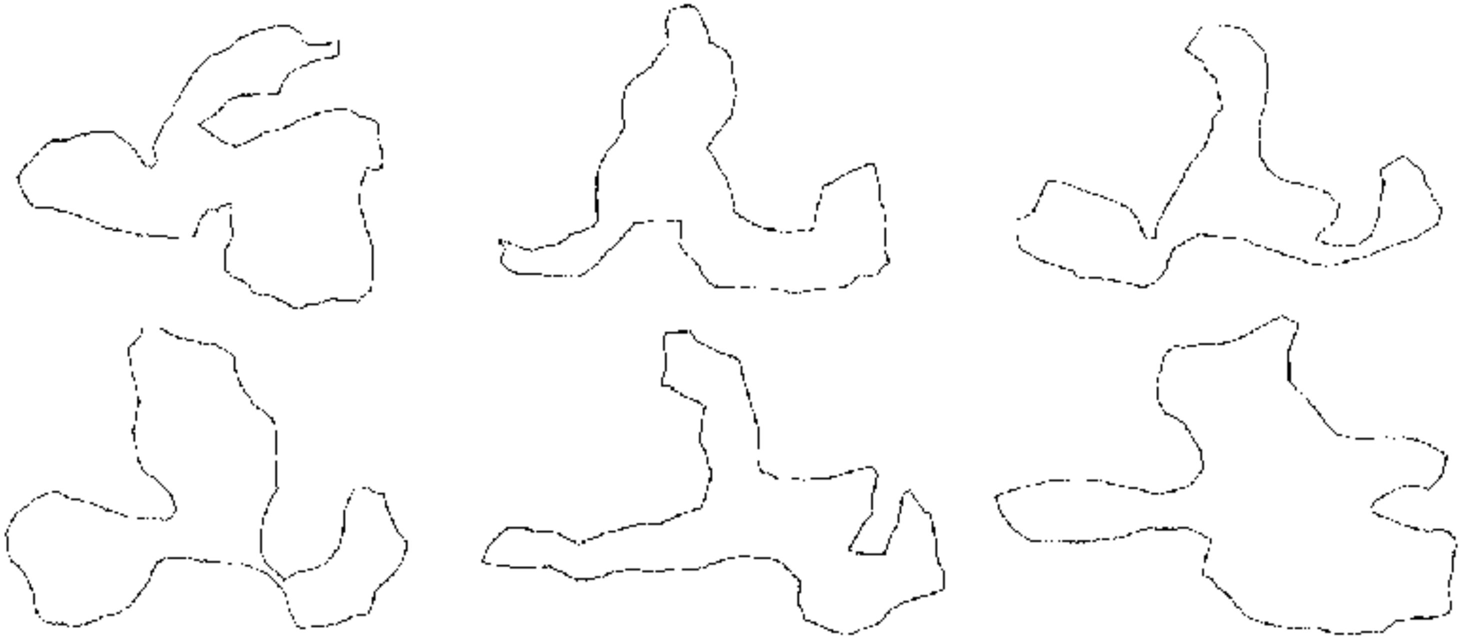


Fig. 11. Six of the synthesized shapes with curvaturo histogram matched to animate shapes, $\mu^{(\cdot)}_{syn} = \mu^{(\cdot)}_{obs}$. The histograms ot these synthesizod shapes aro shown by the dashed curves in Fig. 12.

Technique: Shot Boundary Detection

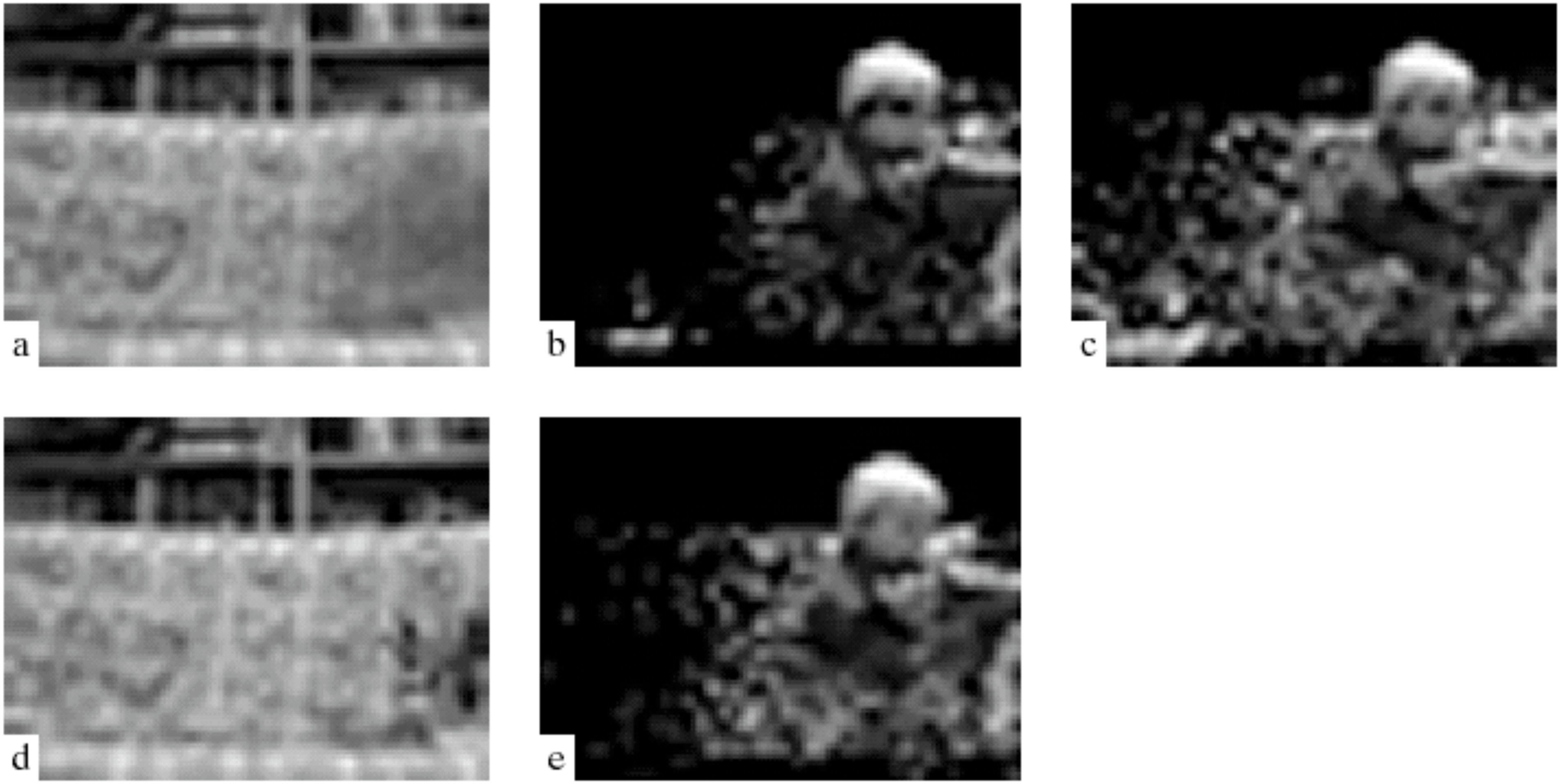
- Find the shots in a sequence of video
  - shot boundaries usually result in big differences between succeeding frames
- Strategy:
  - compute interframe distances
  - declare a boundary where these are big

