SEGMENTATION

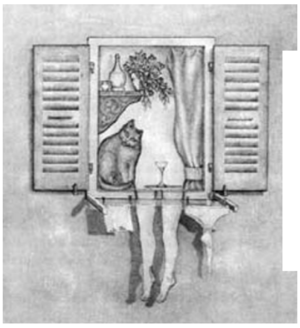
Readings:

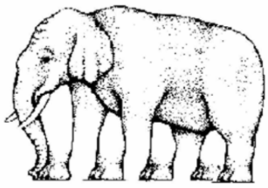
Szeliski: **Chapter 5 (5.2,5.3,5.4)**
Chapter 4 (4.2.2, 4.3.2)

Some Slides adapted from Univ. of Washington
http://www.cs.washington.edu/education/courses/cse576/08sp/

It can get a lot harder

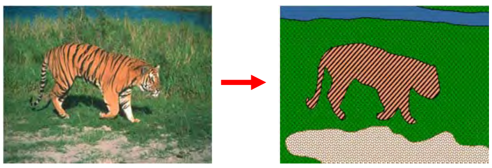
Brady, M. J., & Kersten, D. (2003). Bootstrapped learning of novel objects. J Vis, 3(6), 413-422





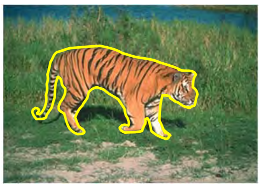
1970s: R. C. James

From images to objects



- What Defines an Object?
  - Subjective problem, but has been well-studied
  - Gestalt Laws seek to formalize this
    - proximity, similarity, continuation, closure, common fate

Extracting objects



- How could this be done?

Segmentation

- Definition:
  Partition Image in Regions Si such that following conditions are satisfied
  (H is homogeneity Criterion)

  1. $I = \bigcup_{i=1}^{m} S_i$
  2. $S_i \cap S_j = \emptyset$ for all $i \neq j$
  3. $H(S_i) = TRUE$ For all i
  4. $H(S_i \cup S_j) = FALSE$ for neighboring S_i and S_j , $i \neq j$

- Neither partition nor number of Segments (m) needs to unique
- m need not be minimal

Segmentation

Thresholding
- Simple & Fast

Edge based segmentation
- How to find closed edges

Region-based Methods
- Region Growing
- Split & Merge
- Clustering
- Mean-Shift

Thresholding

Partition image according to homogeneity (gray value, color, etc.)

Partition according to gray value (i.e. partition Histogram
→ cf. Relation to Quantization

Main Question:
- How to determine the Threshold

Thresholding

g(i,j) =1 for f(i,j)>T
=0 otherwise

Global Threshold

T=T(f)

Local Threshold

$T=T(f,f_c)$ (f_c part of image where T is determined)

Methods for Thresholding

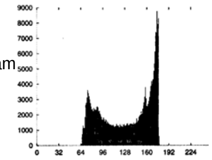
If a priori information is known the selection of the threshold is simplified

Example1: Printed Text p% of page are covered with letters
→ set Threshold (p-tile thresholding) such that %p Pixels are black

Example2: Knowledge about line width
→ set Threshold such that lines have required width

Threshold Selection

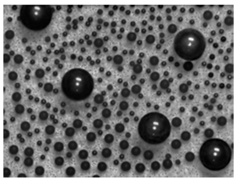
- More complex methods are based on histogram analysis
- If object is uniform (in gray value) and differs from background gray value
  - ➡ Histogram is **bimodal**
  - ➡ Object pixels form Peak in Histogram
  - ➡ Background Pixels form a second Peak in Histogram

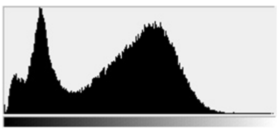


  - ➡ Gray values in between peaks are not common (border pixels between the object and background)
  - ➡ Select Threshold according to minimum histogram value between maxima

Histogram-based segmentation

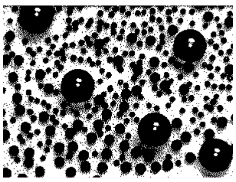
- Goal
  - Break the image into K regions (segments)
  - Solve this by reducing the number of colors to K and mapping each pixel to the closest color

