AARHUS
UNIVERSITET

Computer Vision & Machine Learning

Alexandros Iosifidis
@
Department of Electrical and Computer Engineering
Aarhus University

This week

Image segmentation:

- Dividing (or partitioning) an image into its constituent regions (or objects)
- Cluster-based segmentation
  - K-means clustering
  - Mean shift
  - Graph-based clustering
- Boundary-based segmentation

Image segmentation

Segmentation as clustering:

- Decompose the image into regions that have roughly coherent color and texture.
- Shape is not important – but coherence is!
- Applications:
  - Image compression
  - Finding superpixels
  - Image correspondence (optical flow and registration)
  - Finding repetitions (like windows on the facade of a building)
  - Recognition (for instance, human arms and legs tend to be long and straight)

Image segmentation

Basic clustering methods:

- We represent each pixel with a feature vector
- Natural feature vectors include
  - Pixel intensity
  - Pixel color
  - Pixel location
  - Pixel response to filter bank (= local texture)
- We cluster these feature vectors
  - Each pixel belongs to exactly one cluster (group of similar vectors)
- We obtain the image segment represented by a cluster by replacing the feature vector at each pixel with the number of that feature vector’s cluster center.

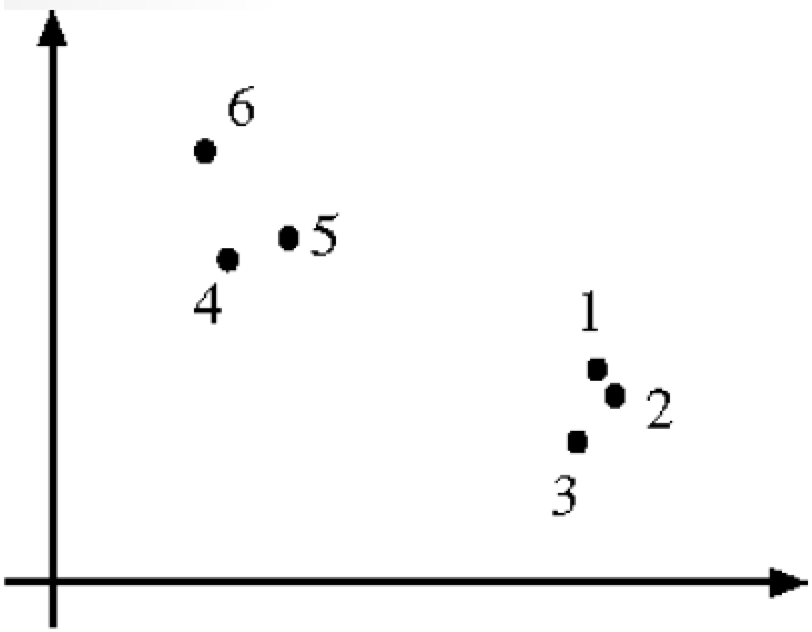
Clustering

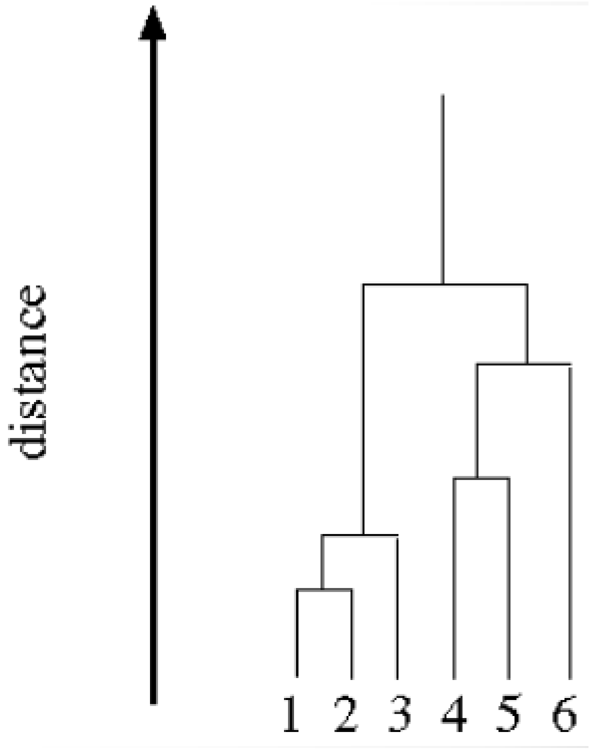
Bottom-up clustering approach (agglomerative clustering or clustering by merging)
- Make each point a separate cluster
- Until the clustering is satisfactory
  - Merge the two clusters with the smallest inter-cluster distance

Top-down clustering approach (divisive clustering, or clustering by splitting)
- Construct a single cluster containing all points
- Until the clustering is satisfactory
  - Split the cluster that yields the two components with the largest inter-cluster distance

Clustering

Example of bottom-up clustering





Clustering

Issues:

- What is a good inter-cluster distance?
  - Distance between closest elements.
  - Maximum distance between an element of the first cluster and of the second.
  - Average of distances between elements in the cluster.
- What is a good distance measure?
  - Euclidian distance
  - Normalized correlation coefficient
- How many clusters are there?

Clustering-based segmentation

Each pixel is described by a vector

$$\mathbf{z} = [r,g,b]^T \quad \text{or} \quad \mathbf{z} = [Y,u,v]^T$$

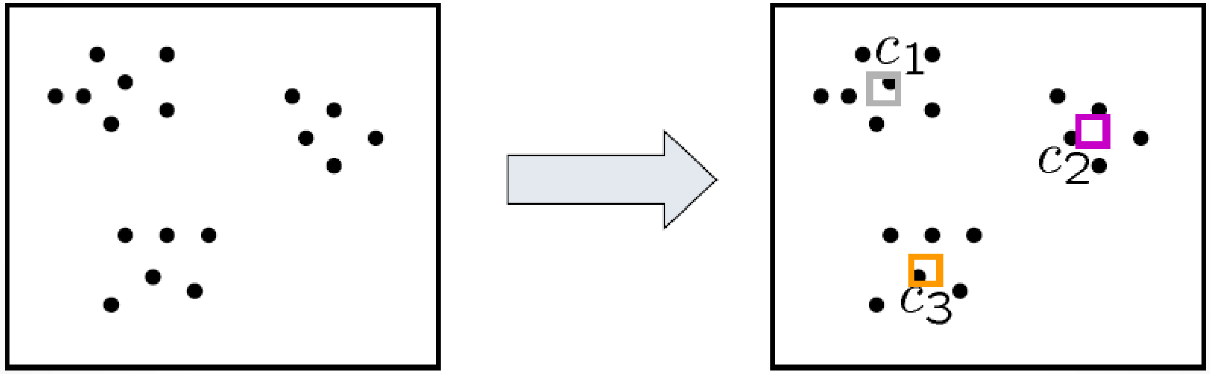
Using all z_i, i=1,…,HW (for an image of HxW pixels) run a clustering algorithm (e.g. K-Means) using the Euclidean distance between pixels:

$$D(\text{pixel}_i,\text{pixel}_j) = \| \mathbf{z}_i - \mathbf{z}_j \|^2$$

Clustering-based segmentation

K-Means algorithm:

- An iterative process formed by two steps (at each iteration)
  - Given a set of prototypes (cluster mean vectors), assign each point to a cluster based on its distance to these prototypes
  - Given the cluster labels assigned above, calculate the best prototypes



Clustering-based segmentation

K-Means algorithm:

- An iterative process formed by two steps (at each iteration)
  - Given a set of prototypes (cluster mean vectors), assign each point to a cluster based on its distance to these prototypes
  - Given the cluster labels assigned above, calculate the best prototypes
- It will always converge to a solution
- The solution will correspond to a local minimum of the objective function

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

Clustering-based segmentation

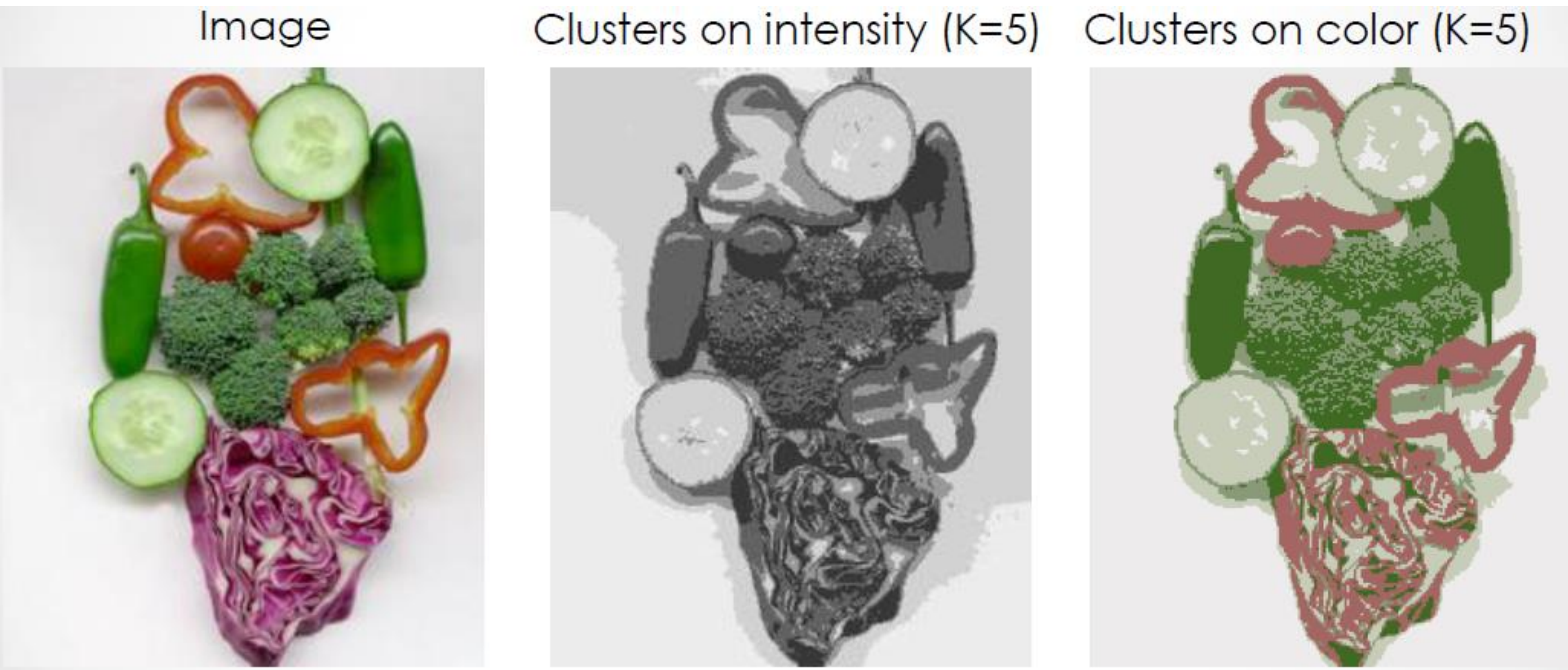
Algorithm 15.5: Clustering by K-Means

Choose k data points to act as cluster centers
Until the cluster centers are unchanged
    Allocate each data point to cluster whose center is nearest
    Now ensure that every cluster has at least
        one data point; possible techniques for doing this include .
        supplying empty clusters with a point chosen at random from
        points far from their cluster center.
    Replace the cluster centers with the mean of the elements
        in their clusters.
end

Clustering-based segmentation

Examples

- K-means clustering using intensity alone and color alone



Clustering-based segmentation

Examples

- K-means using color alone, 11 segments

Image

Clusters on color

Clustering-based segmentation

Examples

- K-means using color alone (11 segments)
- Color alone often will not yield salient segments

One segment may represent many objects

The absence of a texture measure creates serious difficulties.

Clustering-based segmentation

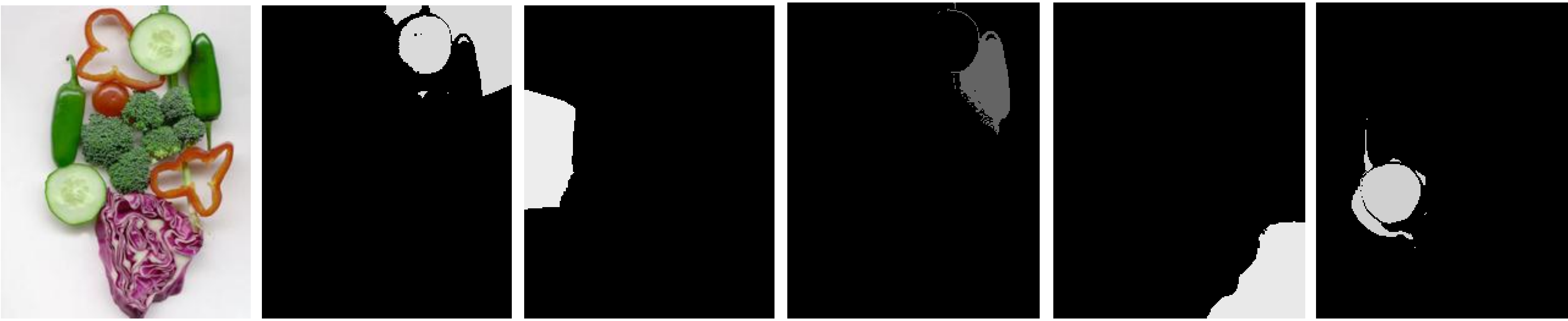
One solution:

- Include spatial relationships
- Augment data to be clustered with spatial coordinates
- We need to be careful on how to 'scale' each coordinate!

$$z = \begin{pmatrix} r \\ g \\ b \\ x \\ y \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{color coordinates} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{spatial coordinates} \end{matrix}$$

Clustering-based segmentation

Examples
- K-means using color and position (20 segments)
- Still misses goal of perceptually pleasing segmentation



Peppers are better separated, but large background regions that should be coherent have been broken up because points got too far from the center.

Clustering-based segmentation

K-Means algorithm:

- An iterative process formed by two steps (at each iteration)
  - Given a set of prototypes (cluster mean vectors), assign each point to a cluster based on its distance to these prototypes
  - Given the cluster labels assigned above, calculate the best prototypes
- It will always converge to a solution
- The solution will correspond to a local minimum of the objective function

$$\sum_{\text{clusters } i} \; \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

- Questions:
  - what's the probability that a point **x** is in cluster m?
  - what's the shape of each cluster?
- K-means doesn't answer these questions
  - Solution: Probabilistic clustering, such as Mixture of Gaussians.

Clustering-based segmentation

Mixture of Gaussians

- each cluster center is augmented by a covariance matrix whose values are re-estimated from the corresponding samples.
- Instead of using nearest neighbors to associate input samples with cluster centers, a Mahalanobis distance is used.
- In this way, samples can be softly assigned to several nearby clusters.

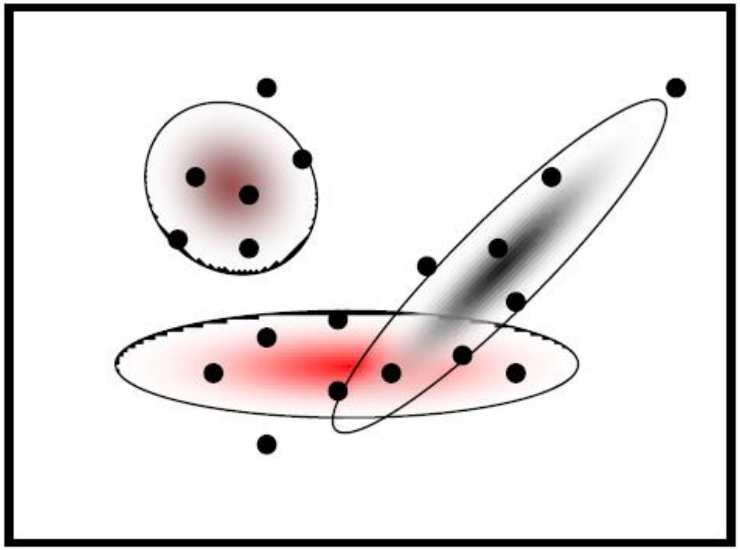


Image segmentation

Mean-shift based segmentation

- Think of the image as a sample from a multi-modal probability distribution
  https://en.wikipedia.org/wiki/Multimodal_distribution
- The density can be modelled using a kernel-based approach (see Szeliski, Eq. 5.34)
- Mean shift segmentation:
  - Assign each pixel to the nearest peak of the multi-modal density function.

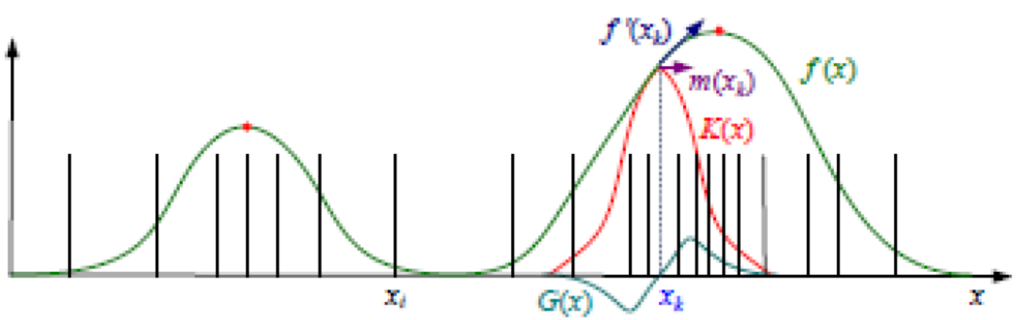


Image segmentation

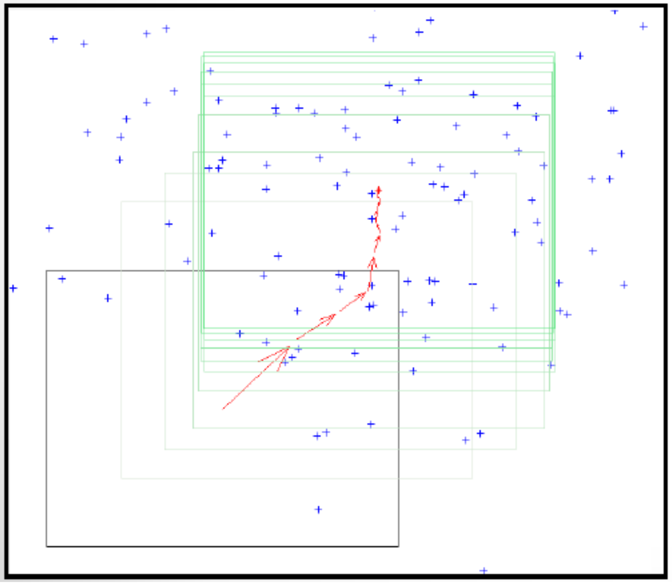
Mean-shift based segmentation

1. Convert each pixel into a feature vector (via color, location, etc.).
2. Choose initial search window locations uniformly in the data.
3. Compute the mean shift window location for each initial position.
4. Merge windows that end up on the same “peak” or mode.
5. The data these merged windows traversed are clustered together.