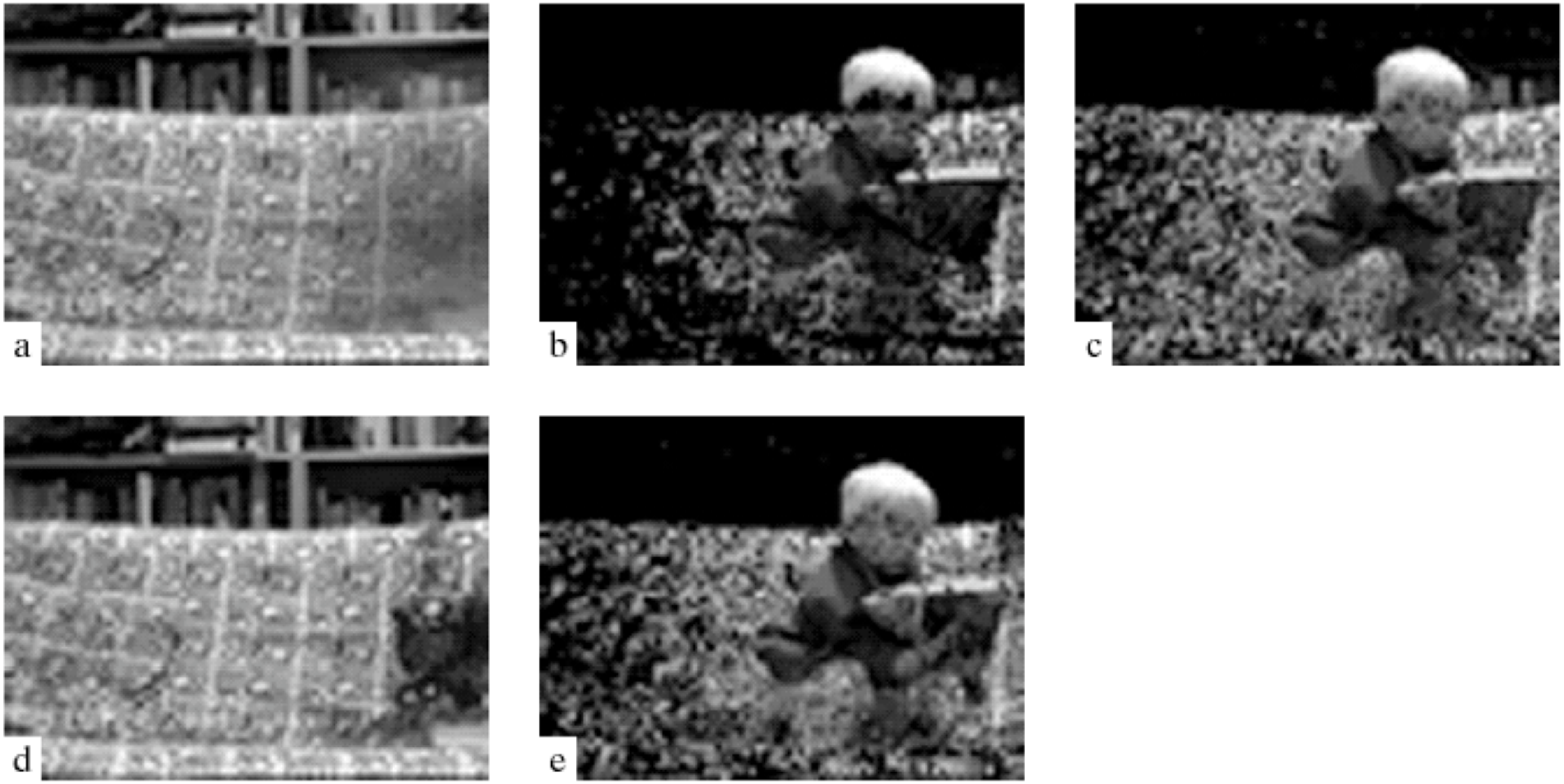
- Possible distances
  - frame differences
  - histogram differences
  - block comparisons
  - edge differences
- Applications:
  - representation for movies, or video sequences
    - find shot boundaries
    - obtain "most representative" frame
  - supports search

Technique: Background Subtraction

- If we know what the background looks like, it is easy to identify “interesting bits”
- Applications
  - Person in an office
  - Tracking cars on a road
  - surveillance

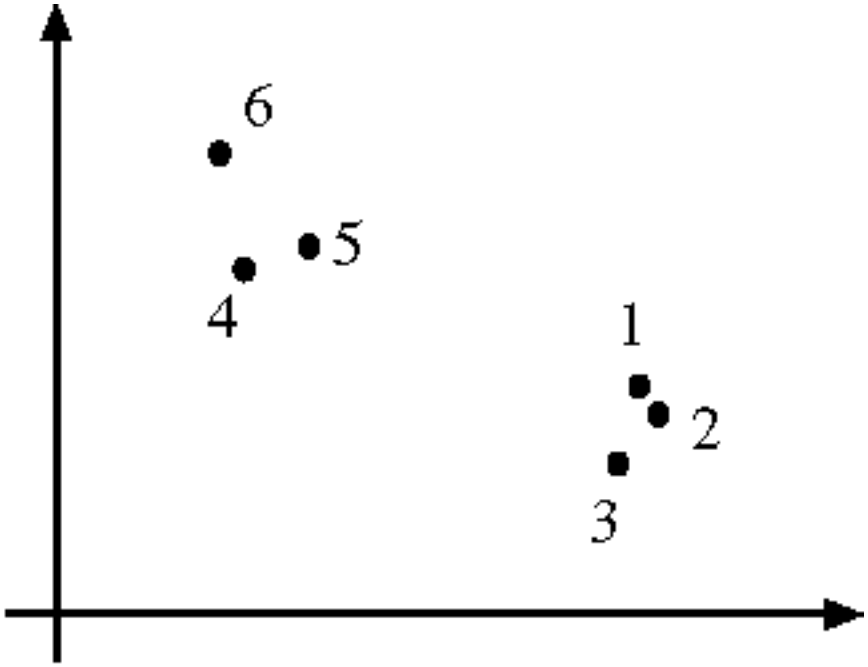
- Approach:
  - use a moving average to estimate background image
  - subtract from current frame
  - large absolute values are interesting pixels
    - trick: use morphological operations to clean up pixels

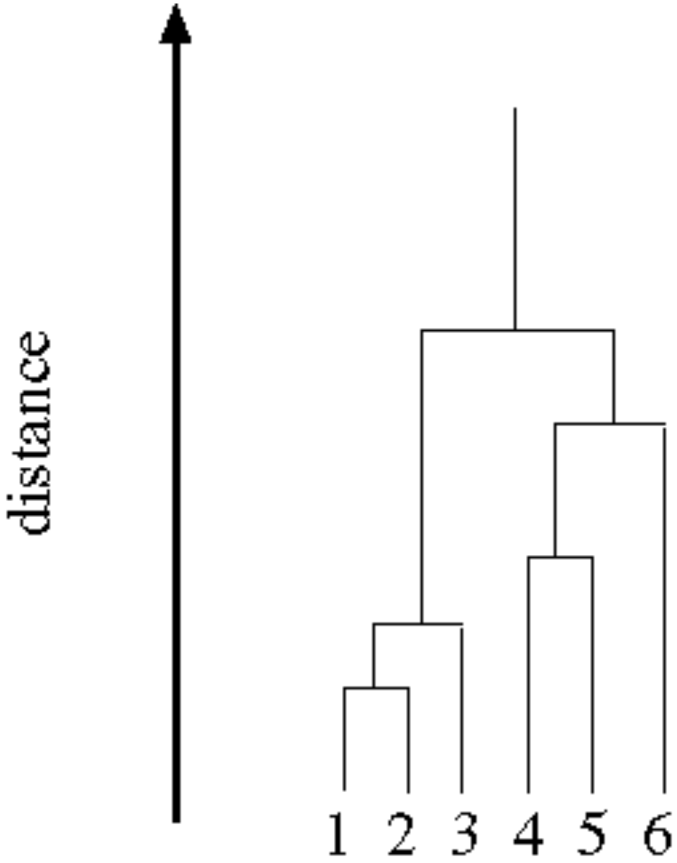

a
b
c
d
e


a
b
c
d
e

Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together
- Agglomerative clustering
  - attach closest to cluster it is closest to
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - Repeat
- Point-Cluster distance
  - single-link clustering
  - complete-link clustering
  - group-average clustering
- Dendrograms
  - yield a picture of output as clustering process continues


6
5
4
1
2
3


distance
1
2
3
4
5
6

K-Means

- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error
- can’t do this by search, because there are too many possible allocations.

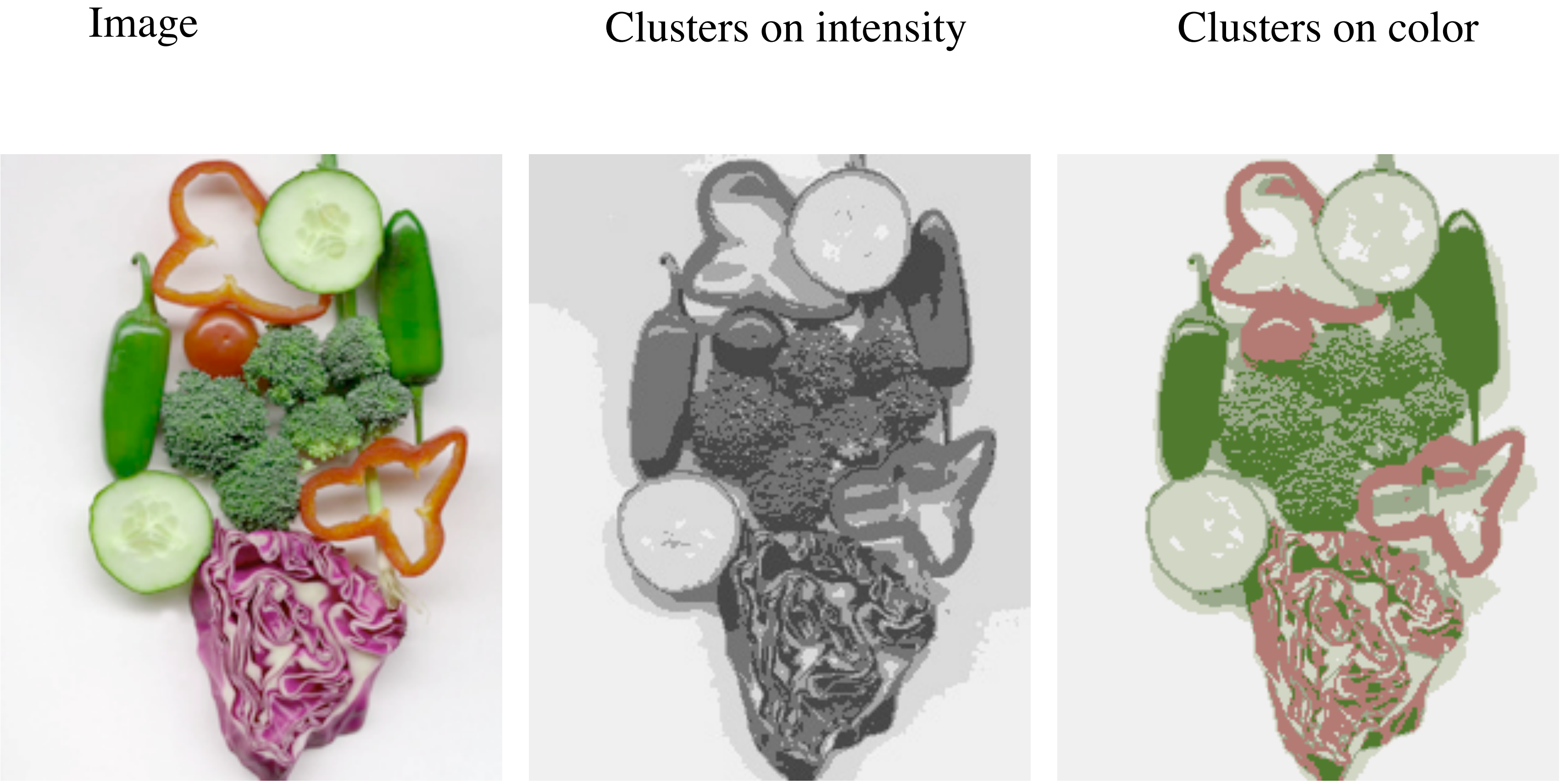
- Algorithm
  - fix cluster centers; allocate points to closest cluster
  - fix allocation; compute best cluster centers
- x could be any set of features for which we can compute a distance (careful about scaling)