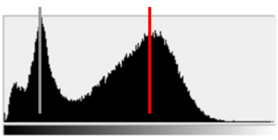




Histogram-based segmentation

- Goal
  - Break the image into K regions (segments)
  - Solve this by reducing the number of colors to K and mapping each pixel to the closest color





Here's what it looks like if we use two colors

Threshold Selection

- Difficult to interpret significance of maxima/minima
- Bimodal Algorithms, detect first highest maxima, select threshold as minimum between these local (**mode method**)
- Minium Distance between Maxima is enforced.
- Histogram smoothing, helps in detecting significant Maxima/Minima
- When constructing histogram use also local neighborhood information (pixels without/with high gradient) to get clearer maxima/minima

Optimal Thresholding

Assume distribution (normal) of gray-values

→ Fit distributions and determine Threshold

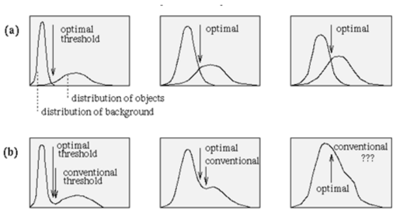


Figure 5.4 *Grey level histograms approximated by two normal distributions; the threshold is set to give minimum probability of segmentation error: (a) Probability distributions of background and objects, (b) corresponding histograms and optimal threshold.*

Iterative Optimal Thresholding

1. Assume corner pixels contain background
2. Step t compute μ^t_B and μ^t_O mean BG & Object gray-level where B and O is obtained using threshold T^t

$$\mu^t_B = \sum\nolimits_{(i,j)\in B} f(i,j)/|B|$$

$$\mu^t_O = \sum\nolimits_{(i,j)\in O} f(i,j)/|O|$$

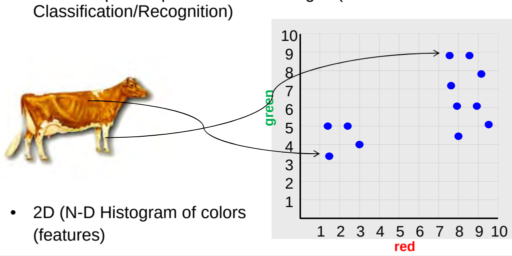
3. Set $T^{t+1} = \dfrac{\mu^t_B + \mu^t_O}{2}$

4. If $T^{t+1}=T^t$ halt, otherwise go to 2

Color Image Segmentation

Segmentation as clustering

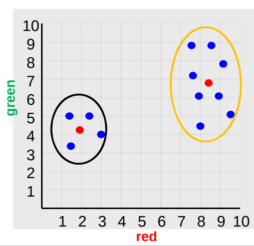
- E.g. Color image segmentation
- Group pixels with similar color
- Feature space representation of images (cf. Classification/Recognition)



- 2D (N-D Histogram of colors (features)

Segmentation as clustering

- Clusters in Feature space correspond to segmentations in image space
- Same as Thresholding but in higher dim. Spaces



- How to find clusters?

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

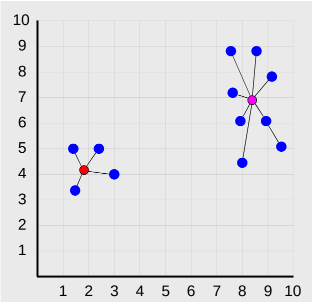
- Point-Cluster distance
  - single-link clustering
  - complete-link clustering
  - group-average clustering
- Dendrograms
  - yield a picture of output as clustering process continues

Squared Error

$$se_{K_i} = \sum_{j=1}^{m} \|t_{ij} - C_k\|^2$$

$$se_K = \sum_{j=1}^{k} se_{K_j}$$

Objective Function

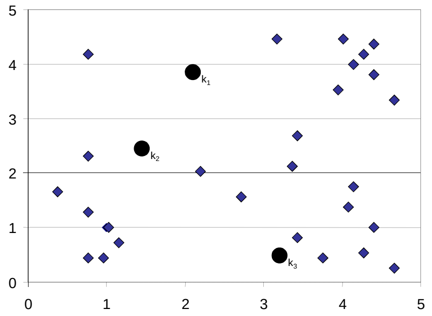


Algorithm *k-means*

1. Decide on a value for *k*.
2. Initialize the *k* cluster centers (randomly, if necessary).
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.
4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.
5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise goto 3.