Image segmentation

Mean-shift based segmentation

1. Convert each pixel into a feature vector (via color, location, etc.)
2. Choose initial search window locations uniformly in the data
3. Compute the mean shift window location for each initial position
4. Merge windows that end up on the same “peak” or mode
5. The data these merged windows traversed are clustered together

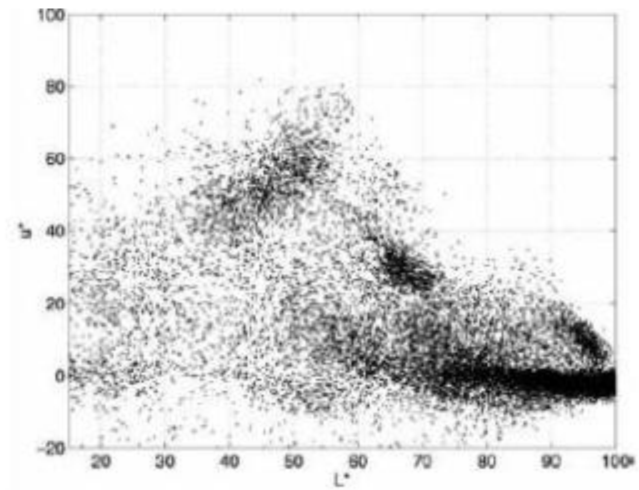


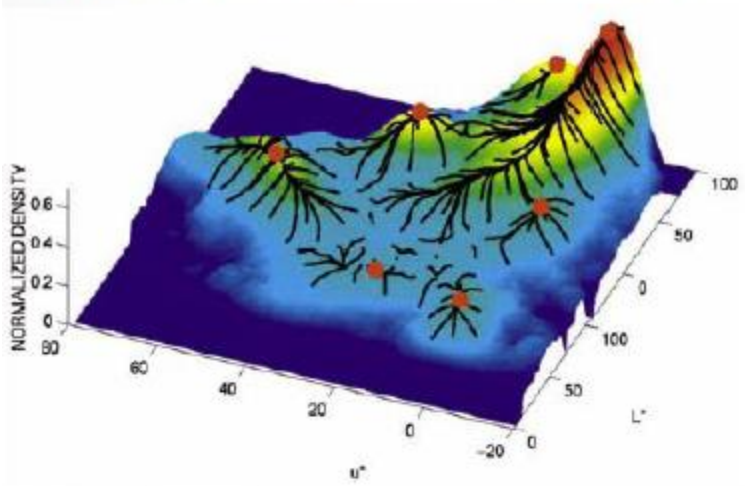
The mean shift algorithm seeks the “mode” or point of highest density of a data distribution

Two issues:

- Kernel to interpolate density based on sample positions.
- Gradient ascent to mode.

Image segmentation





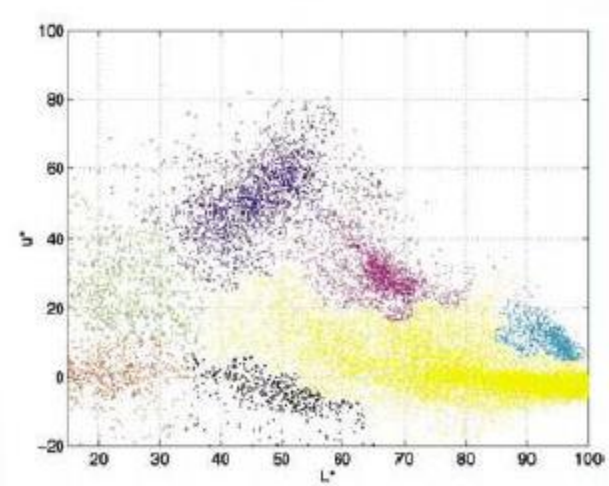


Image segmentation

Mean-shift based segmentation

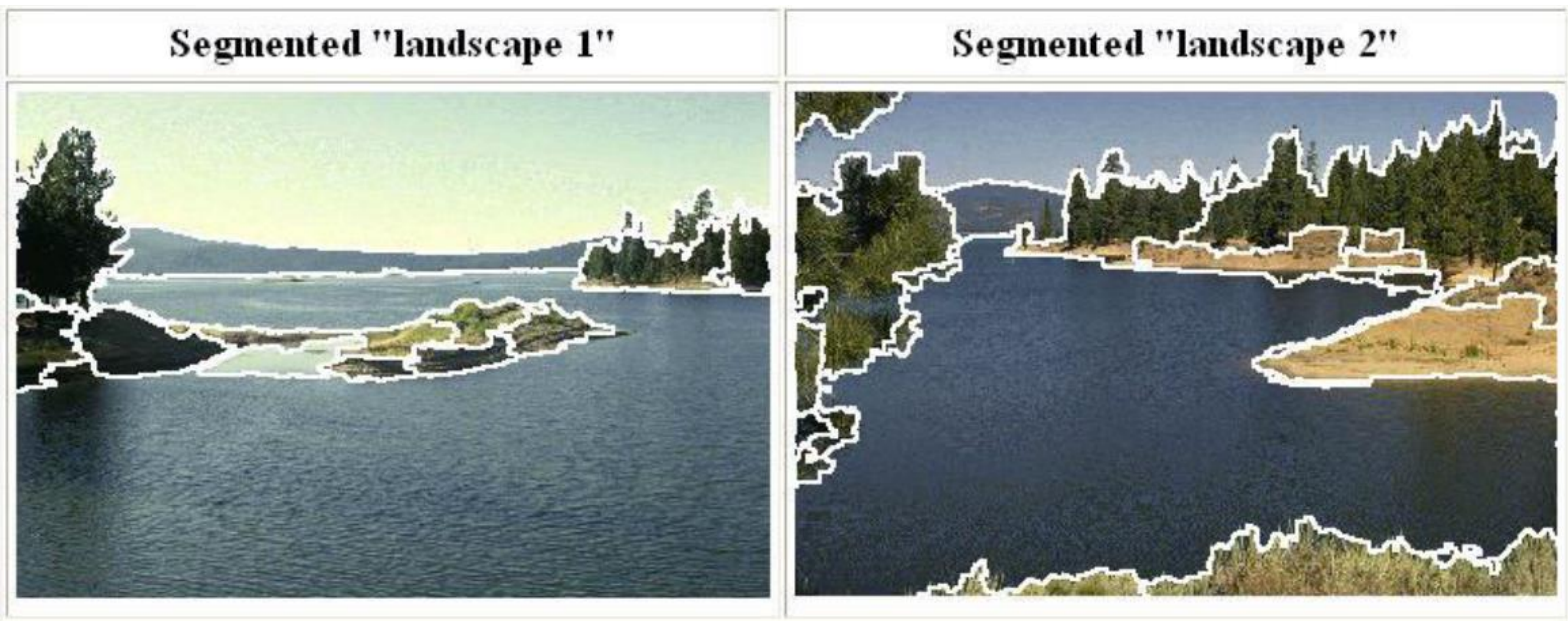


Image segmentation

Mean-shift color and spatial segmentation

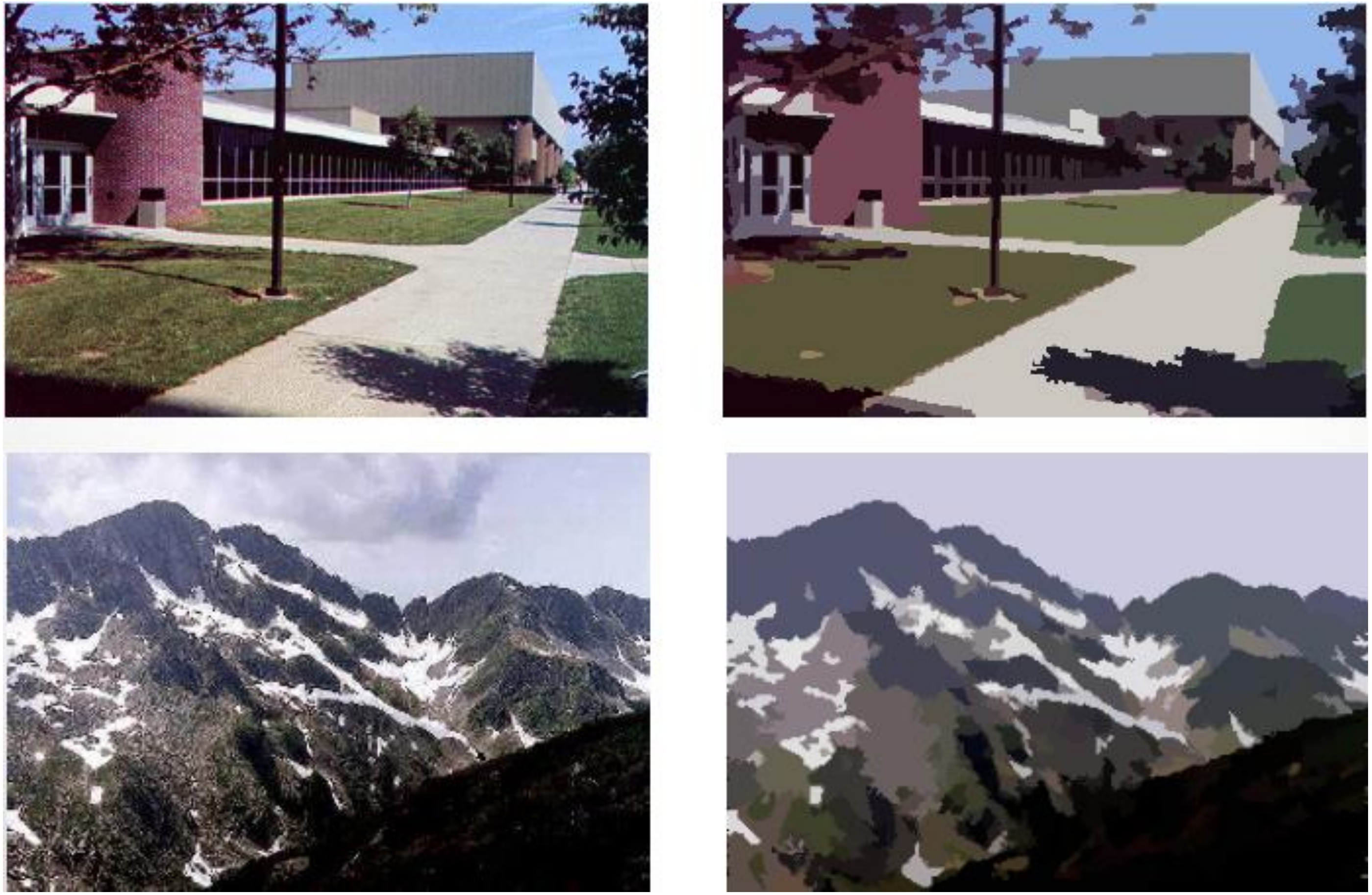


Image segmentation

Mean-shift color and spatial segmentation

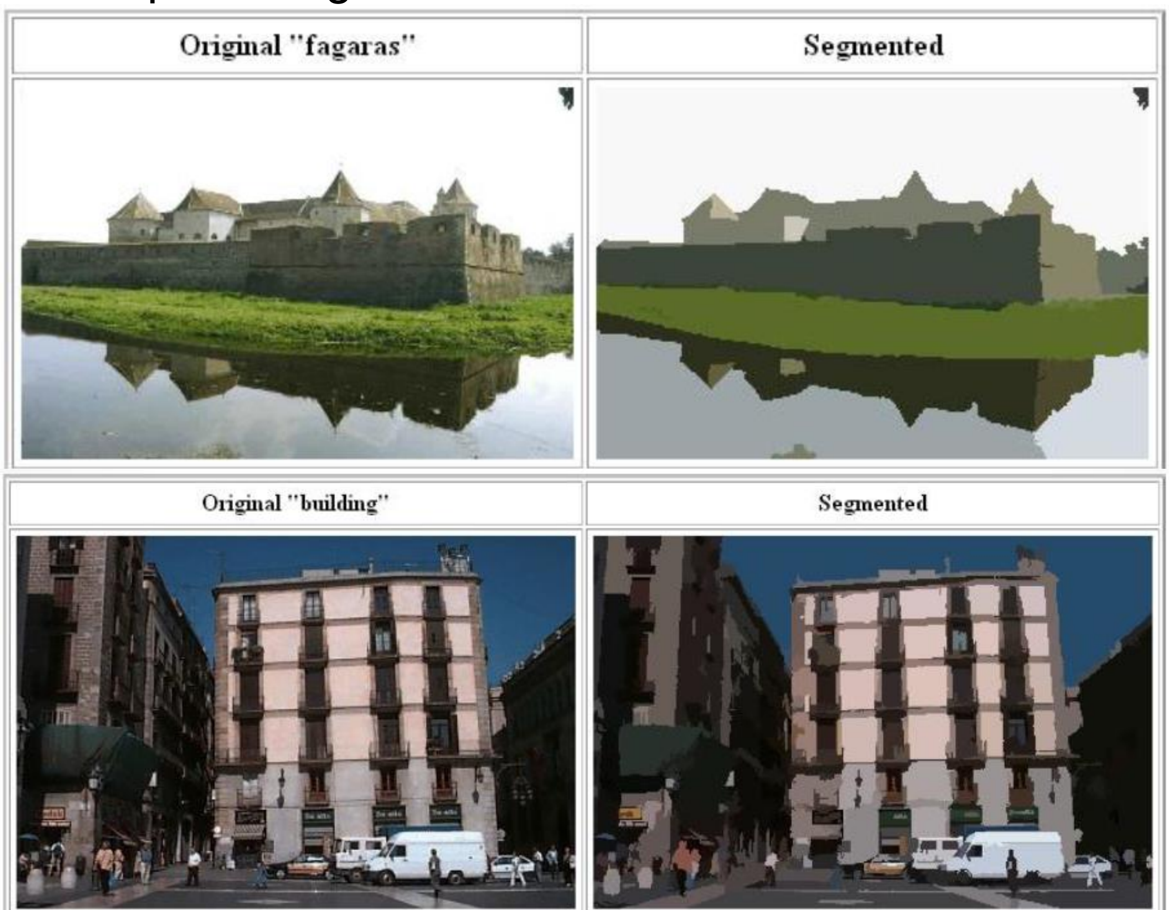


Image segmentation

Watershed segmentation:

- Typically over-segments the image
- Hence, a good way to find “superpixels”
- Algorithm:
  1. Compute map of the image gradient magnitude $||\nabla I||$
  2. Pick zeros of this map (i.e., extrema) as seed points.
  3. Assign pixels to seeds by a procedure that is analogous to filling a height map with water (hence the name). This is done by travelling backward down the gradient of $||\nabla I||$
  4. Merge seeds that end up at the same point.
- Note: there are many other ways to implement this

Image segmentation

Watershed segmentation:

- imshow(double(watershed(imread('cameraman.tif'))))

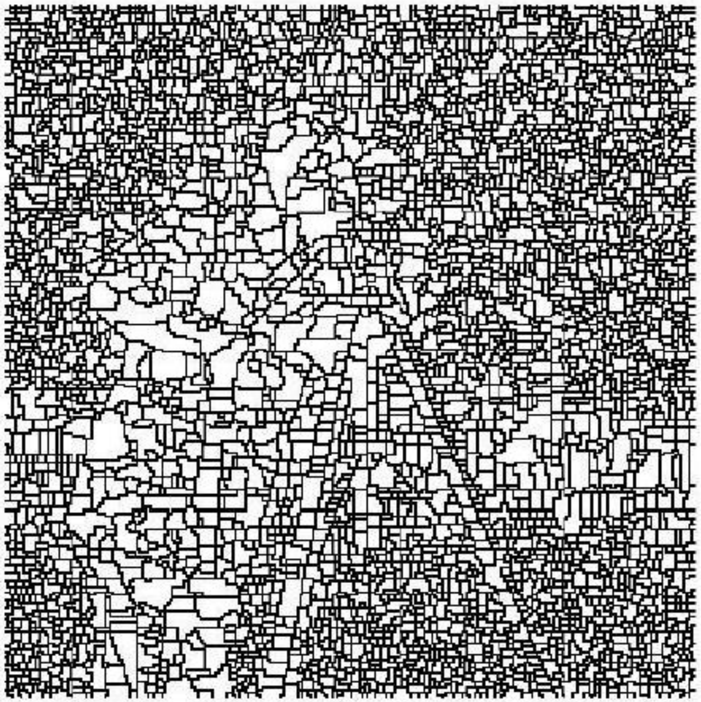


Image segmentation

Issues:

- How many regions are there?
- How to decide that two (super-)pixels belong to the same region?