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

Simple k-means algorithm

- *n* sample feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$
- Assume they fall into *k* compact clusters, $k < n$.
- Let $\boldsymbol{m}_i$ be the mean of the vectors in cluster i.
- Classify $\mathbf{x}$ in cluster i if $\| \mathbf{x} - \boldsymbol{m}_i \|$ is minimum of all the k distances.
- This suggests the following procedure for finding the k means:
  - Make initial guesses for the means $\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_k$
  - DO until no changes in any mean

```
  classify the samples into clusters by assigning  each point to the closest mean
  For i=1:k
          Replace $m_i$ mean of all of points in cluster i
  end
end_DO
```

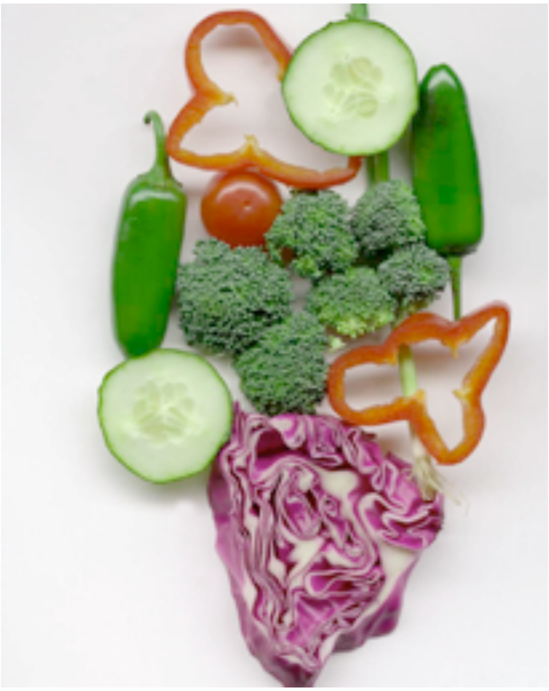


Image

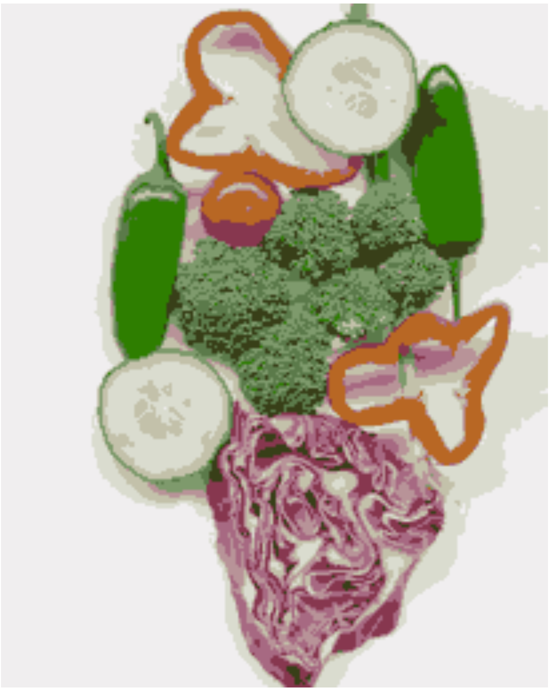