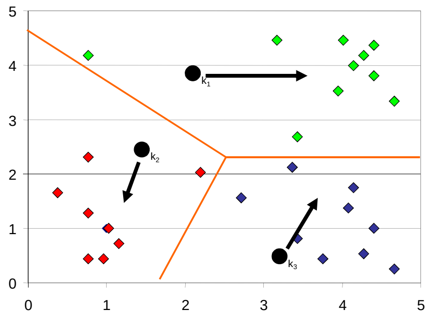
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



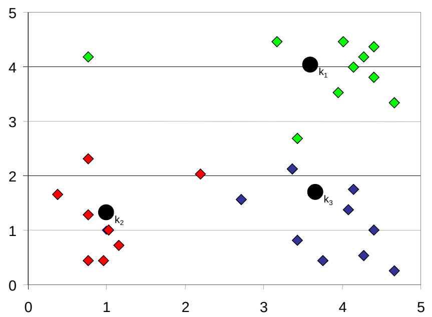
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



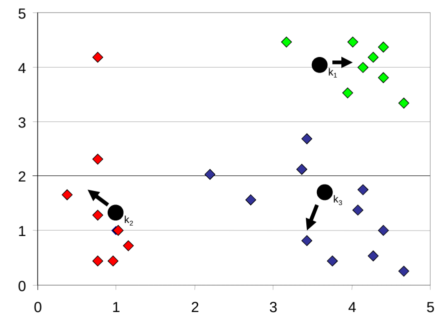
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



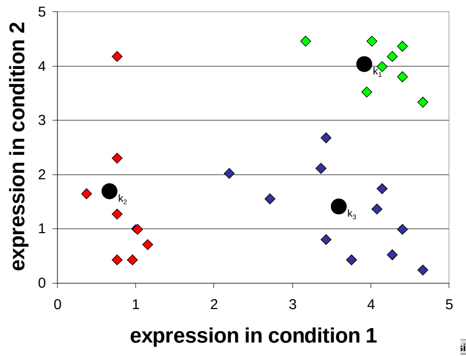
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



Comments on the *K-Means* Method

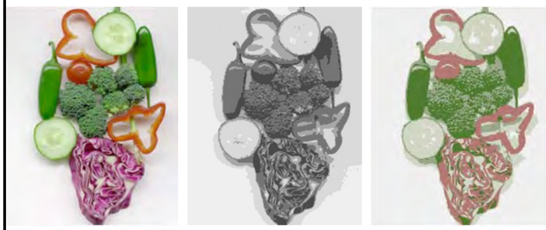
Strength

- *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*

Weakness

- Need to specify *k,* the *number* of clusters, in advance
- Unable to handle noisy data and *outliers*
- Not suitable to discover clusters with *non-convex shapes*

Image Clusters on intensity Clusters on color



K-means clustering using intensity alone and color alone

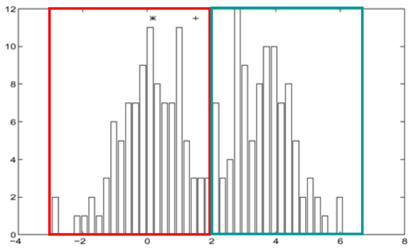
Including spatial relationships

Augment data to be clustered with spatial coordinates.

$$z = \begin{pmatrix} Y \\ u \\ v \\ x \\ y \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{color coordinates} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{spatial coordinates} \end{matrix}$$

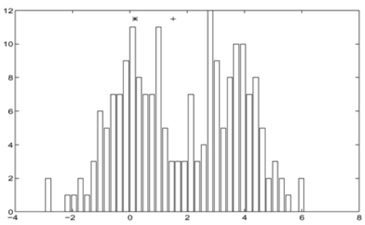
Mean Shift

Finding Modes in a Histogram



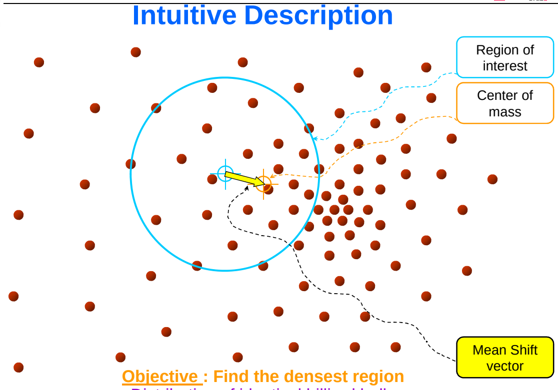
- How Many Modes Are There?
  - Easy to see, hard to compute