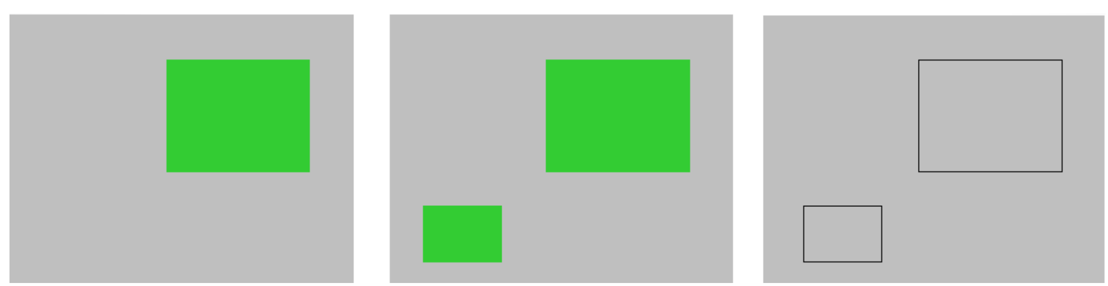


Image segmentation

Graph-based segmentation:

- Build a weighted graph G = (**V**,**E**) where
  - **V**: is the set of graph nodes → image (super-)pixels represented by vectors
  - **E**: is the set of connections of graph node pairs
  - **W**: is the set of graph weights
    - W_{ij} encodes the 'similarity' of nodes i and j
    - This is a form of probability that nodes i and j belong to the same region

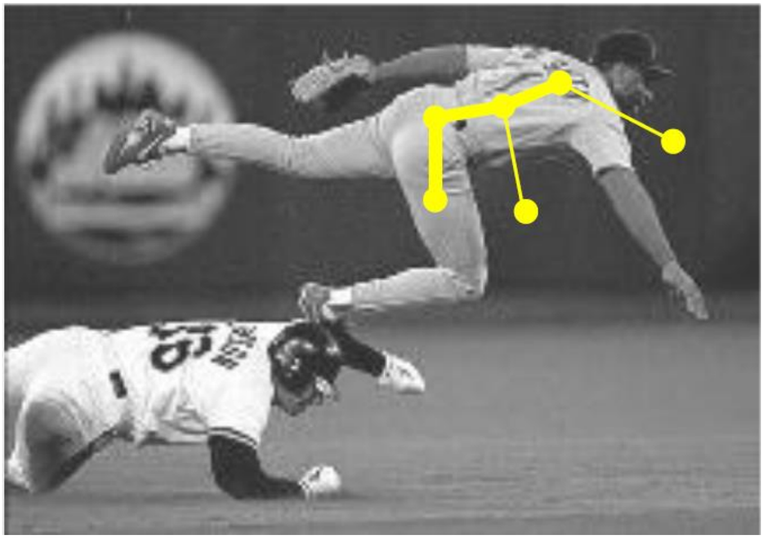
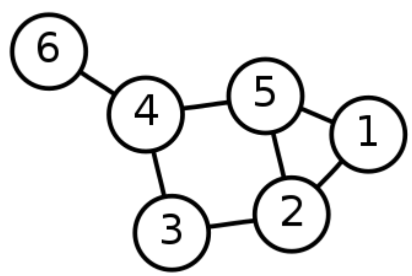


Image segmentation

Graph-based segmentation:

- Build a weighted graph G = (**V**,**E**) where
  - **V**: is the set of graph nodes → image (super-)pixels represented by vectors
  - **E**: is the set of connections of graph node pairs
  - **W**: is the set of graph weights
    - W_{ij} encodes the 'similarity' of nodes i and j
    - This is a form of probability that nodes i and j belong to the same region

- Image segmentation → Graph partition



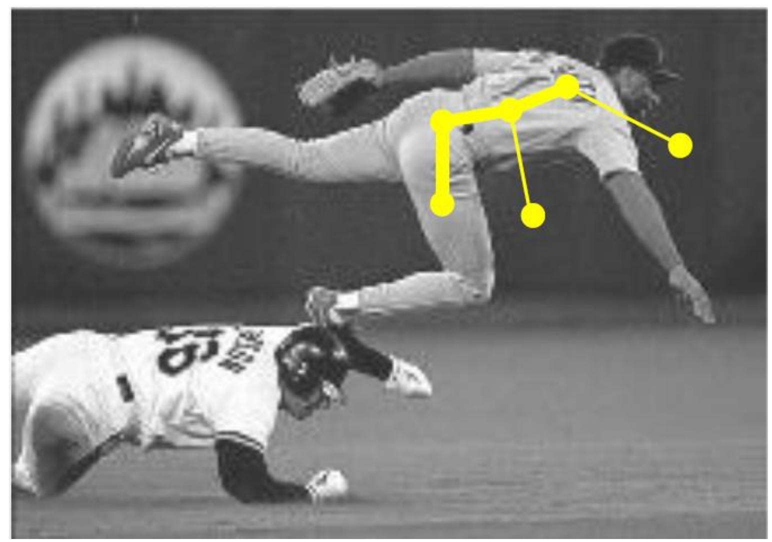


Image segmentation

Graph representations

- Adjacency matrix

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$

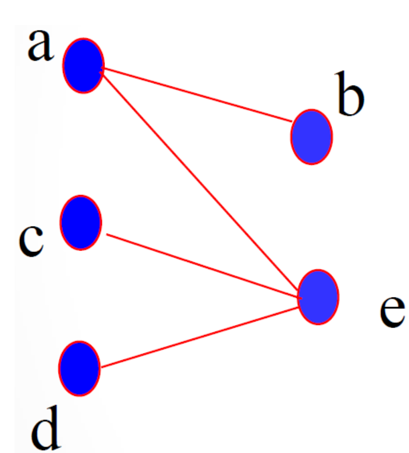


Image segmentation

Graph representations
- Adjacency matrix
- Affinity matrix

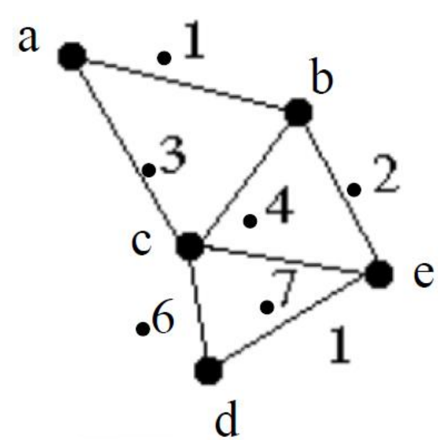
$$\begin{bmatrix} 1 & .1 & .3 & 0 & 0 \\ .1 & 1 & .4 & 0 & .2 \\ .3 & .4 & 1 & .6 & .7 \\ 0 & 0 & .6 & 1 & 1 \\ 0 & .2 & .7 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{c|c} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Several ways to calculate the affinity between z_i and z_j.

The heat kernel function (or radial basis function):

$$A_{ij} = \exp(- ||z_i – z_j||^2 / s^2)$$



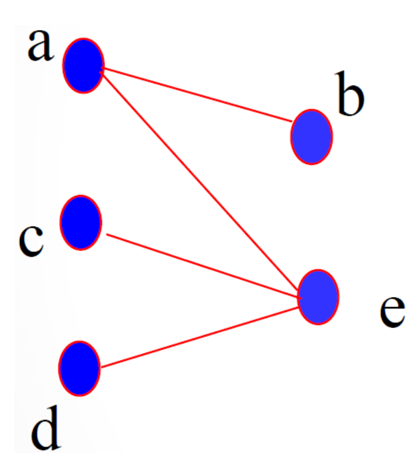
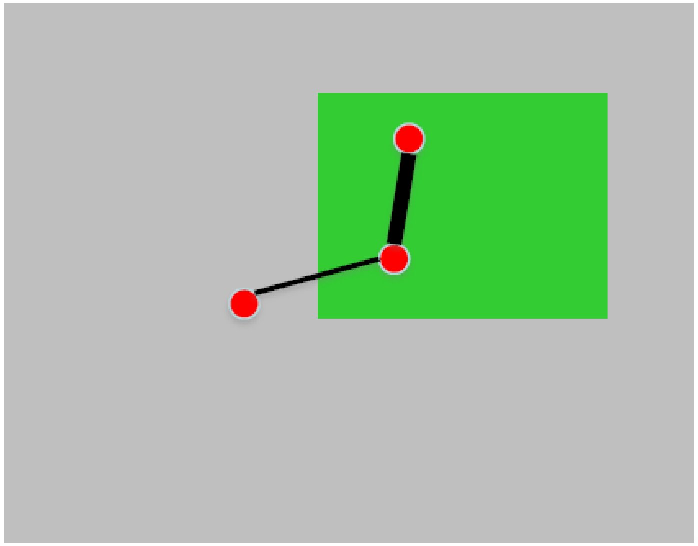


Image segmentation

Graph representations

The heat kernel function (or radial basis function):

$$A_{ij} = \exp(- ||z_i - z_j||^2 / s^2)$$



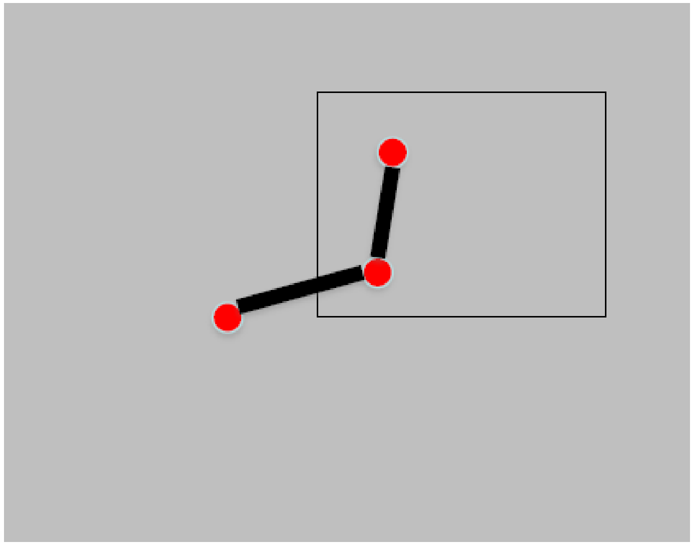


Image segmentation

Extracting a single good cluster:

- Maximize the within-cluster similarity (E)

$$E = w_n^T A w_n$$

- A is the affinity matrix and $\mathbf{w}_n$ is the vector of weights linking elements to the n'th cluster

{association of element i with cluster n}

x {affinity between i and j}

x {association of element i with cluster n}

- We usually require that $\mathbf{w}_n^T \mathbf{w}_n = 1$ (for scale invariance). Then the Lagrangian function is

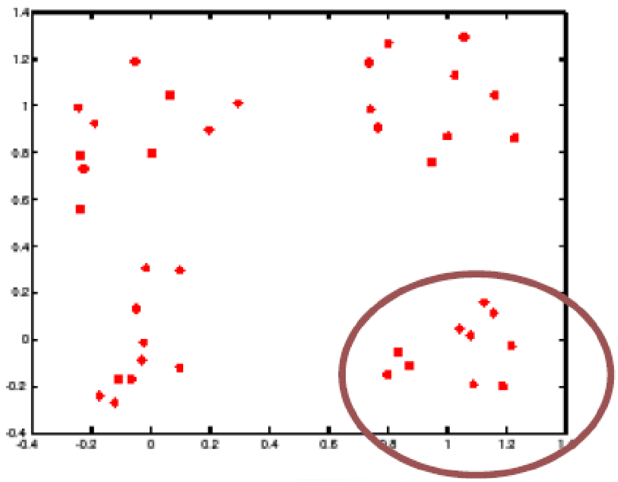
$$E = w_n^T A w_n + \lambda(w_n^T w_n - 1)$$

- Calculating the derivative and setting it to zero leads to eigenanalysis of **A**

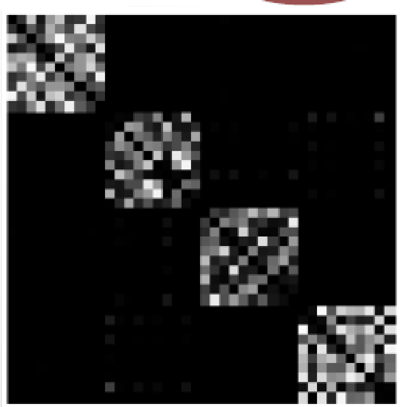
$$A w_n = \lambda w_n$$

Image segmentation

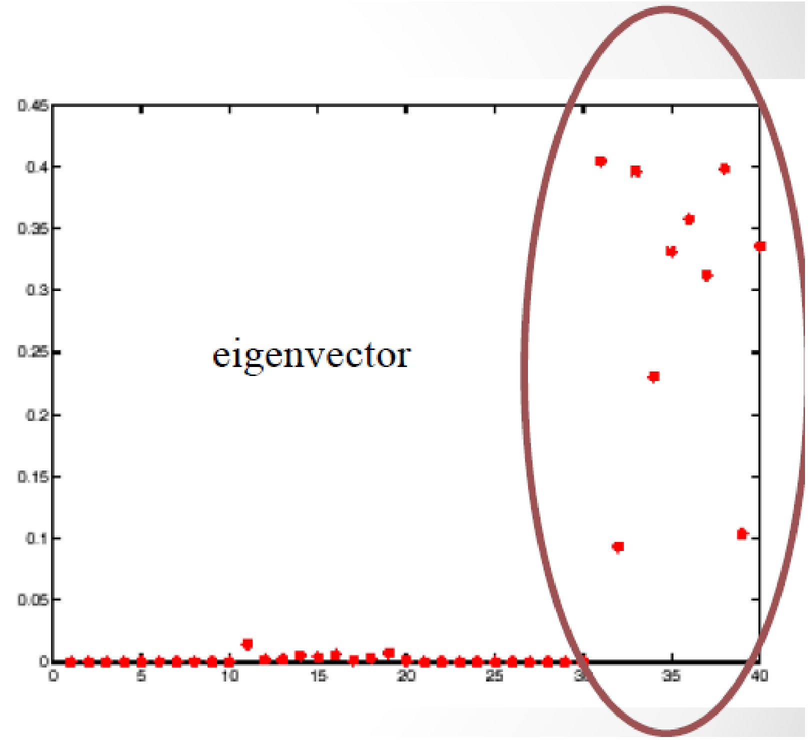
Example



points



Affinity
matrix



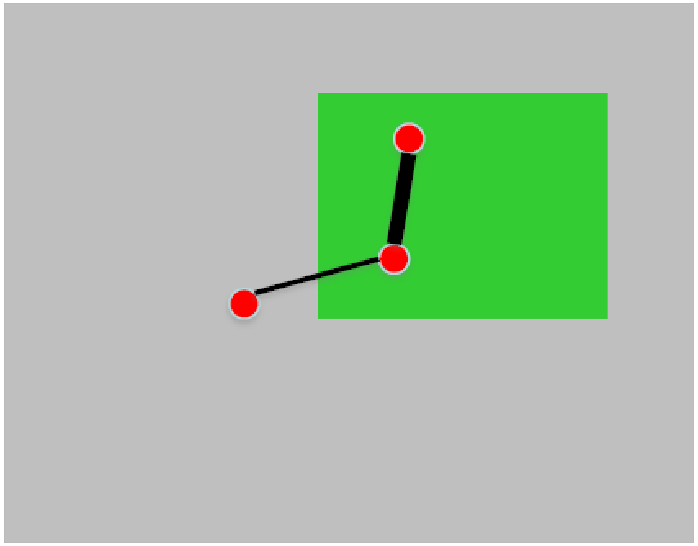
The eigenvector of the other three clusters show a similar structure (i.e., contains mainly zeros, except for elements belonging to the cluster)

Image segmentation

The above strategy will work for case 1

For case 2 it will not work, because pixels inside/outside the rectangle have the same color

Case 1



Case 2

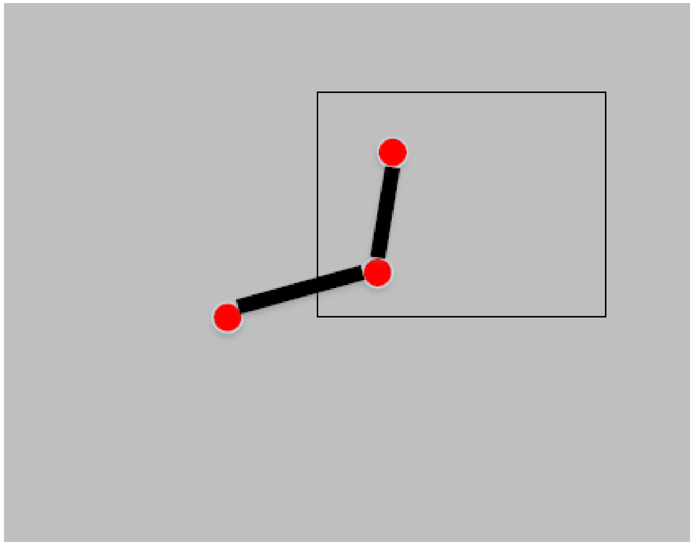
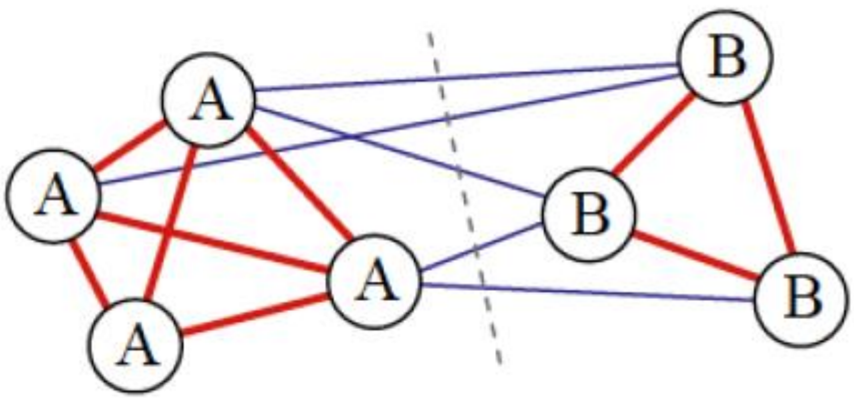


Image segmentation

Terminology for graph-based segmentation

	A	B	sum
A	$assoc(A,A)$	$cut(A,B)$	$assoc(A,V)$
B	$cut(B,A)$	$assoc(B,B)$	$assoc(B,V)$
sum	$assoc(A,V)$	$assoc(B,V)$	



Cut: sum of the weight of the cut edges:

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

with $\mathrm{A} \cap \mathrm{B} = \varnothing$ (disjoint)

Association: sum of the weights of the edges connecting two sets:

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

A and B not necessarily disjoint

Image segmentation

Minimum Cut:

- A cut of a graph G is the set of edges S such that removal of S from G disconnects G
- Minimum cut is the cut of minimum weight, where weight of cut(A,B) is given as

Cut: sum of the weight of the cut edges:

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

with $A \cap B = \emptyset$ (disjoint)

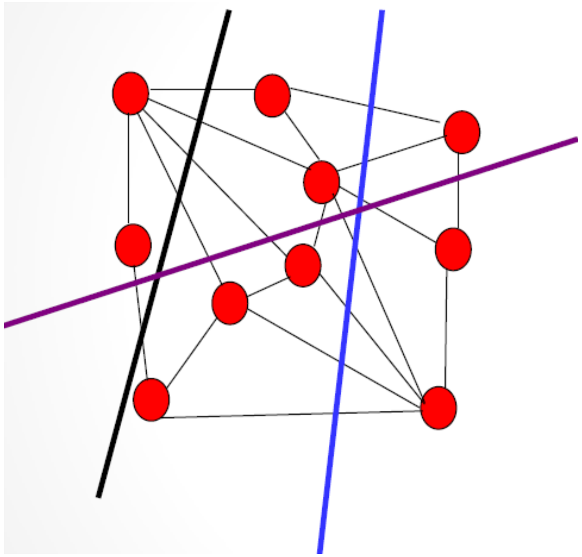


Image segmentation

Minimum Cut:

- A cut of a graph G is the set of edges S such that removal of S from G disconnects G

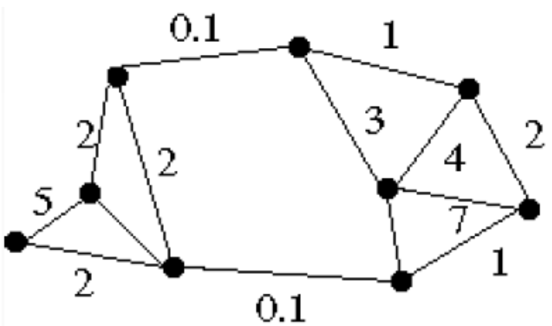
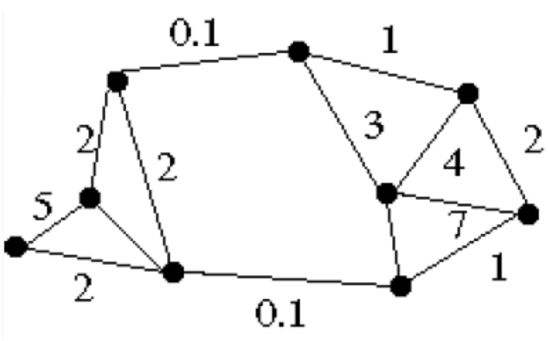


Image segmentation

Minimum Cut:

- A cut of a graph G is the set of edges S such that removal of S from G disconnects G



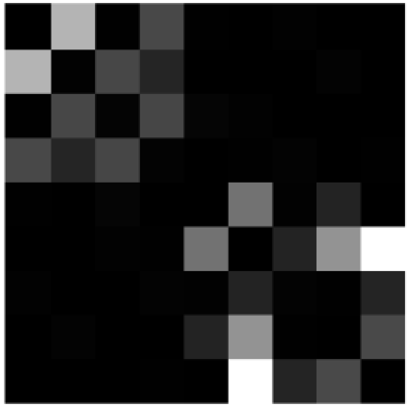
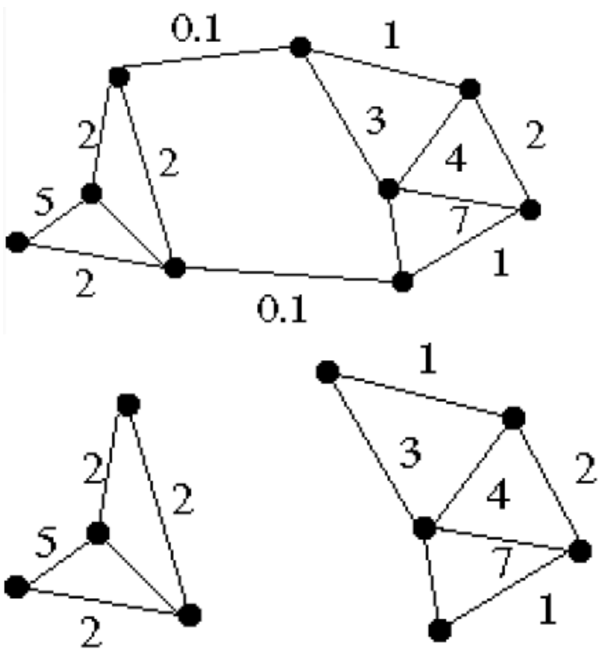


Image segmentation

Minimum Cut:

- A cut of a graph G is the set of edges S such that removal of S from G disconnects G



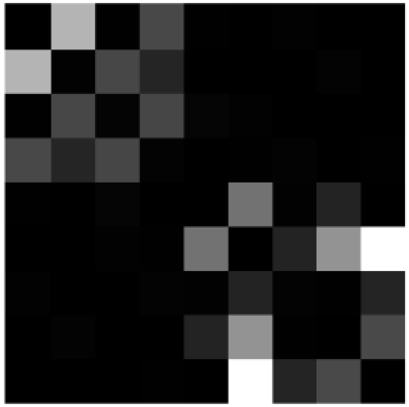
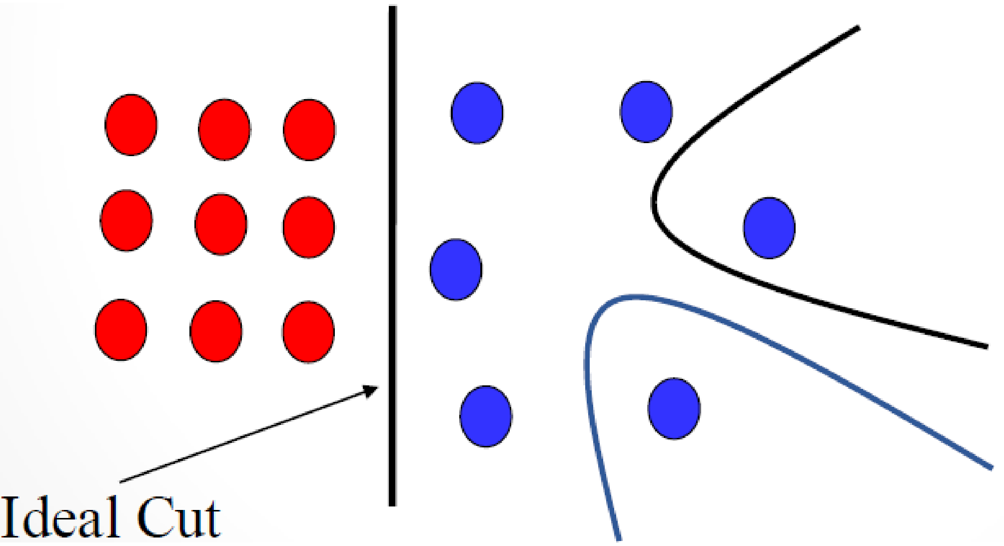


Image segmentation

Minimum Cut:

- Drawback: weight of cut is proportional to the number of edges in the cut



Cuts with lesser weight than the ideal cut

Image segmentation

Normalized Cut:

- First eigenvector of affinity matrix captures within cluster similarity, but not across cluster difference
- Min-cut can find degenerate clusters
- 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

Image segmentation

Normalized Cut:

- First eigenvector of affinity matrix captures within cluster similarity, but not across cluster difference
- Min-cut can find degenerate clusters
- 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
- Construct A and B (sub-graphs) such that their within-cluster similarity is high compared to their association with the rest of the graph.
- This can be expressed as the minimization of

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

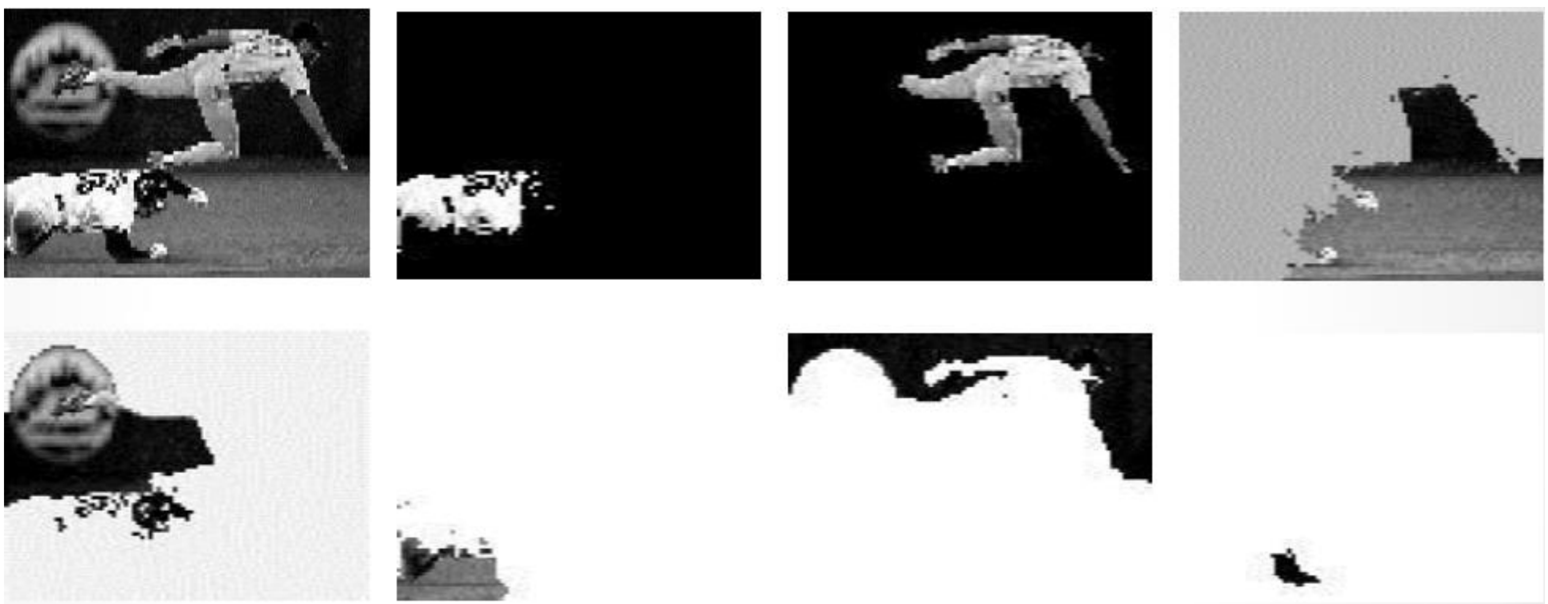
Image segmentation

Normalized Cut:

- Exact discrete solution to NCut is NP-complete even on regular grid
  - [Papadimitriou'97]
- Drawing on spectral graph theory, good approximation can be obtained by solving a generalized eigenvalue problem.

Image segmentation

Normalized Cut examples:
 - brightness image segmentation

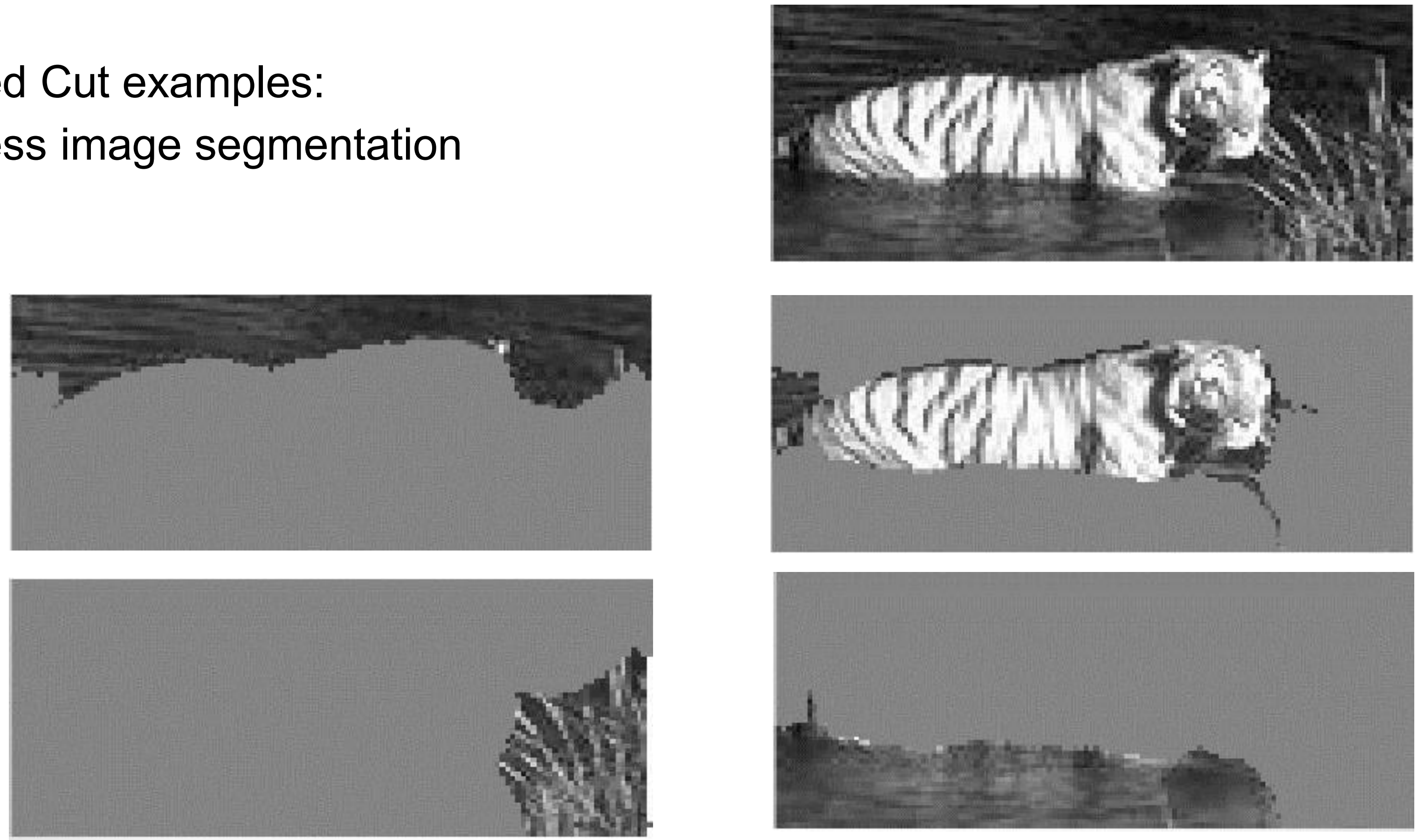


https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf

Image segmentation

Normalized Cut examples:

- brightness image segmentation

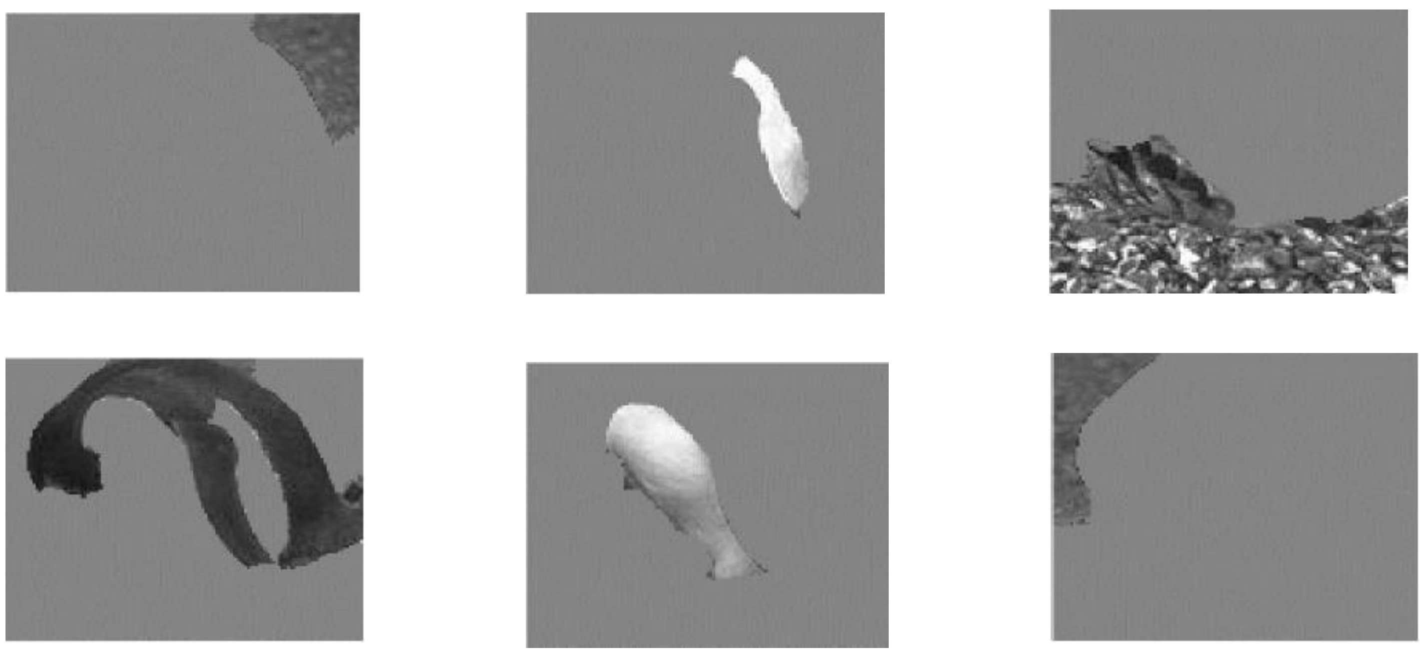


https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf

Image segmentation

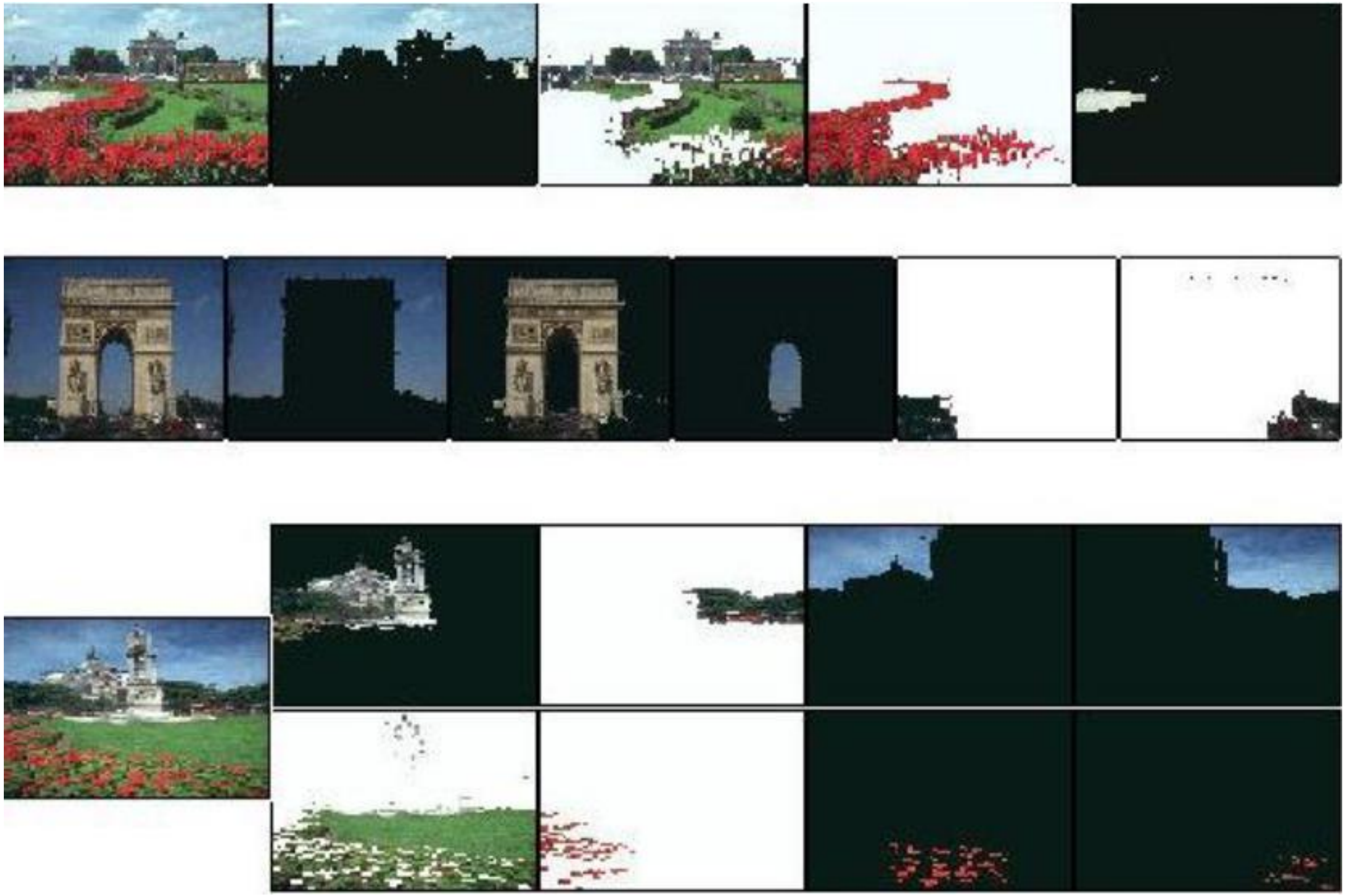
Normalized Cut examples:

- brightness image segmentation



https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf

Image segmentation



https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf

Image segmentation

Solving case 2:

- In subsequent work, Malik, Belongie, Leung et al. (2001) look for intervening contours between pixels i and j and define an intervening contour weight

$$w_{ij}^{IC} = 1 - \max_{x \in l_{ij}} p_{con}(x)$$

where l_{ij} is the image line joining pixels i and j and $p_{con}(x)$ is the probability of an intervening contour perpendicular to this line, which is defined as the negative exponential of the oriented energy in the perpendicular direction

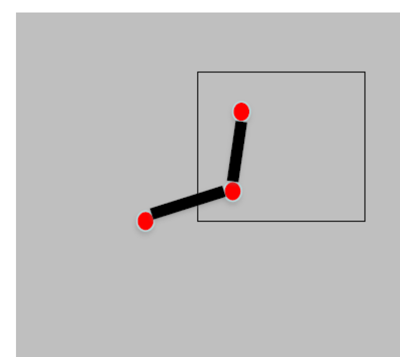


Image segmentation

Solving case 2:

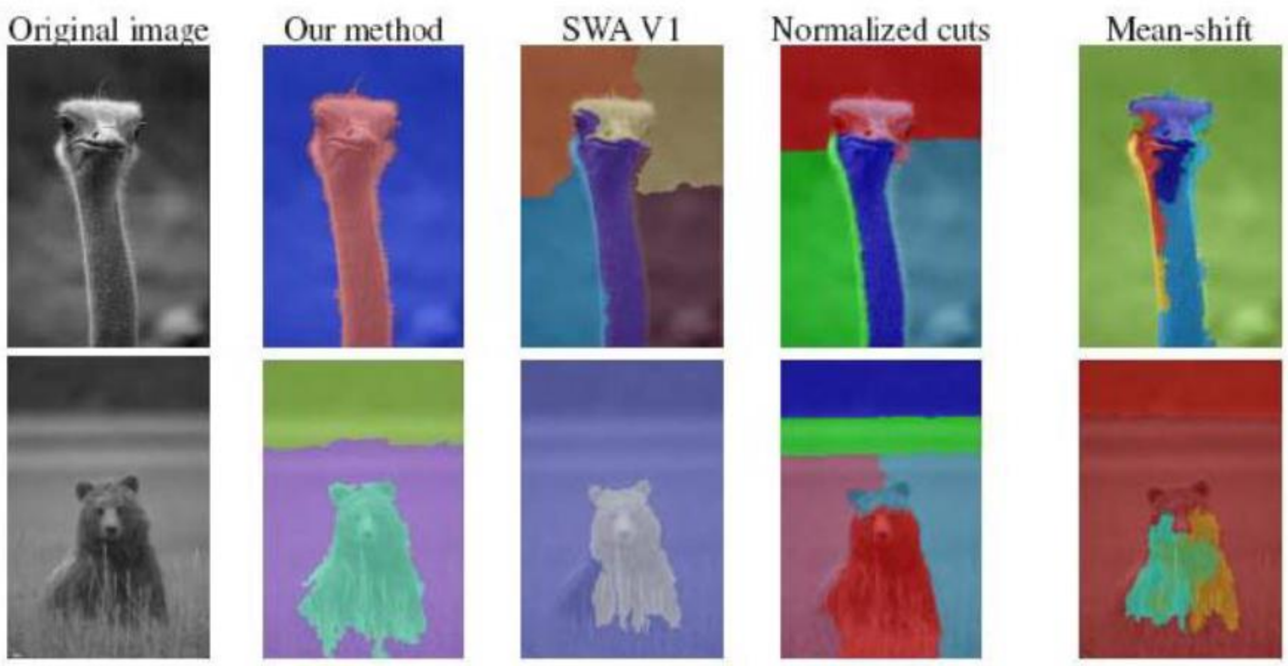
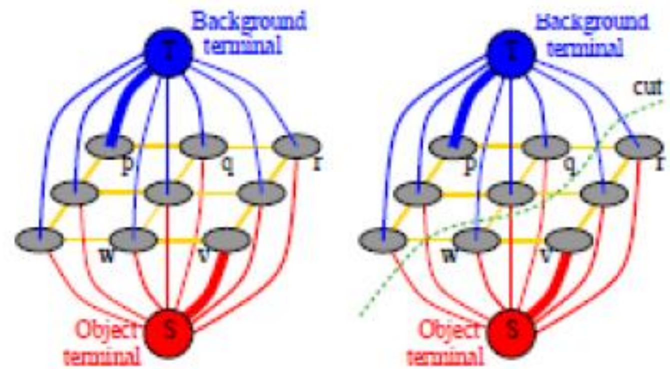


Figure 5.22 Comparative segmentation results (Alpert, Galun, Basri *et al.* 2007) © 2007 IEEE. “Our method” refers to the probabilistic bottom-up merging algorithm developed by Alpert *et al.*

Image segmentation

Foreground/background separation (GrabCut)

Szeliski Ch. 5.5



(a) (b)

Figure 5.23 Graph cuts for region segmentation (Boykov and Jolly 2001) © 2001 IEEE: (a) the energy function is encoded as a maximum flow problem; (b) the minimum cut determines the region boundary.

(a) (b) (c)

Figure 5.24 GrabCut image segmentation (Rother, Kolmogorov, and Blake 2004) © 2004 ACM: (a) the user draws a bounding box in red; (b) the algorithm guesses color distributions for the object and background and performs a binary segmentation; (c) the process is repeated with better region statistics.

Image segmentation

Principles of grouping:

- After segmentation, the image is divided into regions, which we may think of as “super pixels”
- Super pixels often need to be joined in order make meaningful segmentations.
- There are hundreds of different grouping laws!

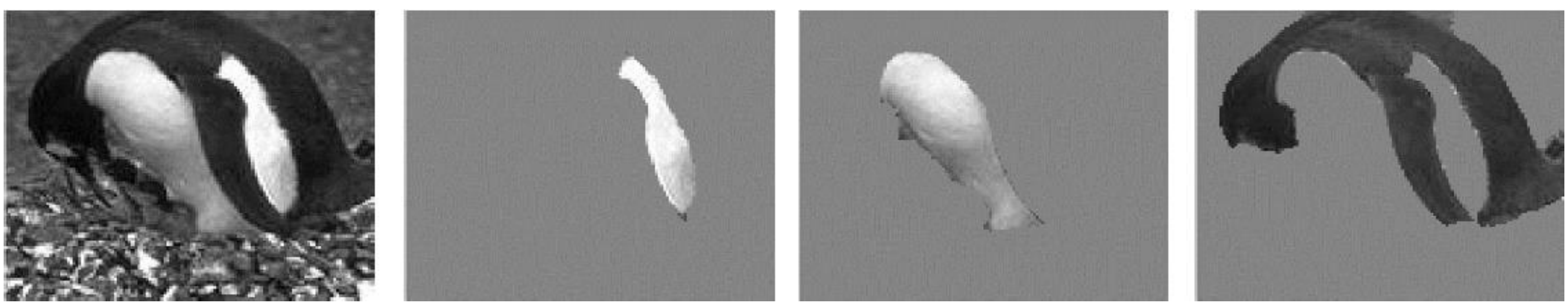


Image segmentation

Principles of grouping:
- humans naturally perceive objects as organized patterns and objects

https://en.wikipedia.org/wiki/Principles_of_grouping

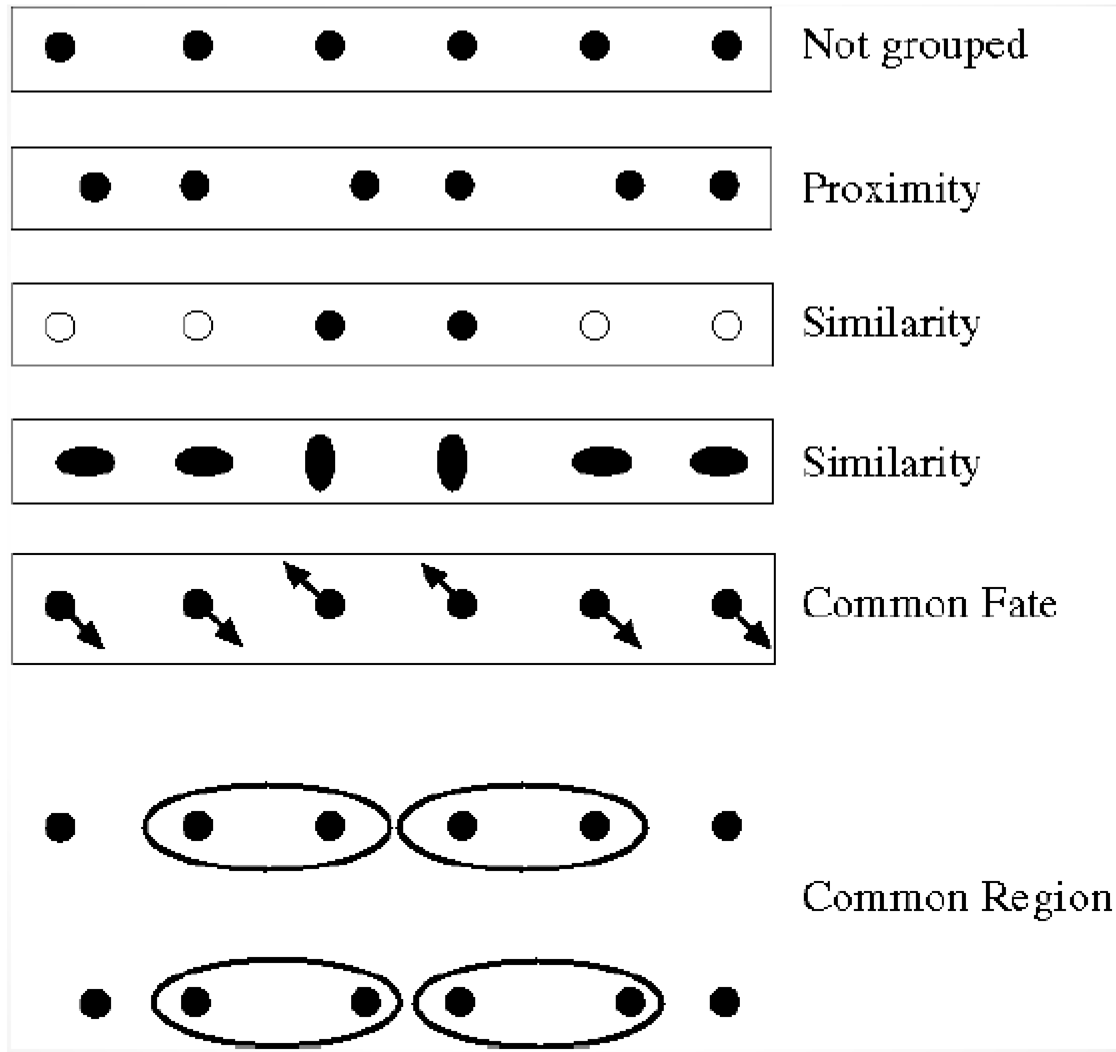
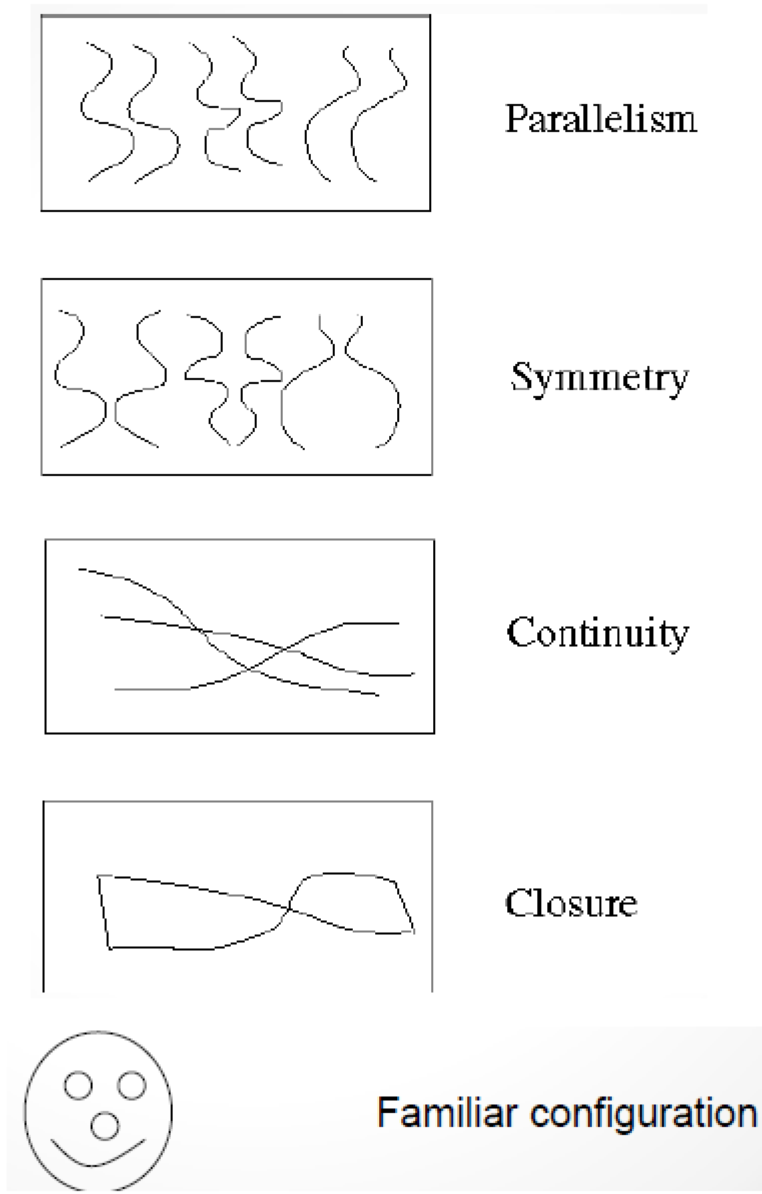
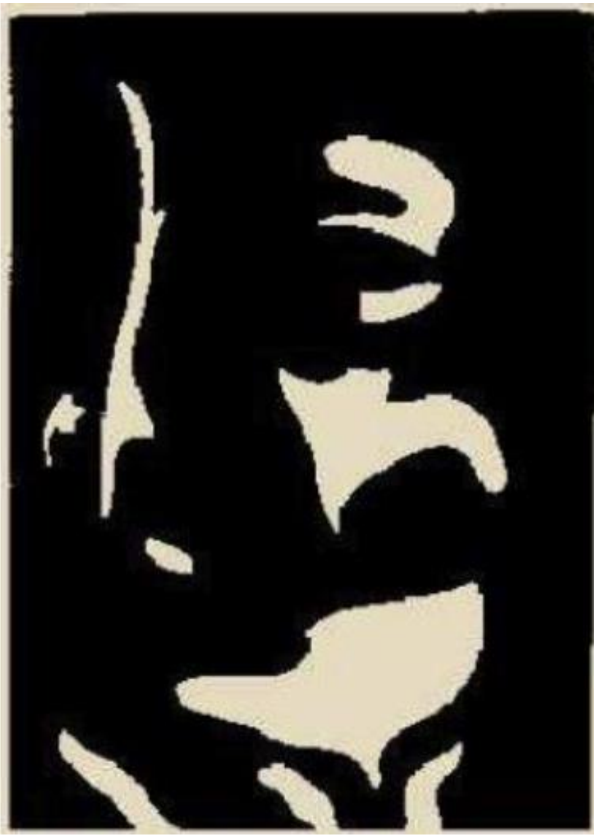


Image segmentation



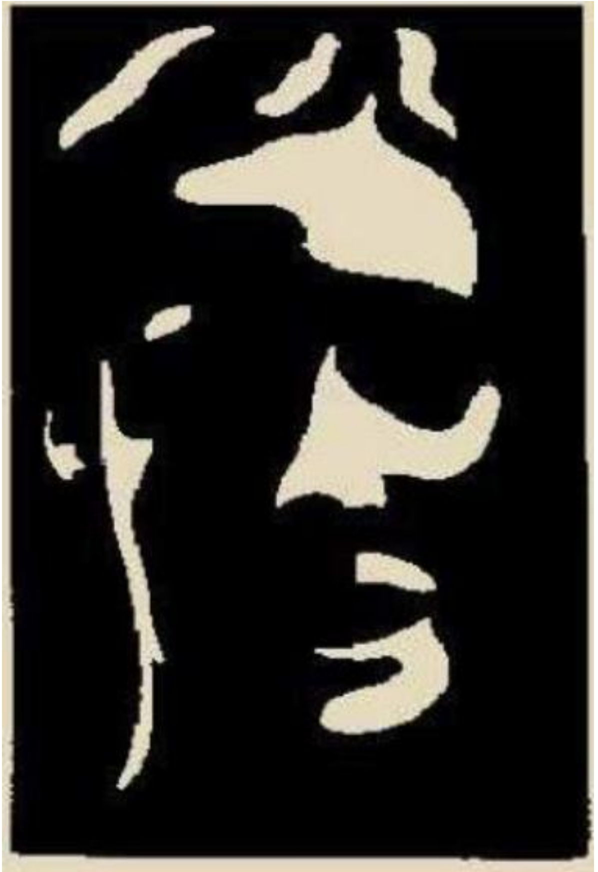
Familiarity

What is depicted in this image?