Clusters on intensity

Clusters on color

K-means clustering using intensity alone and color alone

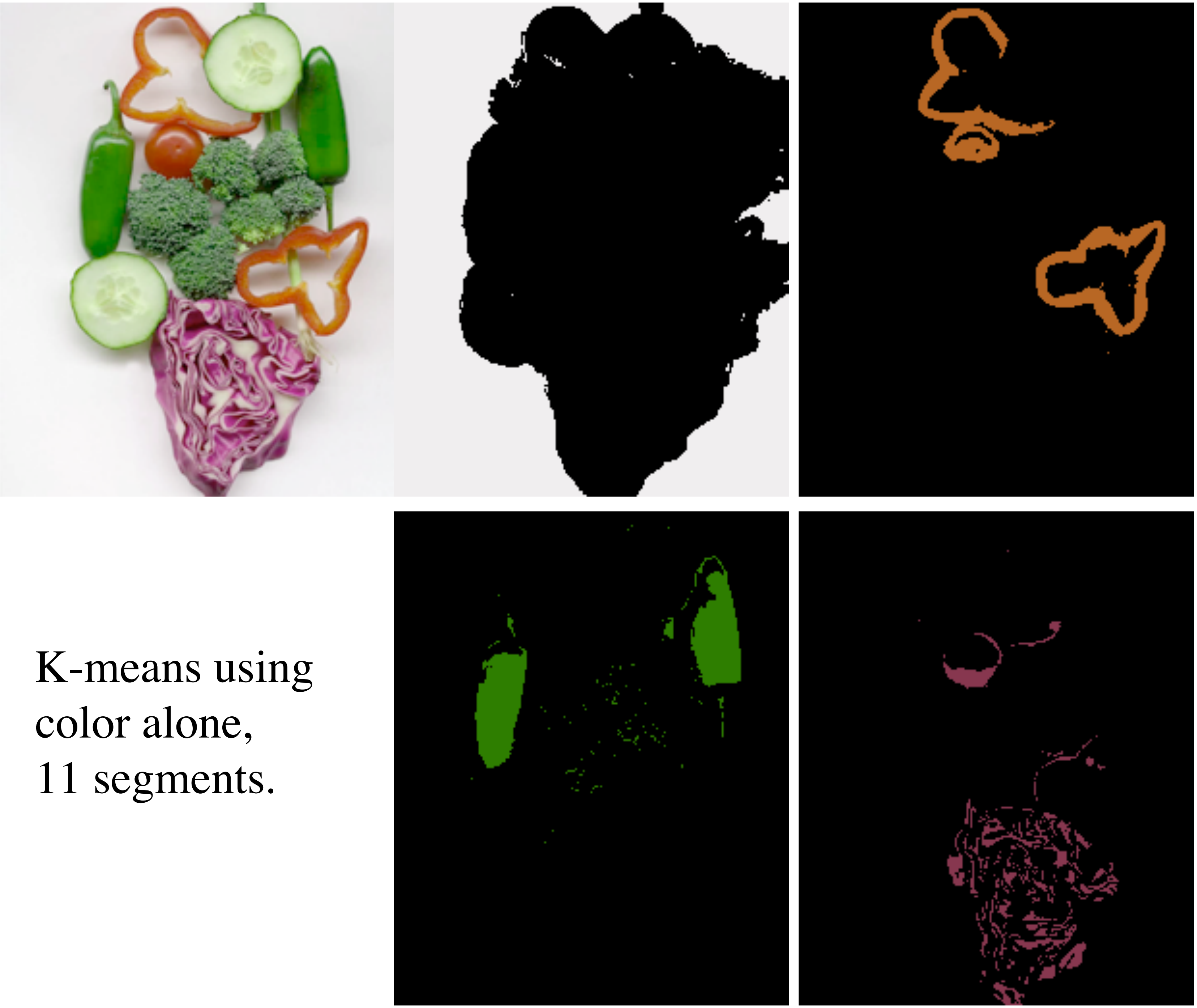


Image

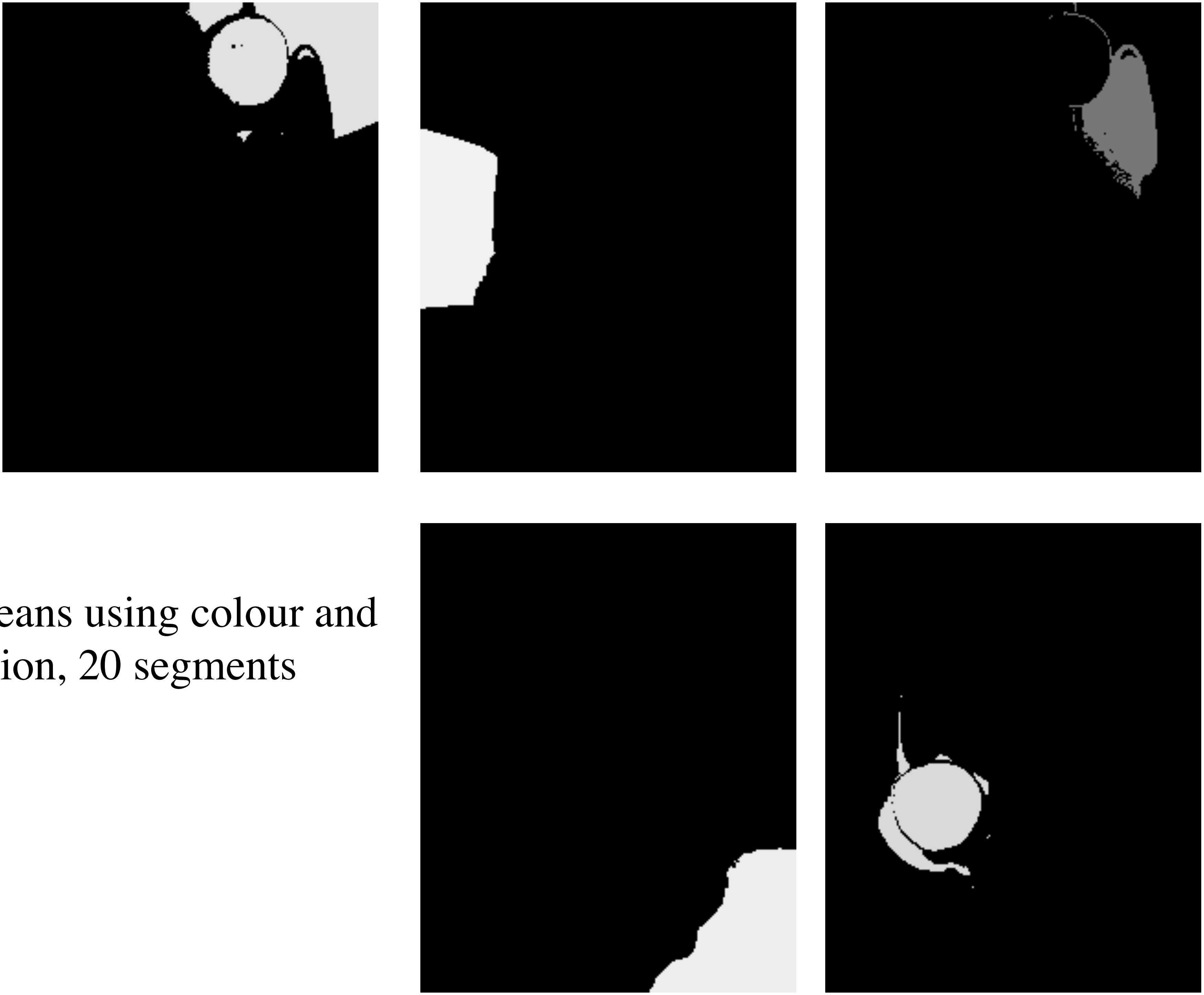


Clusters on color

K-means using color alone, 11 segments



K-means using color alone, 11 segments.



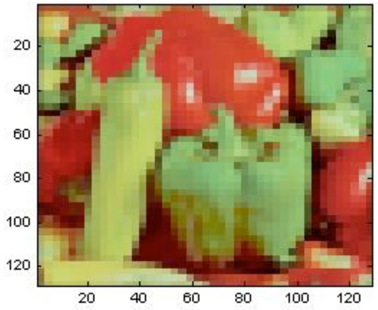
K-means using colour and position, 20 segments

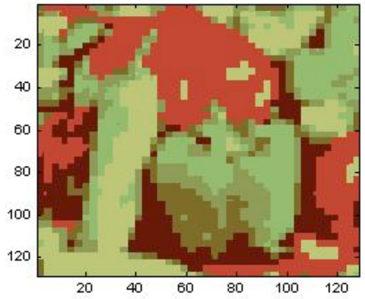
Graph theoretic clustering

- Represent tokens using a weighted graph.
  - affinity matrix
- Cut up this graph to get subgraphs with strong interior links

Clustering code

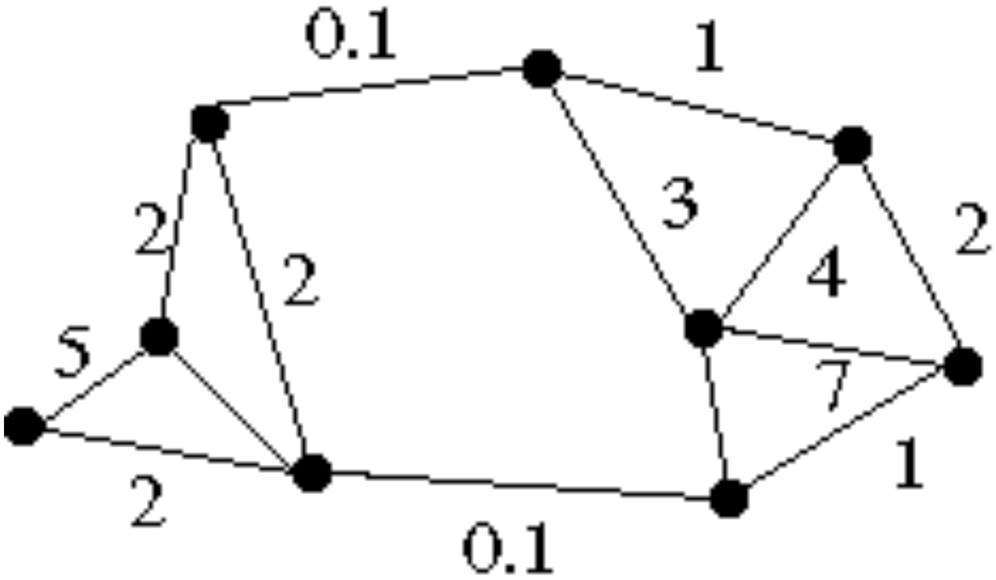
```
im=imread('squaresorig.jpg');
im2 = double(im(1:4:end,1:4:end,:));
Dm = size(im2);
X =reshape(im2, [Dm(1)*Dm(2), 3]);
[IDX,C]=kmeans(X,6);
X2 = C(IDX,:);
Clusterim = uint8(reshape(X2, [Dm(1), Dm(2), 3]));
image(Clusterim)
```