Mean Shift [Comaniciu & Meer]



- Iterative Mode Search
  1. Initialize random seed, and window W
  2. Calculate center of gravity (the "mean") of W: $\sum_{x \in W} xH(x)$
  3. Translate the search window to the mean
  4. Repeat Step 2 until convergence

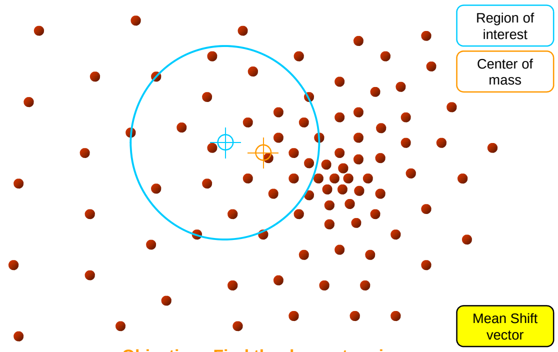
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

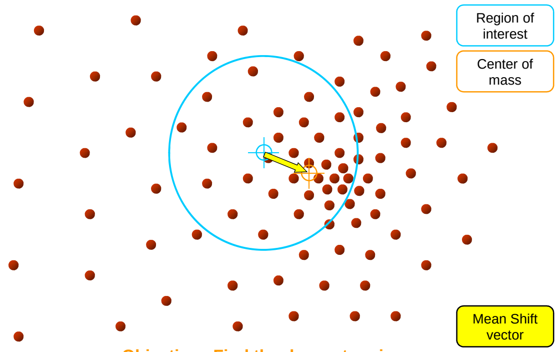
Intuitive Description



Objective : Find the densest region

Distribution of identical billiard balls

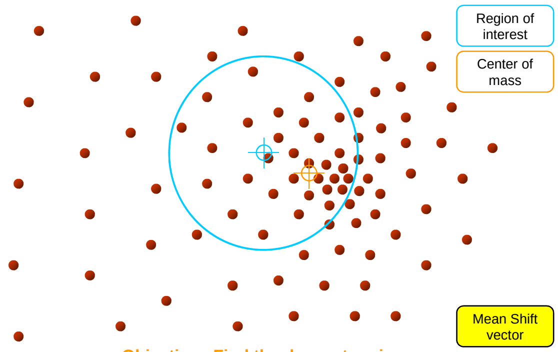
Intuitive Description



Objective : Find the densest region

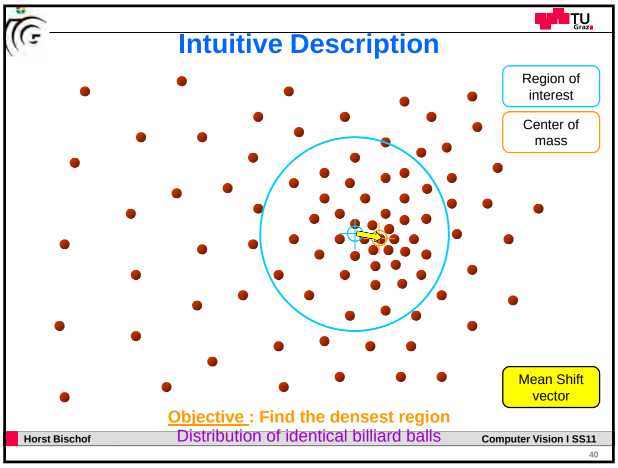
Distribution of identical billiard balls