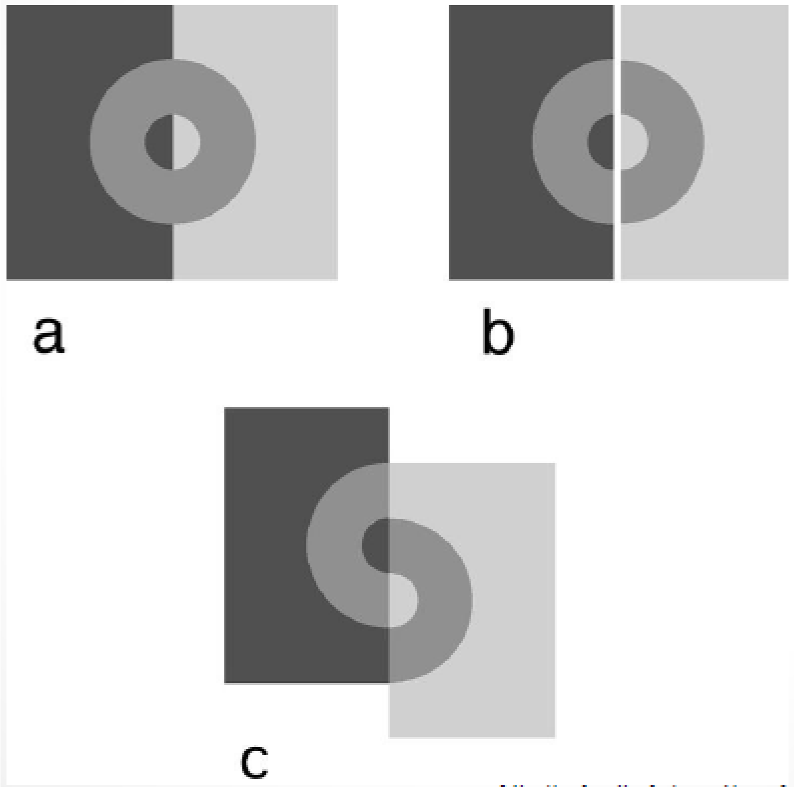
Familiarity

What is depicted in this image?



Influences of grouping

Grouping influences other perceptual mechanisms such as lightness perception

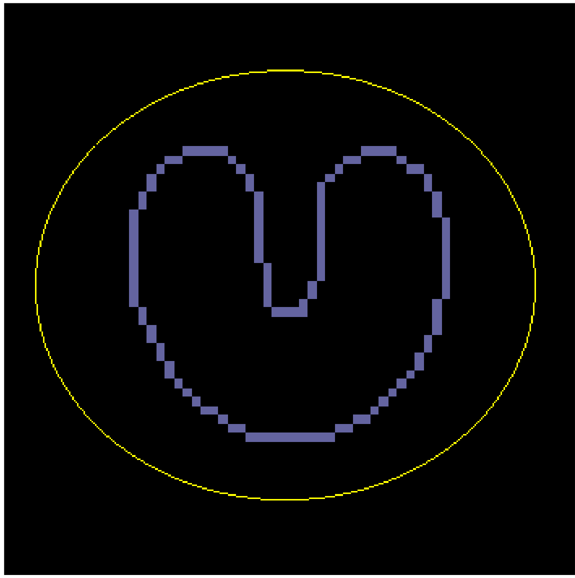


Boundary-based segmentation

Active contour models (or snakes):

- The shape of many objects is not easily represented by rigid primitives
  - Natural objects, e.g. bananas, cars, people, have similar recognizable shapes, which are also slightly different (for objects of the same category)
  - Some objects, such as lips, change over time.

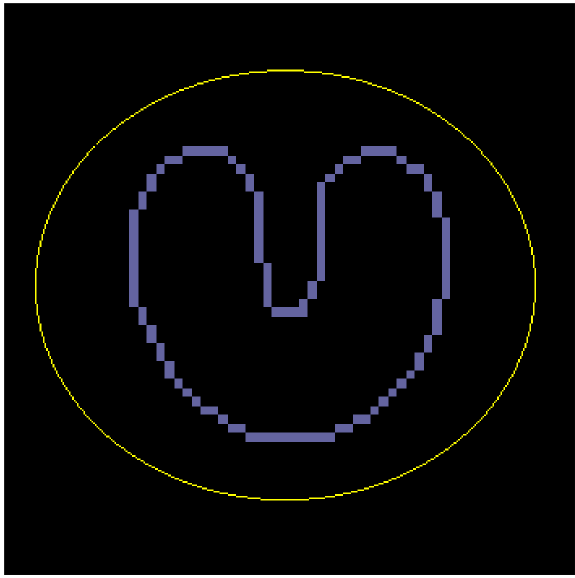
Demo matlab code



Boundary-based segmentation

Active contour models (or snakes):
- The shape of many objects is not easily represented by rigid primitives
  - Natural objects, e.g. bananas, cars, people, have similar recognizable shapes, which are also slightly different (for objects of the same category)
  - Some objects, such as lips, change over time

Process:
- A higher level process or a user initializes any curve close to the object boundary.
- The snake then starts deforming and moving towards the desired object boundary.
- In the end it completely “shrink-wraps” around the object



Boundary-based segmentation

Active contour models (or snakes):

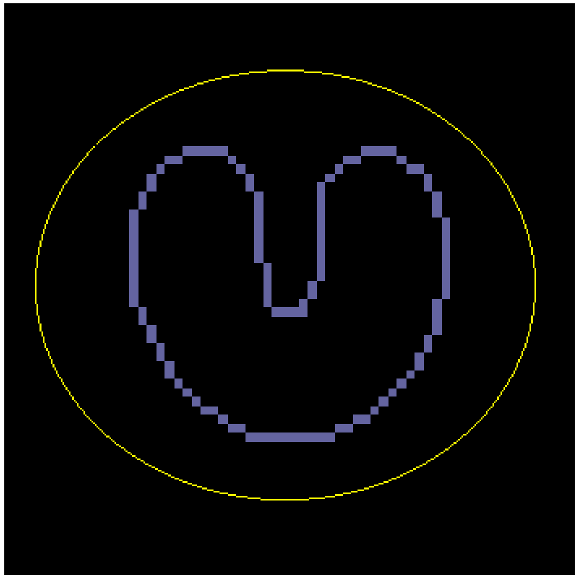
- The shape of many objects is not easily represented by rigid primitives
- The contour is defined in the (x,y) plane of an image as a parametric curve of the form:

$$v(s) = (x(s), y(s))$$

- The contour "possesses" an energy (E_{snake}) which is defined as the sum of the three energy terms

$$E_{snake} = E_{internal} + E_{external} + E_{constraint}$$

- The energy terms are defined such that minimizing E_{snake} means that we reach the final position of the contour



Boundary-based segmentation

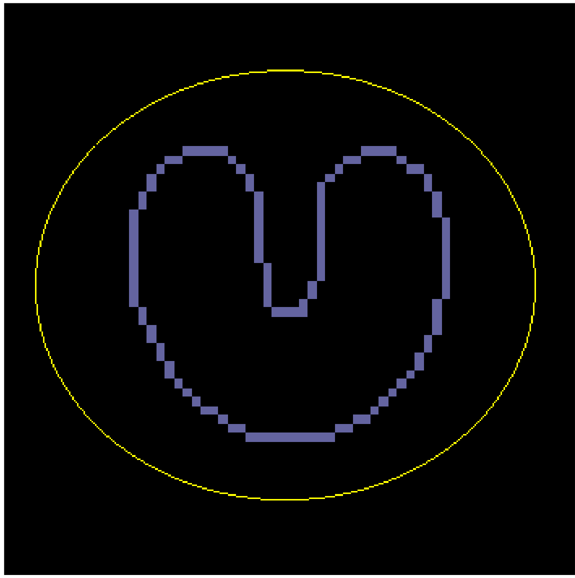
Active contour models (or snakes):

- The contour "possesses" an energy (E_{snake}) which is defined as the sum of the three energy terms

$$E_{snake} = E_{internal} + E_{external} + E_{constraint}$$

- The energy $E_{internal}$
  - Depends on the intrinsic properties of the curve
  - It is the sum of elastic energy and bending energy

$$E_{int} = E_{elastic} + E_{bending}$$



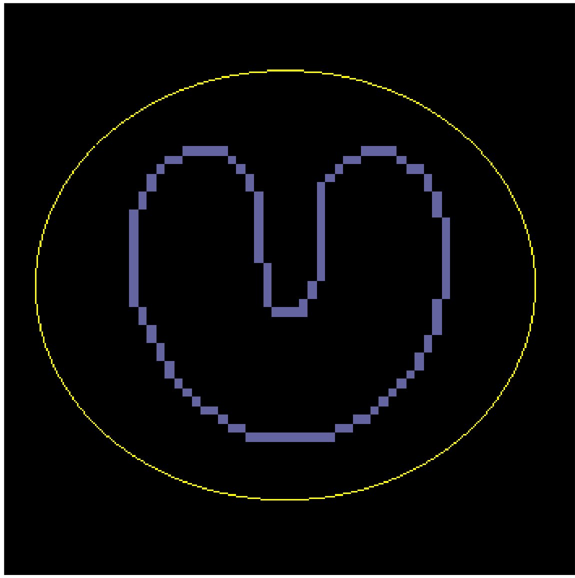
Boundary-based segmentation

Active contour models (or snakes):

- The energy $E_{internal}$
  - Depends on the intrinsic properties of the curve
  - It is the sum of elastic energy and bending energy
    - $E_{elastic}$
      - The curve is treated as an elastic rubber band possessing elastic potential energy.
      - It discourages stretching by introducing tension

$$E_{elastic} = \frac{1}{2}\int_{s} \alpha(s)\,|\,v_s\,|^2\, ds \qquad v_s = \frac{dv(s)}{ds}$$

      - Weight α(s) allows us to control elastic energy along different parts of the contour (usually constant)
      - <u>Responsible for shrinking of the contour</u>



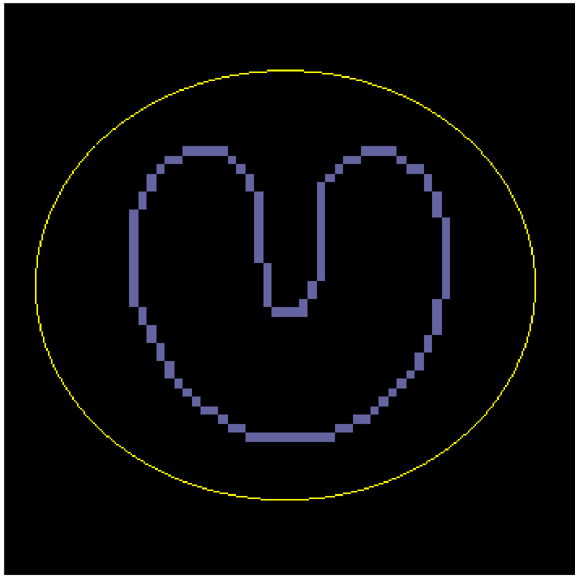
Boundary-based segmentation

Active contour models (or snakes):

- The energy $E_{internal}$
  - Depends on the intrinsic properties of the curve
  - It is the sum of elastic energy and bending energy
    - $E_{bending}$
      - The snake is also considered to behave like a thin metal strip giving rise to bending energy.
      - It is defined as sum of squared curvature of the contour

$$E_{bending} = \frac{1}{2} \int_s \beta(s) \, | v_{ss} |^2 \, ds$$

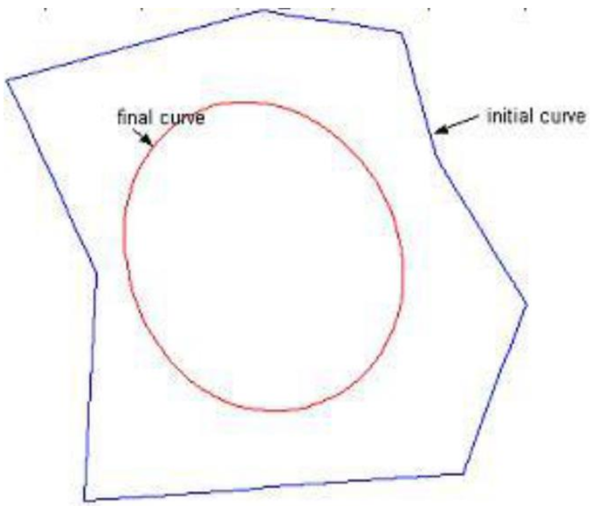
      - β(s) plays a similar role to α(s).
      - <u>Responsible for smoothing of the contour</u>



Boundary-based segmentation

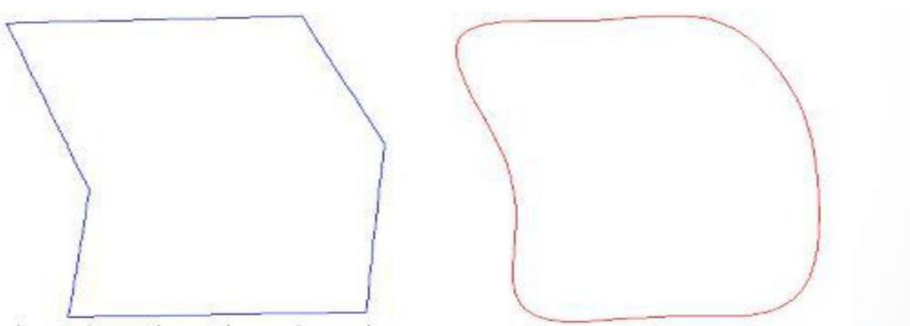
Elastic force

- Generated by elastic potential energy of the curve



Bending force:

- Generated by the bending energy of the contour.



Initial curve (High bending energy)

Final curve deformed by bending force. (low bending energy)

Boundary-based segmentation

Active contour models (or snakes):

- The contour "possesses" an energy (E_{snake}) which is defined as the sum of the three energy terms

$$E_{snake} = E_{internal} + E_{external} + E_{constraint}$$

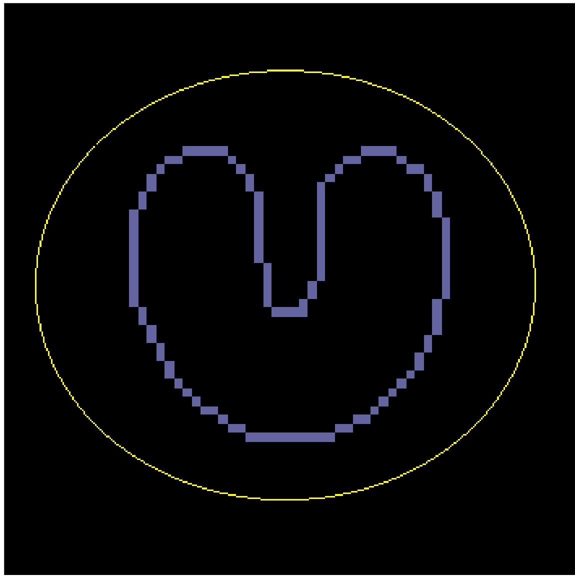
- The energy $E_{external}$
  - Derived from the image.
  - Define a function $E_{image}(x,y)$ so that it takes on its smaller values at the features of interest, such as boundaries

$$E_{ext} = \int_s E_{image}(v(s))ds$$

  - E_{image} needs to be defined, e.g.

$$E_{image}(x,y) = -|\nabla I(x,y)|^2$$

$$E_{image}(x,y) = -|\nabla (G_\sigma(x,y) * I(x,y))|^2$$



Boundary-based segmentation

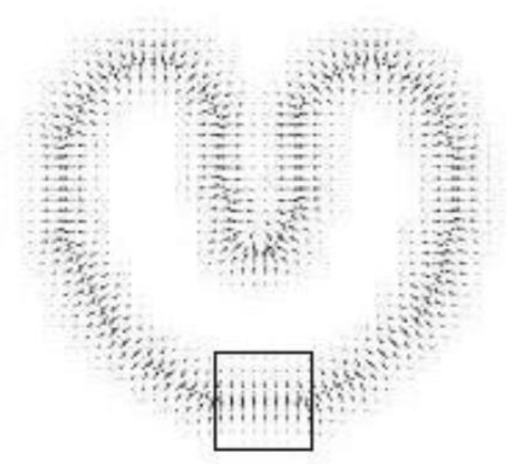
External force

- It acts in the direction needed to minimize E_{ext}

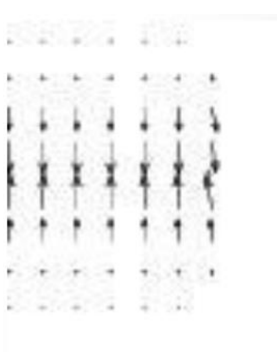
$$F_{ext} = -\nabla E_{image}$$

Image

External force



Zoom in



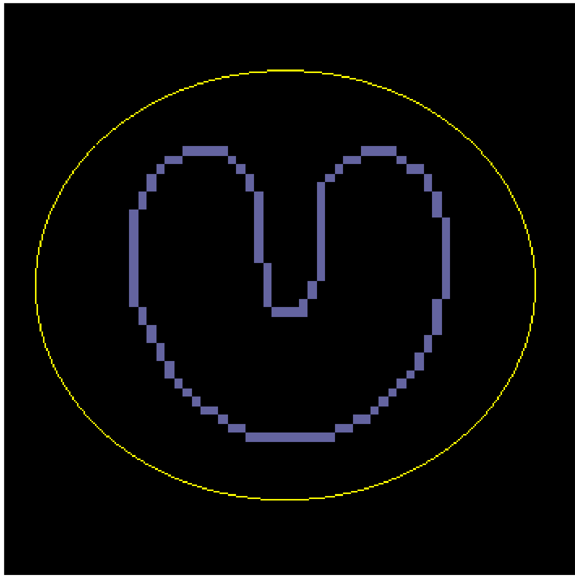
Boundary-based segmentation

Active contour models (or snakes):