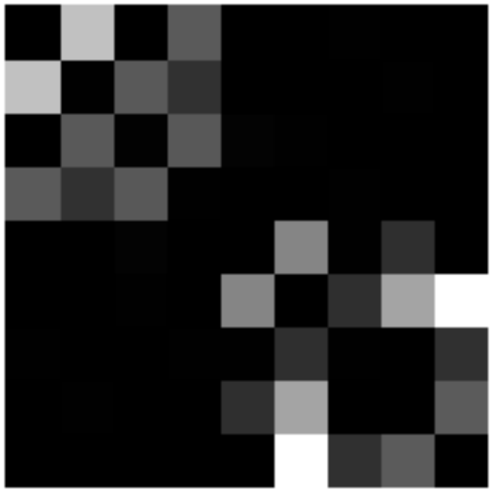


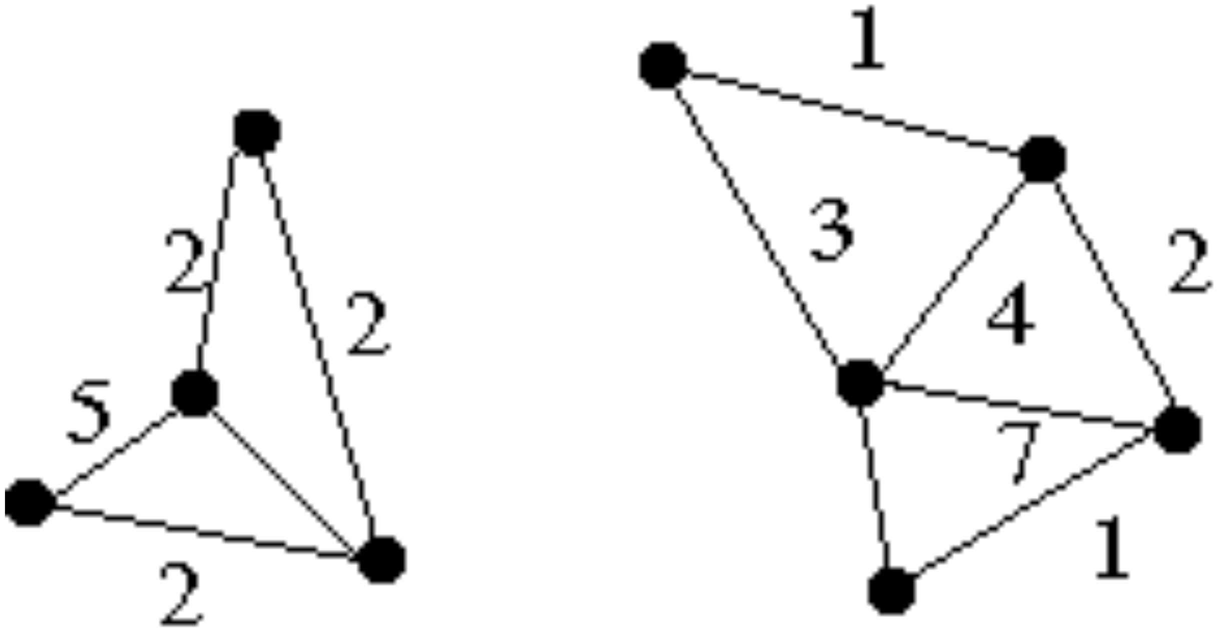


Assign mean color
To each cluster and display

Image Segmentation as Graph Partitioning

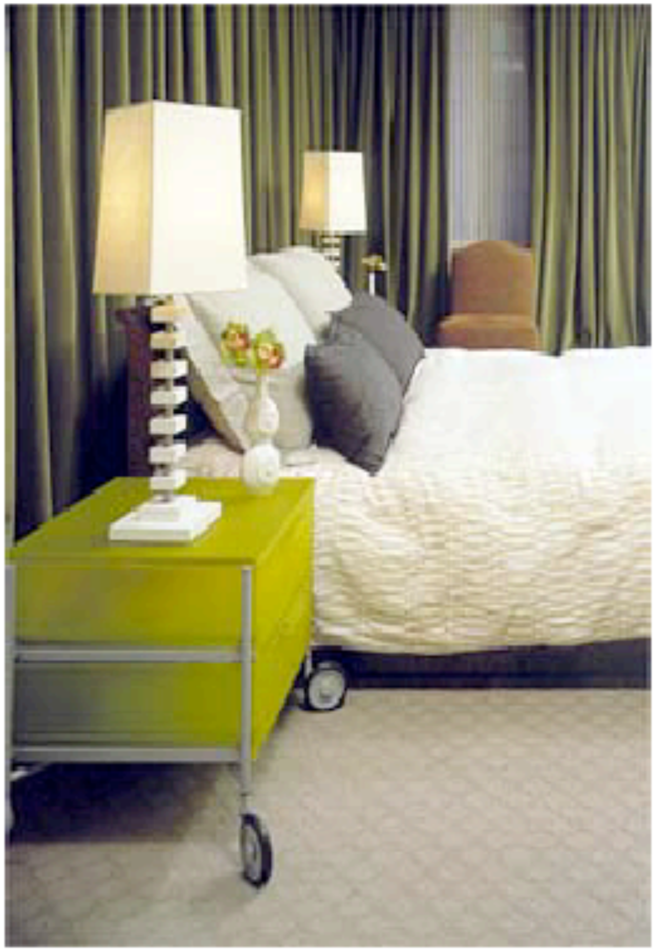

0.1
1
2
3
2
2
2
4
5
7
2
1
0.1




2
2
5
2
1
3
2
4
7
1

Boundaries of image regions defined by a number of attributes

- Brightness/color
- Texture
- Motion
- Stereoscopic depth
- Familiar configuration



Measuring Affinity

Intensity

$$aff(x, y) = \exp\left\{-\left({}^{1}\!\big/_{2\sigma_i^2} \right)\left(\|I(x) - I(y)\|^2 \right)\right\}$$

Distance

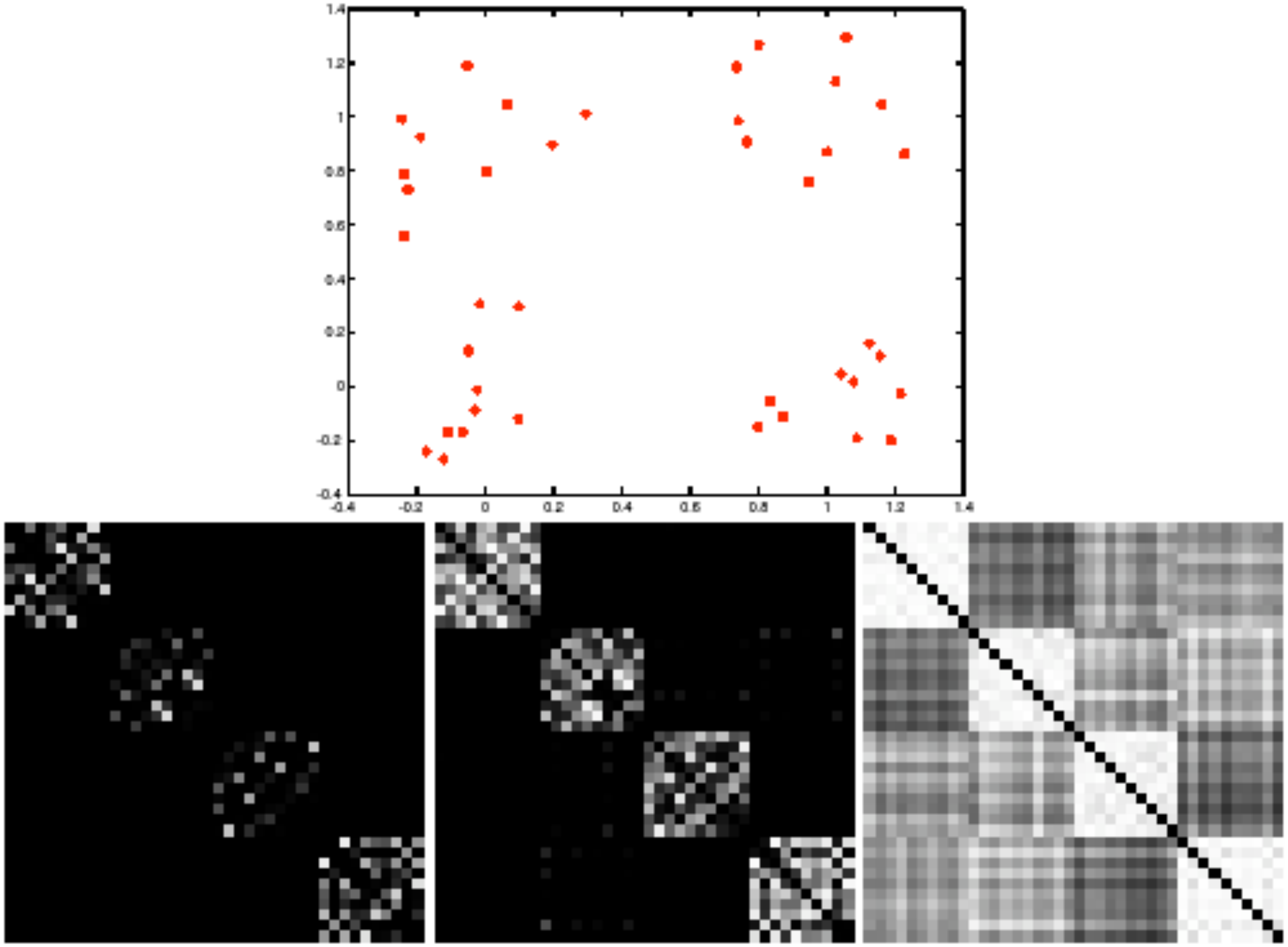
$$aff(x, y) = \exp\left\{-\left({}^{1}\!\big/_{2\sigma_d^2} \right)\left(\|x - y\|^2 \right)\right\}$$

Texture

$$aff(x, y) = \exp\left\{-\left({}^{1}\!\big/_{2\sigma_t^2} \right)\left(\|c(x) - c(y)\|^2 \right)\right\}$$

c(x) denotes a histogram,for instance

Scale affects affinity



Eigenvectors and cuts

- Simplest idea: we want a vector a giving the association between each element and a cluster
- We want elements within this cluster to, on the whole, have strong affinity with one another

$Let \quad w_{ij} = aff(x_i, x_j)$ for pixels x_i & x_j

- We could maximize

$$E = a^T W a$$

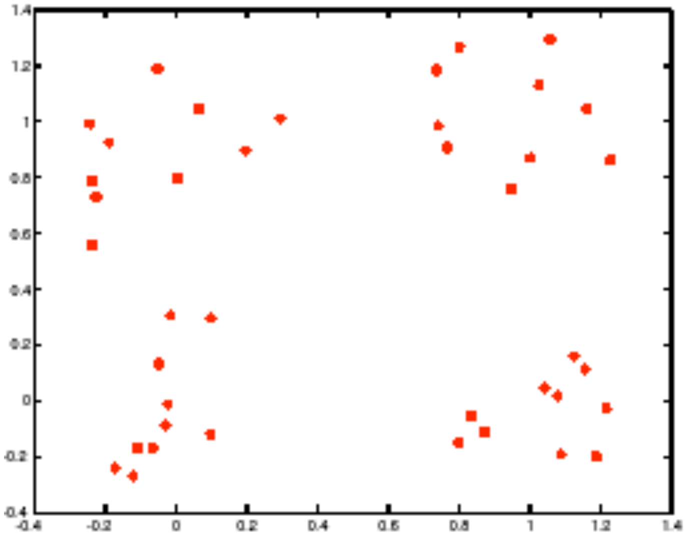
- But need the constraint

$$a^T a = 1$$

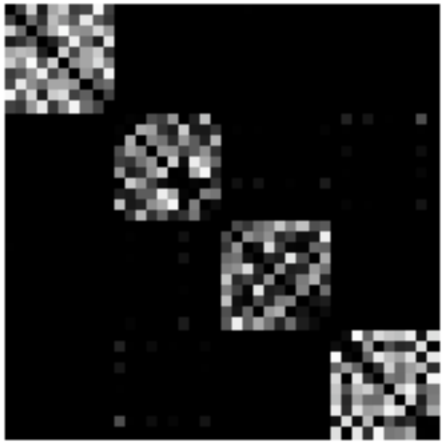
- This is an eigenvalue problem - choose the eigenvector of A with largest eigenvalue

$$Maximize:$$
$$E = a^T W a + \lambda(1 - a^T a)$$
$$\text{Set}: \quad \frac{\partial E}{\partial a} = 0$$
$$= 2Wa - 2\lambda a$$
$$\Rightarrow Wa = \lambda a$$