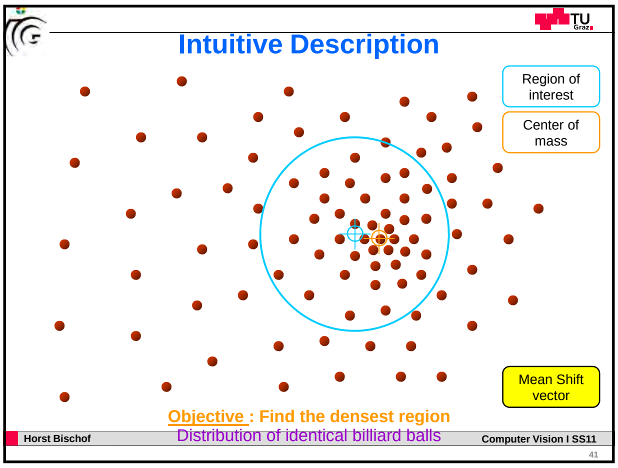
Intuitive Description

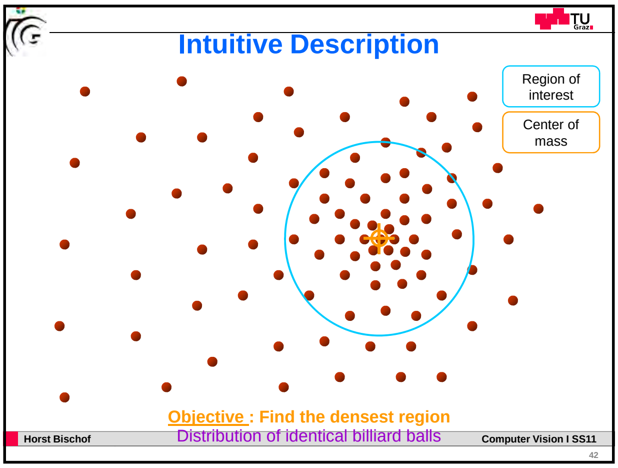


Objective : Find the densest region

Distribution of identical billiard balls


Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Horst Bischof
Computer Vision I SS11
40

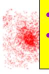

Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls
Horst Bischof
Computer Vision I SS11
41

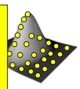

Intuitive Description
Region of interest
Center of mass
Objective : Find the densest region
Distribution of identical billiard balls
Horst Bischof
Computer Vision I SS11
42

What is Mean Shift ?

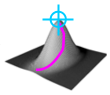
A tool for:
Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in R^N

PDF in feature space
- **Color space**
- **Scale space**
- **Actually any feature space you can conceive**
- **…**

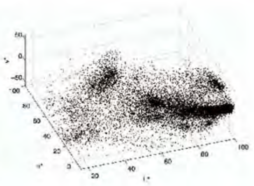




Non-parametric
Density **GRADIENT** Estimation
(Mean Shift)



Clustering
Real Example

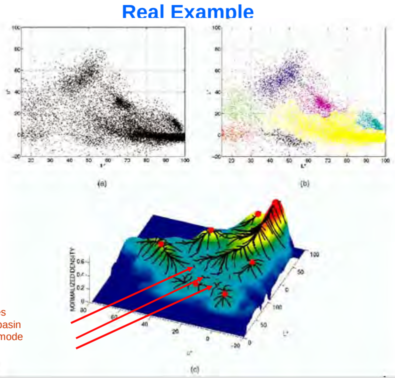


L*u*v space representation

Clustering
Real Example

2D (L*u) space representation

Final clusters



Not all trajectories in the attraction basin reach the same mode

Segmentation
Example

Segmentation
Example

Segmentation
Example

Segmentation
Example

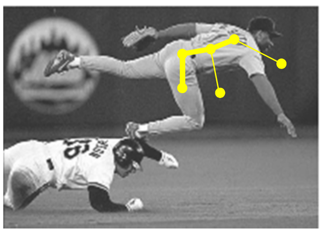
Horst Bischof — Computer Vision I SS11 — 49