- The contour "possesses" an energy (E_{snake}) which is defined as the sum of the three energy terms

$$E_{snake} = E_{internal} + E_{external} + E_{constraint}$$

- The energy $E_{constraint}$
  - is determined by external constraints.
  - may come in the form of a spring attached by the user.
    Or, may come from higher knowledge about the images
    - for instance "shape prior"



Boundary-based segmentation

Active contour models (or snakes):

- The problem to be solved is to find a contour v(s) that minimizes the energy function

$$E_{snake} = \int_s \frac{1}{2}(\alpha(s)\,|\,v_s\,|^2 + \beta(s)\,|\,v_{ss}\,|^2) + E_{image}(v(s))ds$$

- Using variational calculus and by applying Euler-Lagrange differential equation we get following equation

$$\alpha v_{ss} - \beta v_{ssss} - \nabla E_{image} = 0$$

Boundary-based segmentation

Active contour models (or snakes):

- The problem to be solved is to find a contour v(s) that minimizes the energy function

$$E_{snake} = \int_s \frac{1}{2}(\alpha(s)\,|\,v_s\,|^2 + \beta(s)\,|\,v_{ss}\,|^2) + E_{image}(v(s))ds$$

- Using variational calculus and by applying Euler-Lagrange differential equation we get following equation

s denotes derivative!

$$\alpha v_{ss} - \beta v_{ssss} - \nabla E_{image} = 0$$

Boundary-based segmentation

Active contour models (or snakes):

- The problem to be solved is to find a contour v(s) that minimizes the energy function

$$E_{snake} = \int_s \frac{1}{2}(\alpha(s)\,|\,v_s\,|^2 + \beta(s)\,|\,v_{ss}\,|^2) + E_{image}(v(s))ds$$

- Using variational calculus and by applying Euler-Lagrange differential equation we get following equation

$$\underbrace{\boxed{\alpha v_{ss} - \beta v_{ssss}}}_{F_{int}} \underbrace{\boxed{-\nabla E_{image}}}_{F_{ext}} = 0$$

- Equation can be interpreted as a force balance equation
- Each term corresponds to a force produced by the respective energy terms. The contour deforms under the action of these forces

Boundary-based segmentation

Active contour models (or snakes):

- Optimizing E(x)

Linear approximation E(x):

$$E(\mathbf{x} + \delta\mathbf{x}) \approx E(\mathbf{x}) + \frac{\partial E}{\partial \mathbf{x}} \cdot \delta\mathbf{x}$$

Direction of the steepest descent:

$$\delta\mathbf{x} \propto -\frac{\partial E}{\partial \mathbf{x}}$$

$$E(\mathbf{x} + \delta\mathbf{x}) \approx E(\mathbf{x}) - \tau \left(\frac{\partial E}{\partial \mathbf{x}}\right)^2$$

Boundary-based segmentation

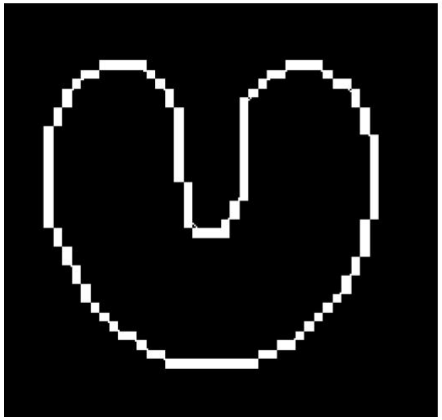
Active contour models (or snakes):

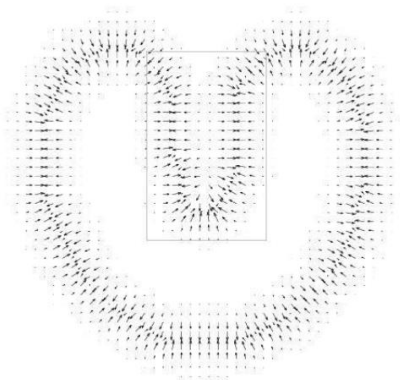
- Discretizing the solution:
  - The contour v(s) is represented by a set of control points v_0, v_1, …, v_{n-1}
  - The curve is piecewise linear obtained by joining each control point.
  - Force equations applied to each control point separately.
  - Each control point allowed to move freely under the influence of the forces.
  - The energy and force terms are converted to discrete form with the derivatives substituted by finite differences.

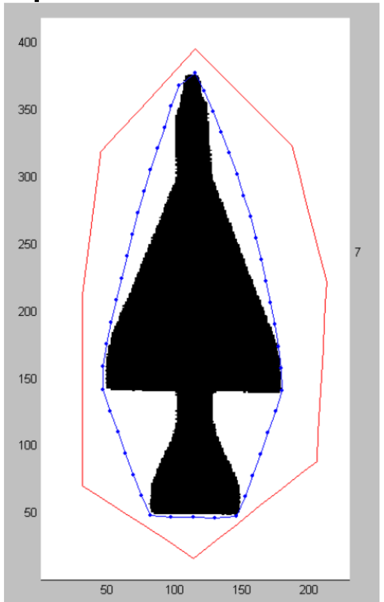
Boundary-based segmentation

Weakness of traditional snakes:

- Sensitive to parameters
- Small capture range
- No external force acts on points which are far away from the boundary
- Convergence is dependent on initial position
- Fails to detect concave boundaries. External force can't pull control points into boundary concavity







Boundary-based segmentation

Gradient Vector Flow (GVF):

- A “new” external force for snakes
- Detects shapes with boundary concavities.
- Large capture range.

http://www.iacl.ece.jhu.edu/static/gvf/

Boundary-based segmentation

Gradient Vector Flow (GVF):

- A “new” external force for snakes
- Detects shapes with boundary concavities.
- Large capture range.
- The GVF field is defined to be a vector field V(x,y) = (u(x,y), v(x,y))

- Force equation of GVF snake

$$\alpha v_{ss} - \beta v_{ssss} + V = 0$$

- V(x,y) is defined such that it minimizes the energy functional

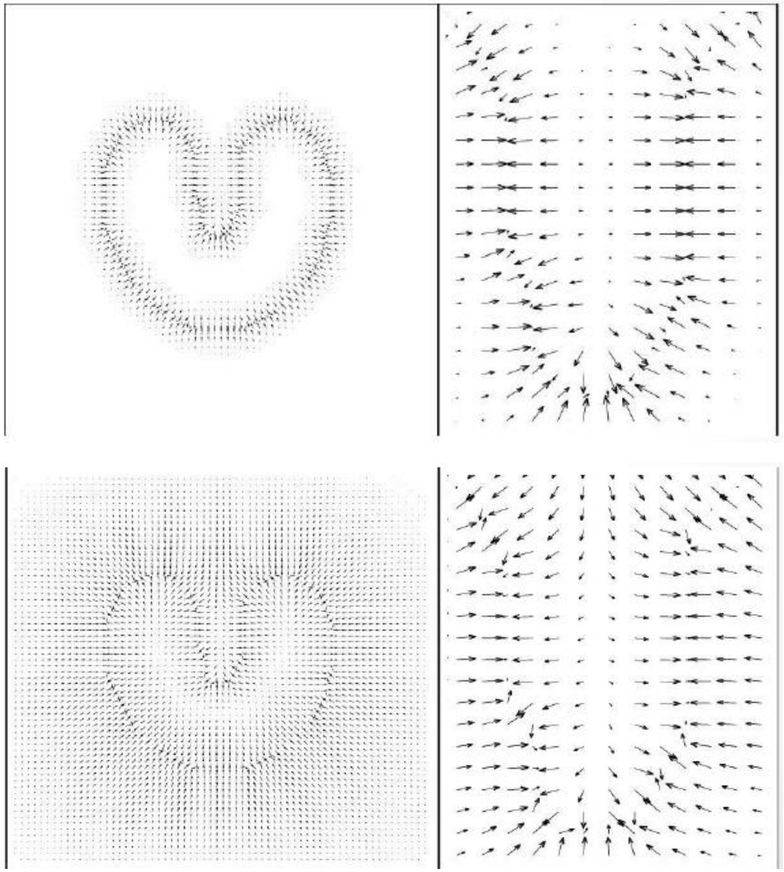
$$E = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |V - \nabla f|^2 \, dxdy$$

- f(x,y) is the edge map of the image

Boundary-based segmentation

Traditional force

GVF force



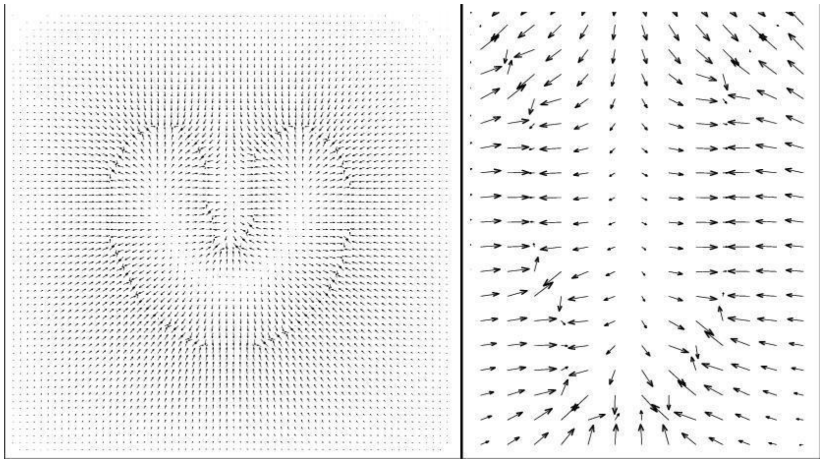
Boundary-based segmentation

GVF model:
- V(x,y) is defined such that it minimizes

$$E = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |V - \nabla f|^2 \, dxdy$$

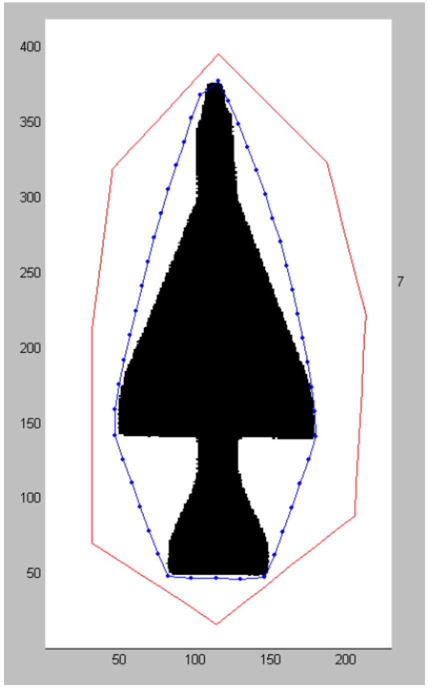
The norm of the gradients of V(x,y) must be small to minimize E
→ forces V(x,y) to become smooth

V(x,y) must be similar to the image gradient (external force) to minimize E.

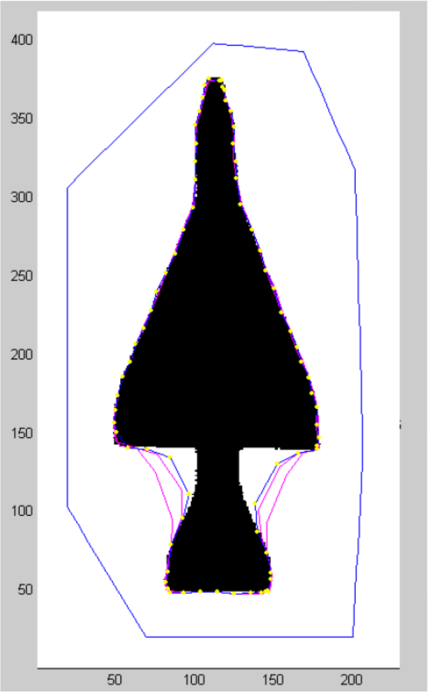


Boundary-based segmentation

Results:



Traditional snake

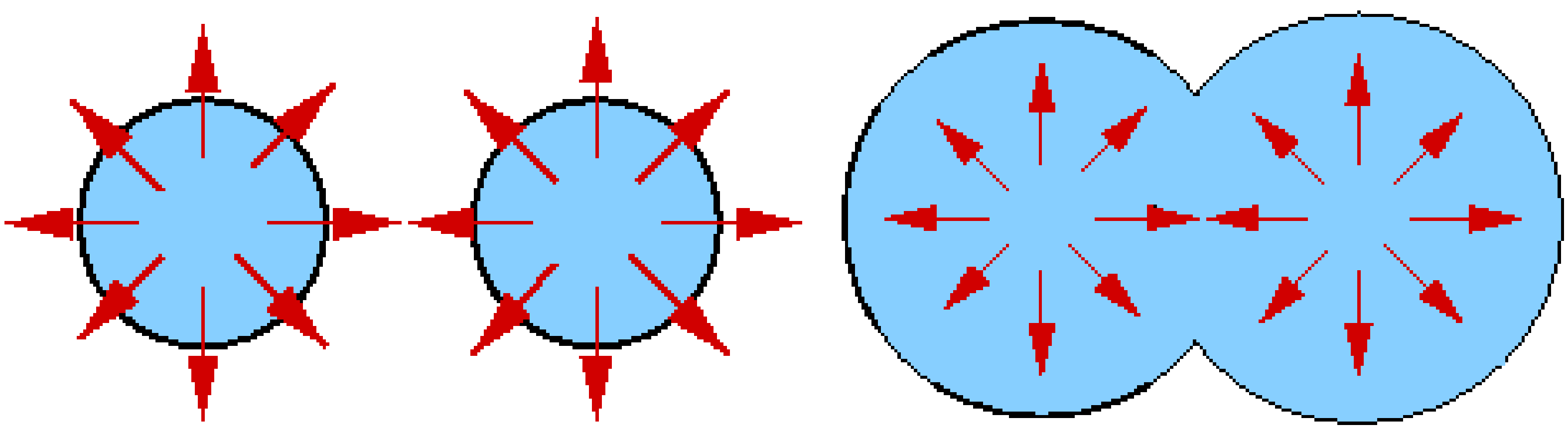


GVF model

Level sets

It is a contour evolution method
- Difficulties with snake-type methods
  - Hard to keep track of contour if it self-intersects during its evolution
  - Hard to deal with changes in topology

https://en.wikipedia.org/wiki/Level-set_method



Initial frame

Later in time

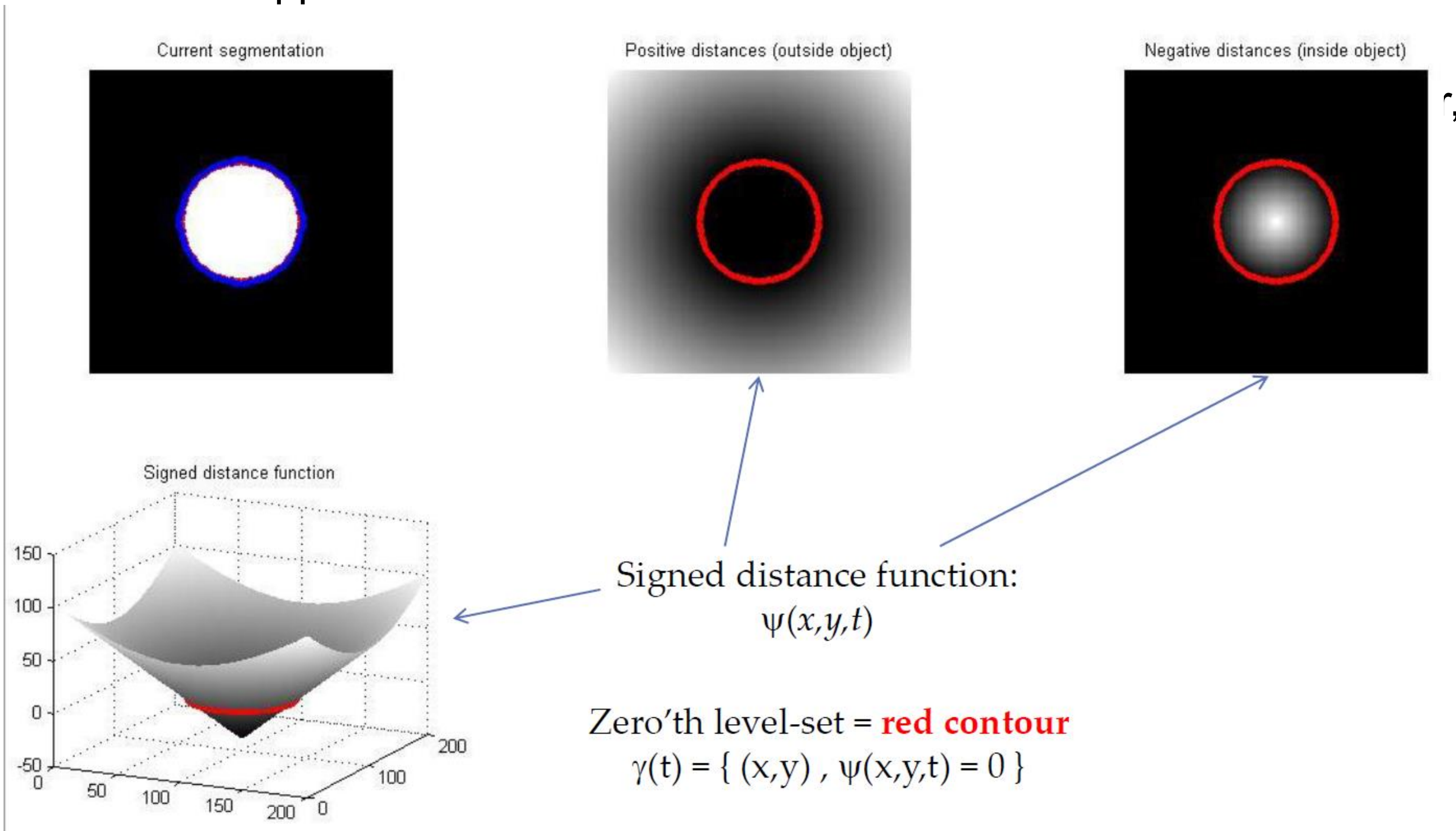
Level sets

The level set approach:

- Define problem in 1 higher dimension.
- Define level set function $\psi(x,y,t)$ where the (x,y) plane contains the contour, and $\psi(x,y,t)$ = signed Euclidean distance transform value (negative means inside closed contour, positive means outside contour)

Level sets

The level set approach:



Level sets

The level set approach:

- Define problem in 1 higher dimension.
- Define level set function $\psi(x,y,t)$ where the (x,y) plane contains the contour, and $\psi(x,y,t)$ = signed Euclidean distance transform value (negative means inside closed contour, positive means outside contour)
- define a function $\psi(x,y,t)$ so that at any time,

$$\gamma(t) = \{ (x,y) \text{ , } \psi(x,y,t) = 0 \}$$

  - there are many such ψ
  - ψ has many other level sets, more or less parallel to γ
  - only γ has a meaning for segmentation, not any other level set of ψ

Level sets

The level set approach:

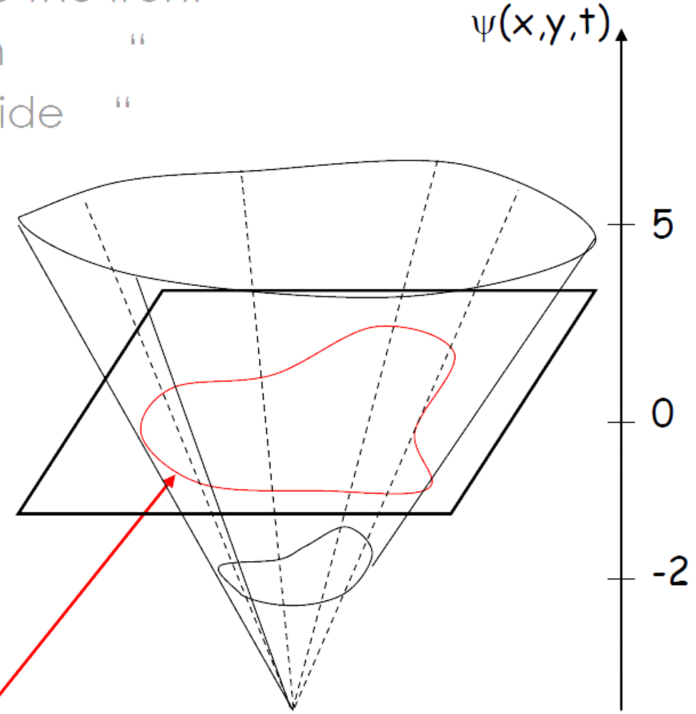
- Usual choice for ψ is the signed distance to the front γ(t)

$$\psi(x,y,t) = \begin{cases} -d(x,y,) & \text{if } (x,y) \text{ inside the front} \\ 0 & \text{" on "} \\ d(x,y) & \text{" outside "} \end{cases}$$

7	6	5	4	4	4	3	2	1	1	1	2	3	4	5
6	5	4	3	3	3	2	1	0	0	0	1	2	3	4
5	4	3	2	2	2	1	0	-1	-1	-1	0	1	2	3
4	3	2	1	1	1	0	-1	-2	-2	-2	-1	0	1	2
3	2	1	0	0	0	-1	-2	-3	-3	-2	-1	0	1	2
2	1	0	-1	-1	-1	-2	-3	-3	-2	-1	0	1	2	3
2	1	0	-1	-2	-2	-3	-3	-2	-1	0	1	2	3	4
2	1	0	-1	-2	-2	-2	-2	-1	0	1	2	3	4	5
3	2	1	0	-1	-1	-1	-1	-1	0	1	2	3	4	5
4	3	2	1	0	0	0	0	-1	-1	0	1	2	3	4
5	4	3	2	1	1	1	1	0	0	1	2	3	4	5
6	5	4	3	2	2	2	2	1	1	2	3	4	5	6

ψ(x,y,t)

γ(t)



Level sets

The level set approach:

7	6	5	4	4	4	3	2	1	1	1	2	3	4	5
6	5	4	3	3	3	2	1	0	0	0	1	2	3	4
5	4	3	2	2	2	1	0	-1	-1	-1	0	1	2	3
4	3	2	1	1	1	0	-1	-2	-2	-2	-1	0	1	2
3	2	1	0	0	0	-1	-2	-3	-3	-2	-1	0	1	2
2	1	0	-1	-1	-1	-2	-3	-3	-2	-1	0	1	2	3
2	1	0	-1	-2	-2	-3	-3	-2	-1	0	1	2	3	4
2	1	0	-1	-2	-2	-2	-2	-1	0	1	2	3	4	5
3	2	1	0	-1	-1	-1	-1	-1	0	1	2	3	4	5
4	3	2	1	0	0	0	0	-1	-1	0	1	2	3	4
5	4	3	2	1	1	1	1	0	0	1	2	3	4	5
6	5	4	3	2	2	2	2	1	1	2	3	4	5	6

$\psi(x,y,t)$

- no movement, only change of values
- the front may change its topology
- the front location may be between samples

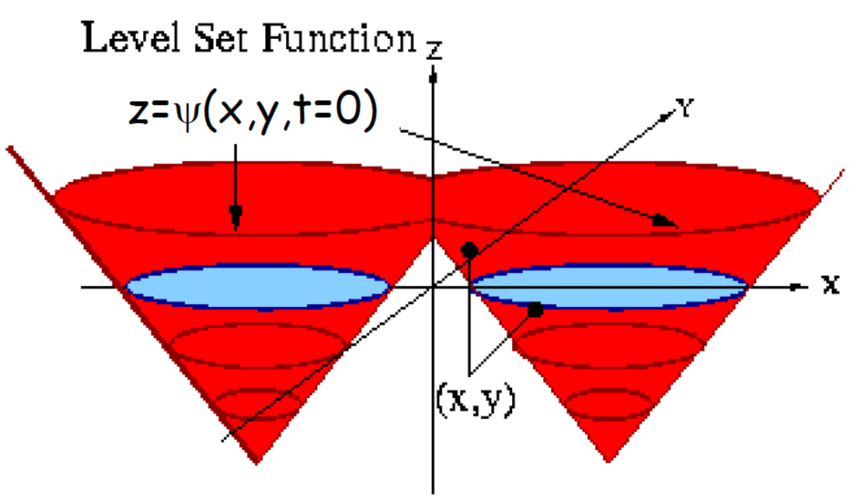
$$\psi(x,y,t+1) = \psi(x,y,t) + \Delta\psi(x,y,t)$$

7	6	5	4	4	4	3	2	1	1	1	2	3	4	5
6	5	4	3	3	3	2	0	-1	0	0	1	2	3	4
5	4	3	2	2	2	1	-1	-2	-1	-1	0	1	2	3
4	3	2	1	1	1	0	-1	-2	-2	-2	-1	0	1	2
3	2	1	0	0	0	-1	-2	-3	-3	-2	-1	0	1	2
2	1	0	-1	-1	-1	-2	-3	-3	-2	-1	0	1	2	3
2	1	0	-1	-2	-2	-3	-3	-2	-1	0	1	2	3	4
2	1	1	0	-2	-2	-2	-2	0	0	1	2	3	4	5
3	2	1	0	-1	-3	-1	0	1	1	1	2	3	4	5
4	3	2	0	-1	-2	0	1	1	0	0	2	2	3	4
5	4	3	2	0	0	1	1	0	-1	0	1	2	4	5
6	5	4	3	2	1	2	2	0	0	1	2	4	5	6

Level sets

Merging of contours

- The zero level set (in blue) at one point in time as a slice of the level set surface (in red)

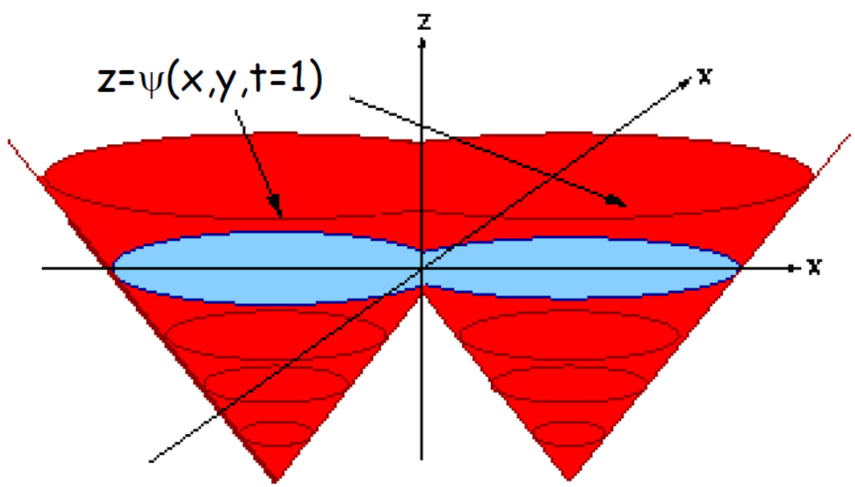


The Level Set Surface (in red) plots the distance from each point (x, y) to the Interface (in blue)

Level sets

Merging of contours

- The zero level set (in blue) at one point in time as a slice of the level set surface (in red)
- Later in time the level set surface (red) has moved and the new zero level set (blue) defines the new contour



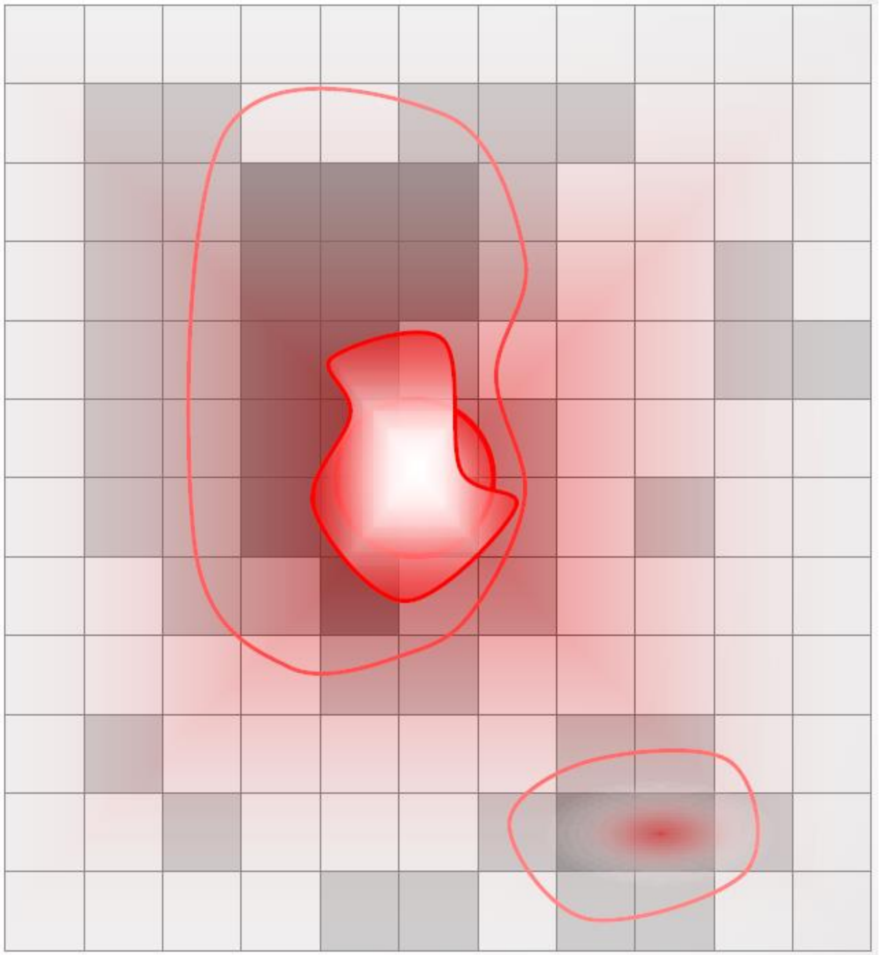
Later in Time: Red Level Set Surface has moved, yielding new Blue Interface

Level sets

Segmentation with Level Set

- Initialize the front $\gamma(0)$
- Compute $\psi(x,y,0)$
- Iterate:

  $\psi(x,y,t+1) = \psi(x,y,t) + \Delta\psi(x,y,t)$ until convergence
- Mark the front $\gamma(t_{end})$



Level sets

How to move the level set

- When the curve passes over places where the image gradient is small, we let the curve expand quickly.
- When the curve passes over places where the image gradient is large, we suspect we are near the boundary, and slow the curve down.
- In addition, we include a curvature term to the speed to add a little surface tension to the expanding contour

Level sets

How to move the level set

1. Define a velocity field, F, that specifies how contour points move in time
   - Based on application-specific physics such as time, position, normal, curvature, image gradient magnitude
2. Build an initial value for the level set function, $\psi(x,y,t=0)$, based on the initial contour position
3. Adjust ψ over time; current contour defined by $\psi(x(t), y(t), t) = 0$

Level sets

Equation of motion:

- Constrain the level set value of a point on the contour with motion x(t) to be equal to zero $\psi(x(t),t) = 0$
- By the chain rule (or the first order Taylor expansion)

$$\psi_t + \nabla\psi(x(t), t) \cdot x'(t) = 0$$

- Choosing the velocity field, F, such that it supplies the speed in the outward normal direction, we have

$$x'(t) \cdot n = F, \text{ where } n = \nabla\psi / |\nabla\psi|$$

- Hence the evolution equation ψ is

$$\psi_t + F|\nabla\psi| = 0$$

Level sets

Speed function

$$F(K) = F_0 + F_1(k) = (1 - \varepsilon K)$$

$$F(K) = k_I(x,y) * (1 - \varepsilon K)$$

$$k_I = \frac{1}{1 + \left|\nabla G_\sigma * I(x,y)\right|}$$

$$k_I = e^{-\left|\nabla G_\sigma * I(x,y)\right|}$$

Level sets

Equation of motion

product of influences

$$\frac{\partial \psi}{\partial t} + \hat{k}_I \cdot \left(F_A + F_G(\kappa)\right) \cdot \left\|\nabla \psi\right\| = 0$$

$\psi(x,y,t+1) - \psi(x,y,t)$

extension of the speed function k_I
(image influence)

constant "force"
(balloon pressure)

smoothing "force" depending on the local curvature κ
(contour influence)

spatial derivative of ψ

$$\kappa = \operatorname{div}\left(\frac{\nabla \psi}{\left\|\nabla \psi\right\|}\right)$$

link between spatial and temporal derivatives,
but not the same type of motion as contours!

Level sets

Equation of motion

$$\frac{\partial \psi}{\partial t} + \hat{k}_I \cdot \left(F_A + F_G(\kappa)\right) \cdot \left\|\nabla \psi\right\| = 0$$

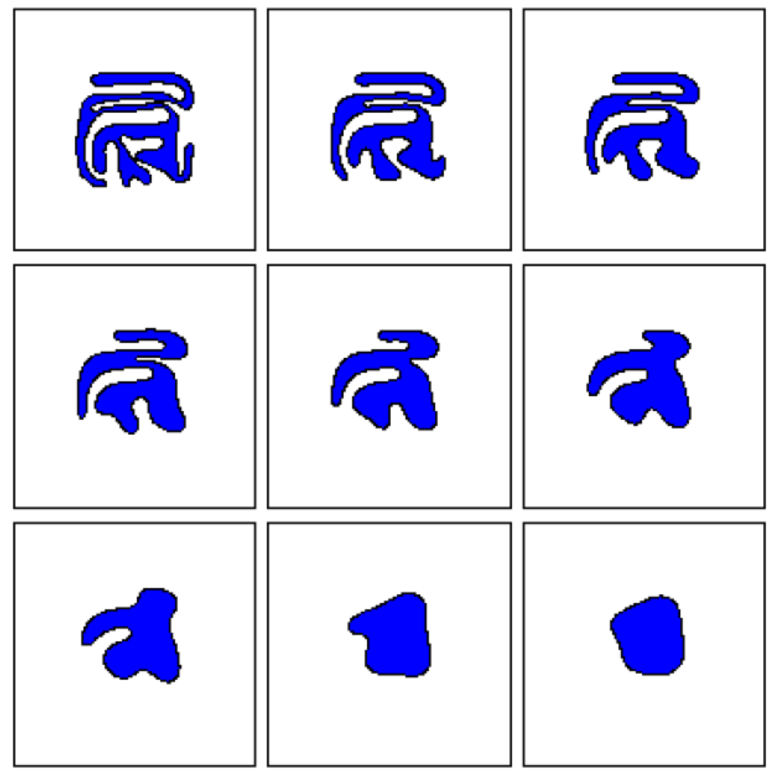
$$\boxed{\psi(x,y,t+1) - \psi(x,y,t)}$$

$$\psi(x,y,t+1) = \psi(x,y,t) - \hat{k}_I(x,y) \cdot \left(F_A + F_G(x,y)\right) \cdot \left\|\nabla \psi(x,y)\right\|$$

Level sets

Example of shape simplification:

- $F = 1 - 0.1\kappa$ where κ is the curvature at each contour point

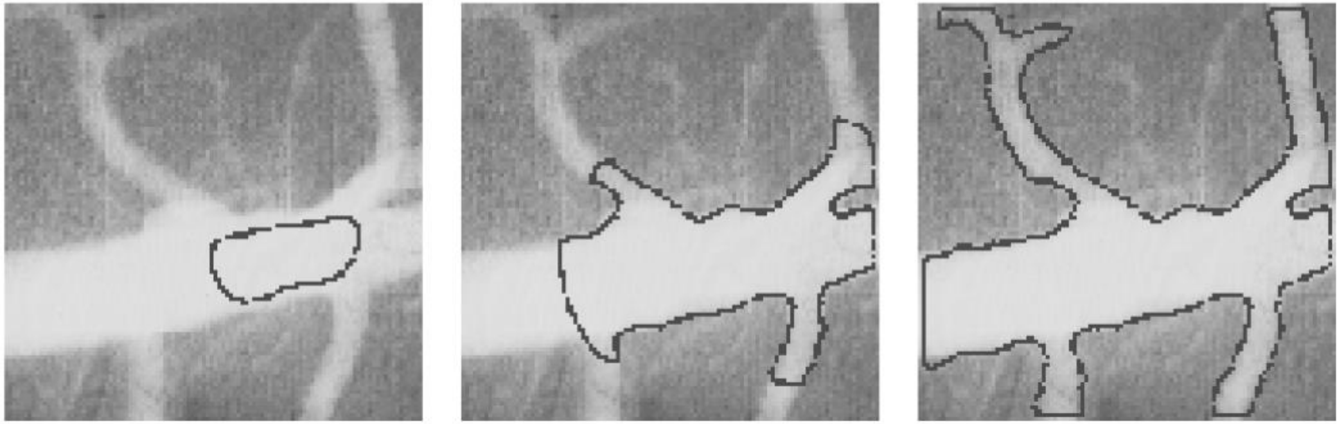


Motion under Curvature: Collapse of a Curve to a Single Point.

Level sets

Example of segmentation

- Digital Subtraction Angiogram
- F based on image gradient and contour curvature



Evolving Front Driven by Function of Image Gradient.