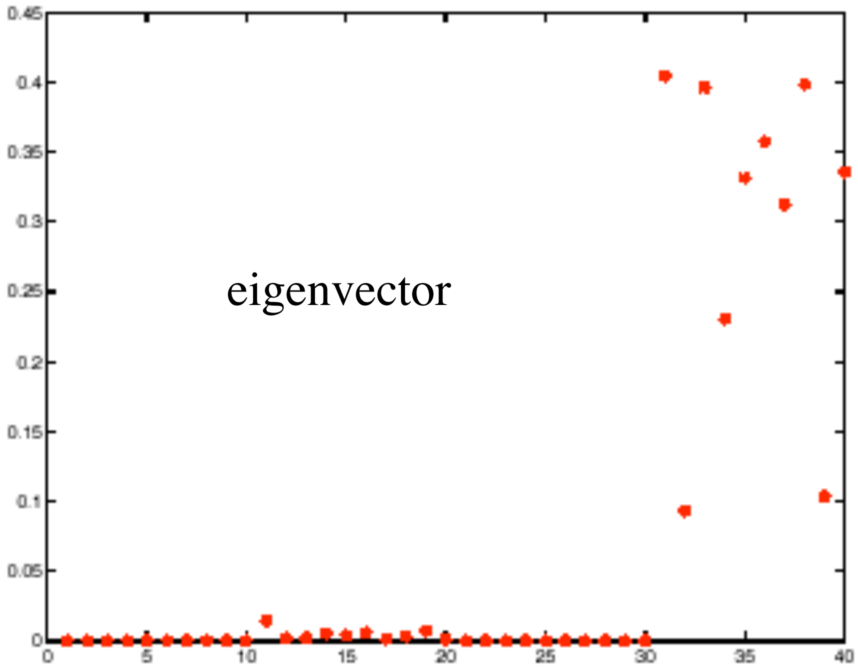
Example eigenvector



points



matrix



eigenvector

More than two segments

- Many options
  - Recursively split each side to get a tree, continuing till the eigenvalues are too small
  - Use the other eigenvectors

Normalized cuts

- Current criterion evaluates within cluster similarity, but not across cluster difference
- Instead, we'd like to maximize the within cluster similarity compared to the across cluster difference
- Write graph as V, one cluster as A and the other as B

$$cut(A,B) =$$

$$assoc(A,B) = \sum_{i \in A,\, j \in B} W(i,j)$$

- Maximize

$$Nassoc(A,B) =$$

$$\left(\frac{assoc(A,A)}{assoc(A,V)}\right) + \left(\frac{assoc(B,B)}{assoc(B,V)}\right)$$

- i.e. construct A, B such that their within cluster similarity is high compared to their association with the rest of the graph

Normalized cuts

- Write a vector y whose elements are :

  1 if item is in A,

  -b if it's in B

  (b = fraction(in)/fraction(out))

- Write the affinity matrix of the graph as W, and the matrix which has the row sums of W on its diagonal as D, 1 is the vector with all ones.
- Criterion becomes

$$\min_{\mathrm{y}}\left(\frac{y^T(D-W)y}{y^T Dy}\right)$$

- and we have a constraint

$$y^T D1 = 0$$

- This is hard to do, because y's values are quantized

Binary Case

If 10 of 100 in A

Then 90 in B

b = 10/90

Normalized cuts

- Instead, solve the generalized eigenvalue problem

$$\max_y \left(y^T (D - W) y \right) \text{ subject to } \left(y^T D y = 1 \right)$$

- which gives

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

- Now look for a quantization threshold that maximises the criterion --- i.e all components of y above that threshold go to one,
- all below go to -b

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, $x_i = 1$ 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 $\boldsymbol{i}$ to all other nodes. With the definitions $\boldsymbol{x}$ and $\boldsymbol{d}$ we can rewrite $Ncut(A,B)$ as:

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

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 $\mathrm{W(i,j)} = w_{ij}$, $k = \frac{\sum_{x_i>0} \boldsymbol{d}_i}{\sum_i \boldsymbol{d}_i}$, 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 $x_i > 0$ and $x_i < 0$ respectively, we can rewrite $4[Ncut(\boldsymbol{x})]$ as:

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

$$b = \frac{k}{1-k},$$

Setting $\boldsymbol{y} = (\mathbf{1} + \boldsymbol{x}) - b(\mathbf{1} - \boldsymbol{x}),$

Shi & Malik(1997) show:

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D} - \mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}},$$

with the condition $\boldsymbol{y}(i) \in \{1, -b\}$ and $\boldsymbol{y}^T\mathbf{D}\mathbf{1} = 0.$

This is Equivalent to:

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}\boldsymbol{z} = \lambda \boldsymbol{z},$$

$$\mathbf{D}^{-\frac{1}{2}}\mathbf{D}\mathbf{D}^{-\frac{1}{2}}z - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}z = \lambda z$$

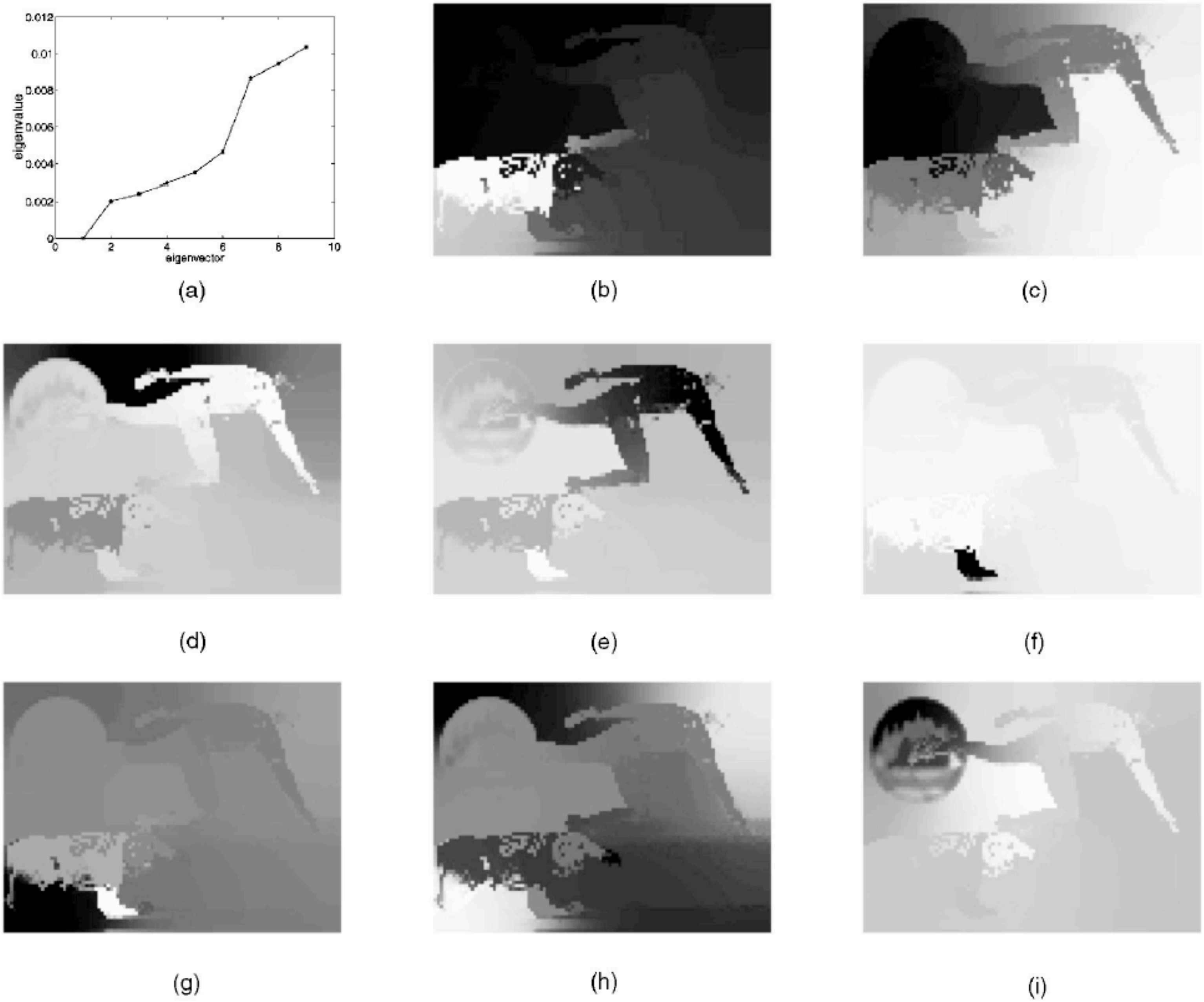
$$\boldsymbol{z} = \mathbf{D}^{\frac{1}{2}}\boldsymbol{y}.$$

$$\mathbf{I}z - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}z = \lambda z$$

But solutions to above are solutions to :

$$\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}z = \lambda z$$

Using Intensity



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 3. Subplot (a) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplots (b)-(i) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

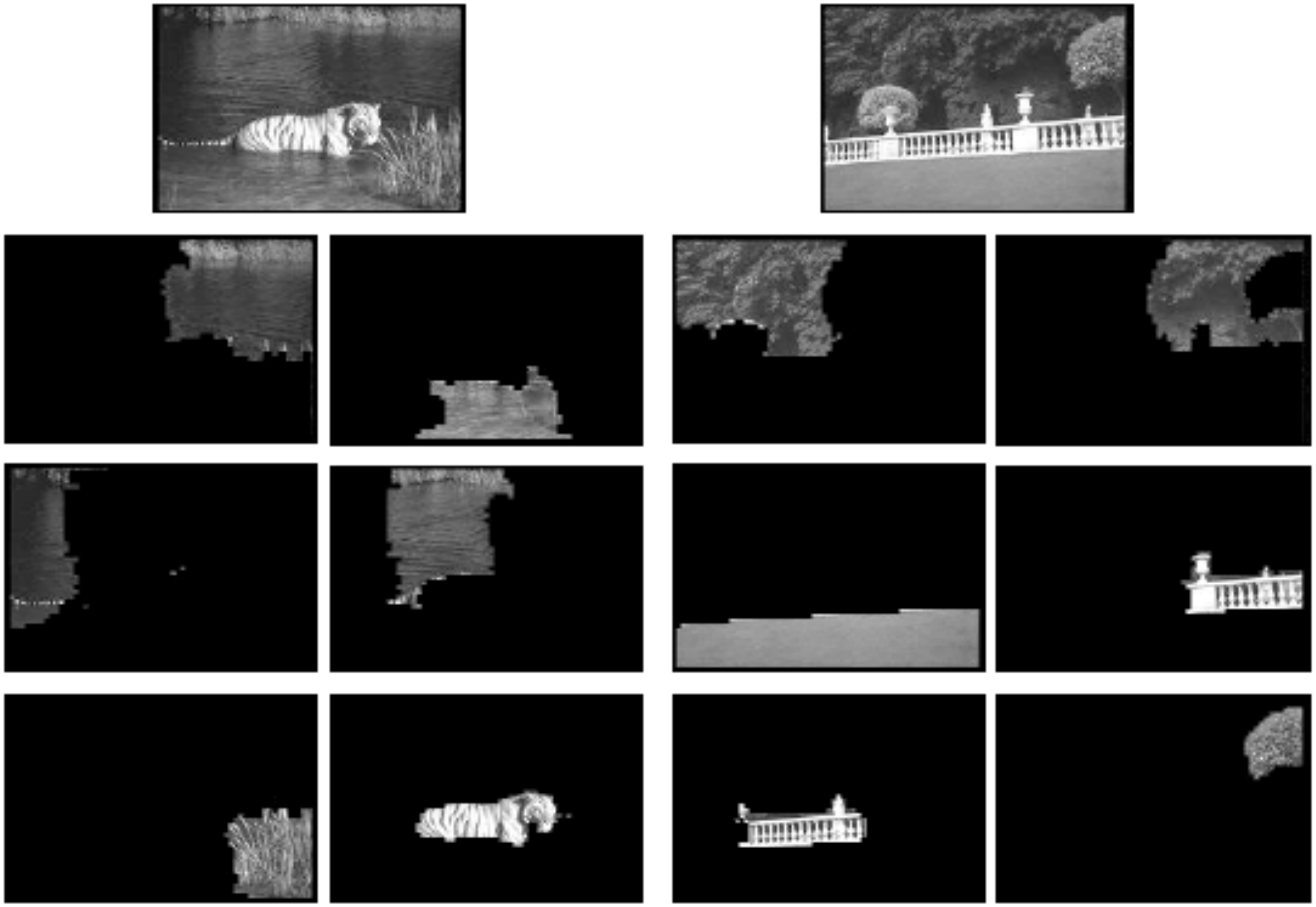
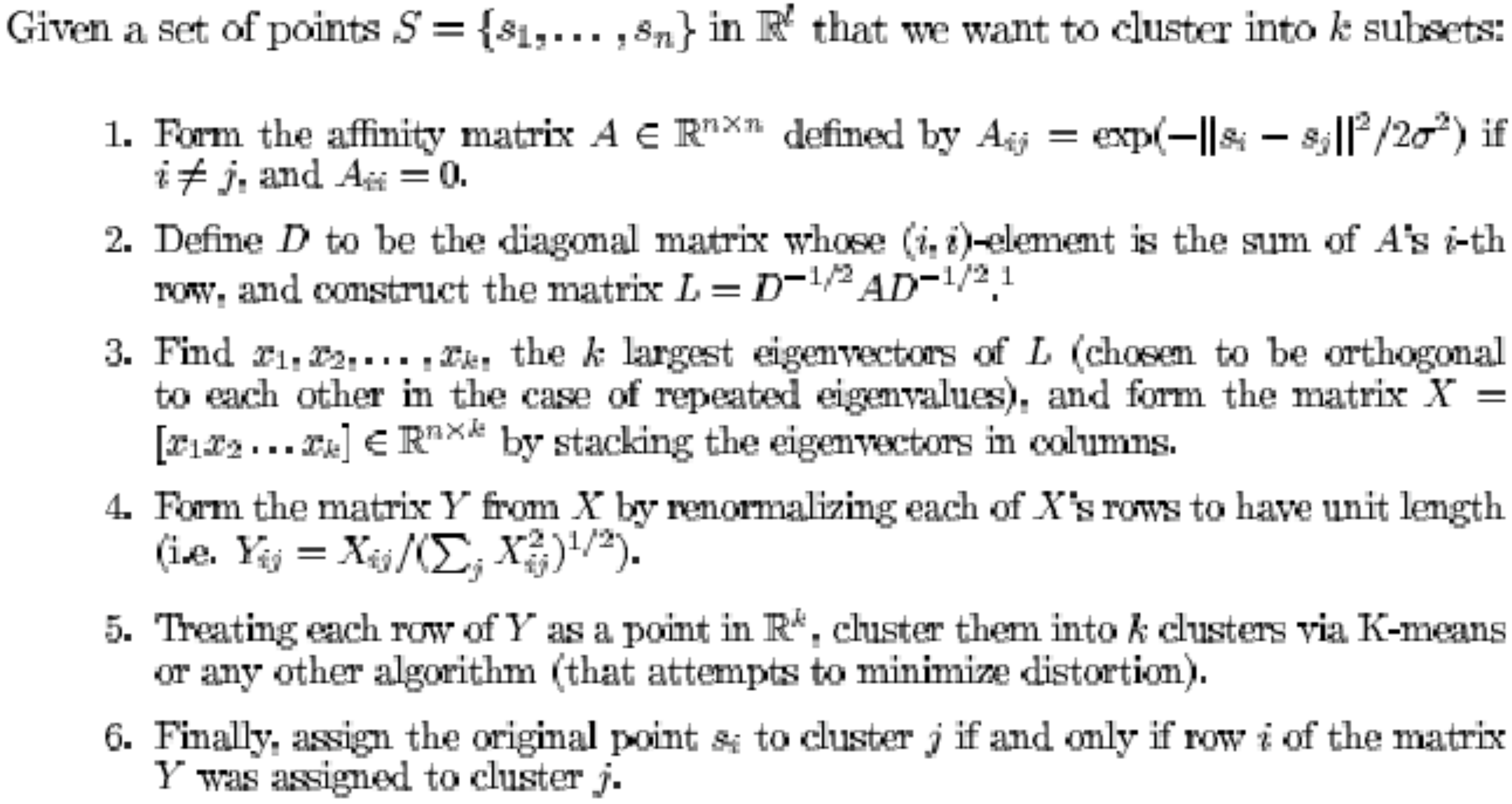


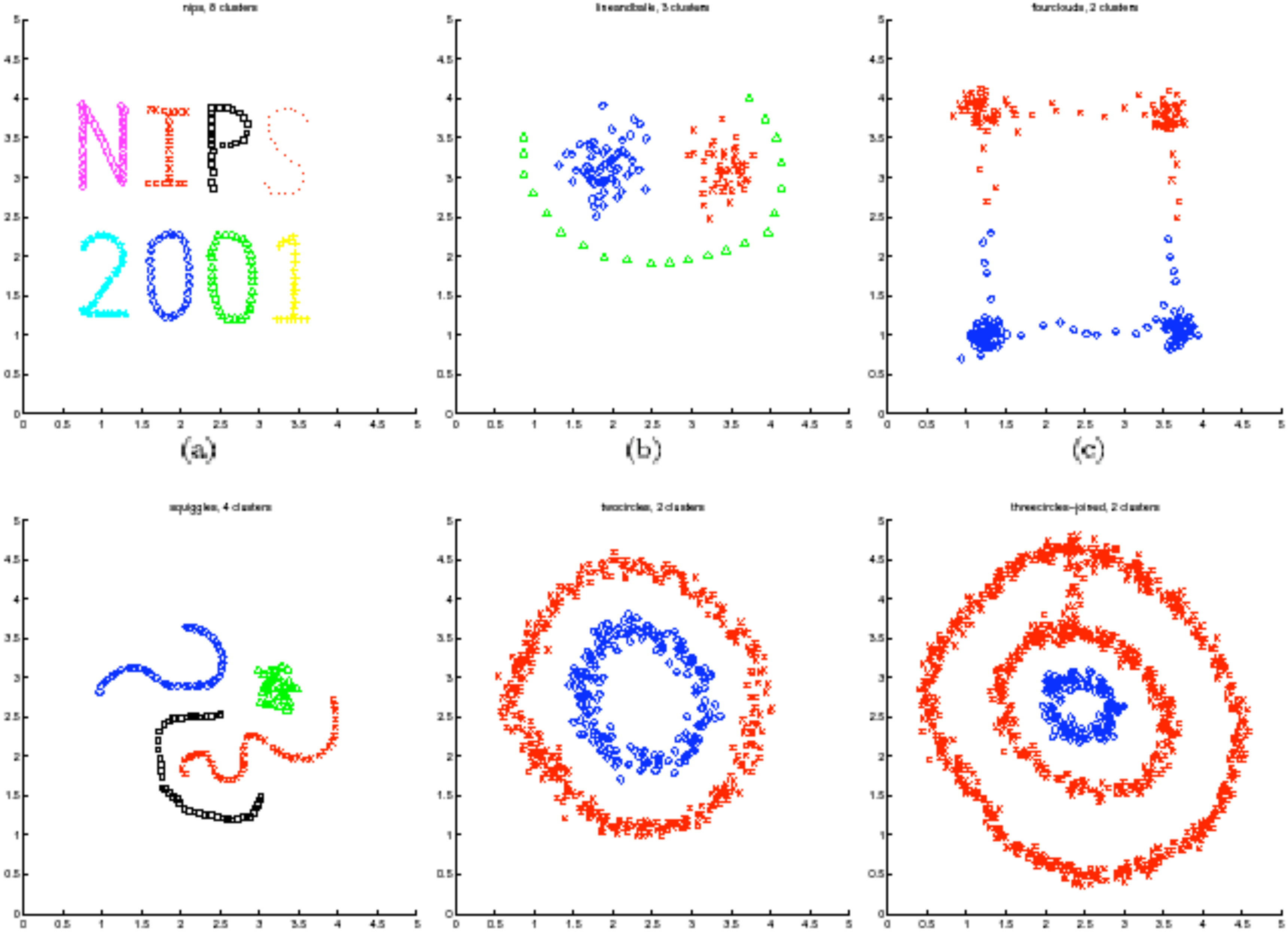
Figure from “Image and video segmentation: the normalised cut framework”, by Shi and Malik, copyright IEEE, 1998

Ng,Jordan,Weiss, 2002



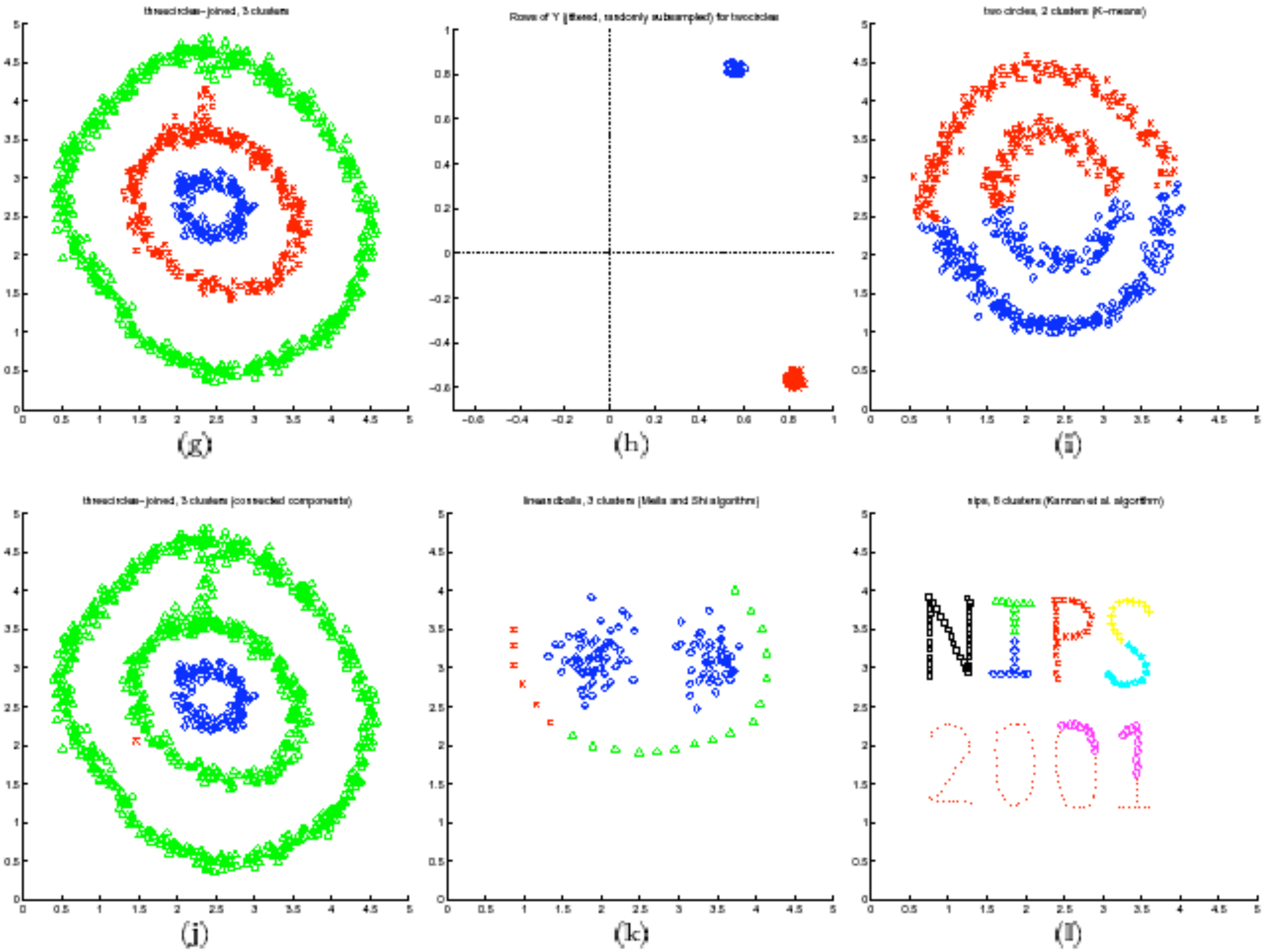
Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-\|s_i - s_j\|^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i, i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.


nips, 8 clusters
lineandballs, 3 clusters
fourclouds, 2 clusters
squiggles, 4 clusters
twocircles, 2 clusters
threecircles-joined, 2 clusters
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

(a) (b) (c)

(d) (e) (f)



threecircles-joined, 3 clusters

(g)

Rows of Y (jittered, randomly subsampled) for twocircles

(h)

two circles, 2 clusters (K-means)

(i)

threecircles-joined, 3 clusters (connected components)

(j)

lineandballs, 3 clusters (Meila and Shi algorithm)

(k)

nips, 8 clusters (Kannan et al. algorithm)

(l)

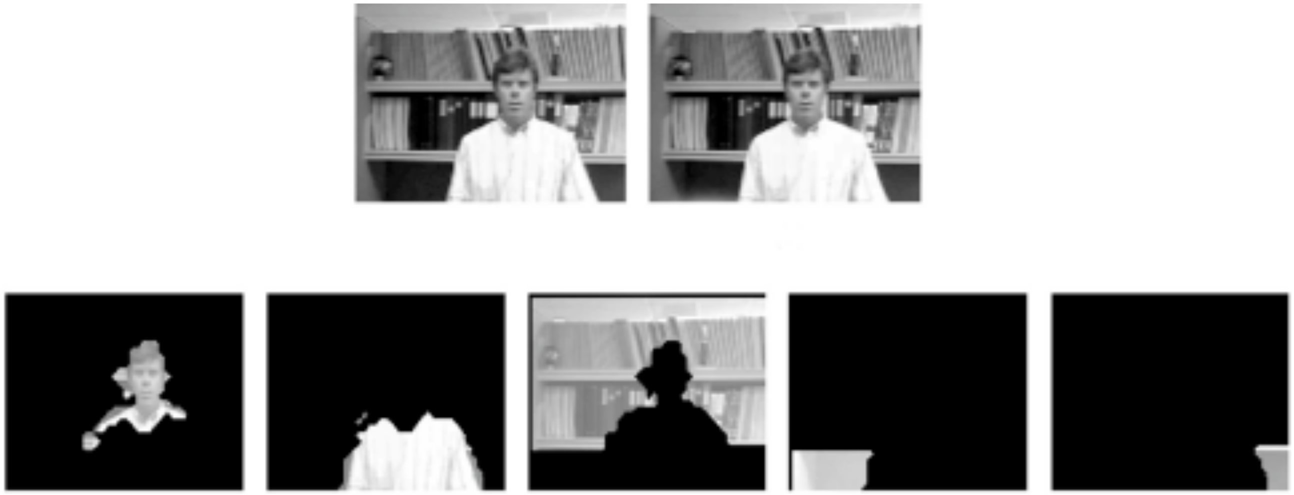


Figure from “Normalized cuts and image segmentation,” Shi and Malik, copyright IEEE, 2000