A different way of thinking about segmentation…

Horst Bischof — Computer Vision I SS11 — 50

Graph-Theoretic Image Segmentation

Build a weighted graph G=(V,E) from image



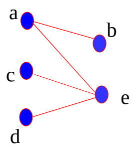
V: image pixels

E: connections between pairs of nearby pixels

Segmentation = graph partition

Horst Bischof — Computer Vision I SS11 — 51

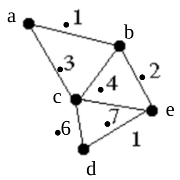
Graphs Representations



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

Adjacency Matrix

A Weighted Graph and its Representation



Affinity Matrix

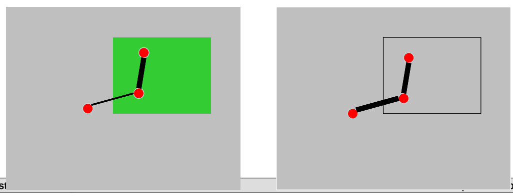
$$
W = \begin{bmatrix} 1 & .1 & .3 & 0 & 0 \\ .1 & 1 & .4 & 0 & .2 \\ .3 & .4 & 1 & .6 & .7 \\ 0 & 0 & .6 & 1 & 1 \\ 0 & .2 & .7 & 1 & 1 \end{bmatrix}
$$

Affinity between pixels

Similarities among pixel descriptors

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

Scale factor… it will hunt us later



Affinity between pixels

Similarities among pixel descriptors

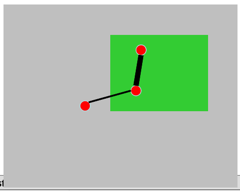
$$W_{ij} = \exp(-\| z_i - z_j \|^2 / \sigma^2)$$

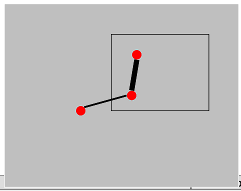
σ = Scale factor…
it will hunt us later

Interleaving edges

$$W_{ij} = 1 - \max_{\text{Line between i and j}} Pb$$

With Pb = probability of boundary

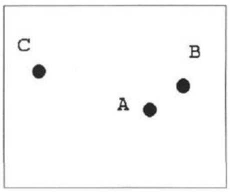




Feature grouping by "relocalisation" of eigenvectors of the proximity matrix

Guy L. Scott
Robotics Research Group
Department of Engineering Science
University of Oxford

H. Christopher Longuet-Higgins
University of Sussex
Falmer
Brighton



Three points in feature space

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

With an appropriate σ

W=

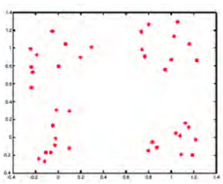
	A	B	C
A	1.00	0.63	0.03
B	0.63	1.00	0.0
C	0.03	0.0	1.00

The eigenvectors of W are:

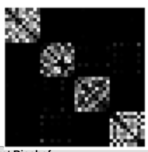
	E_1	E_2	E_3
Eigenvalues	1.63	1.00	0.37
A	-0.71	-0.01	0.71
B	-0.71	-0.05	-0.71
C	-0.04	1.00	-0.03

The first 2 eigenvectors group the points as desired…

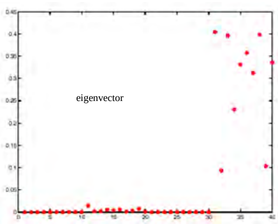
Example eigenvector



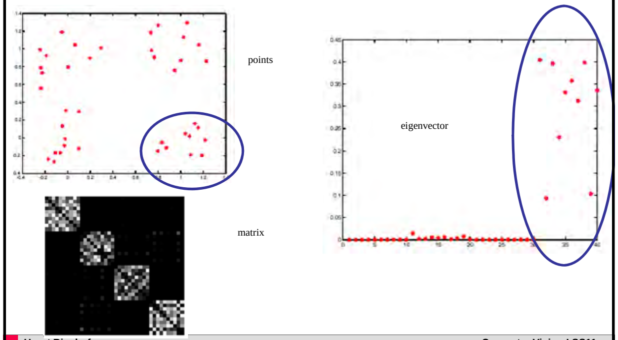
points



matrix

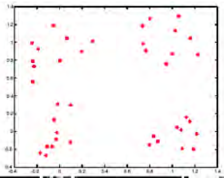


Example eigenvector

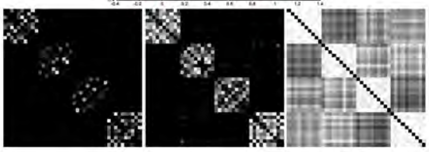


Scale affects affinity

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



σ=.2



σ=.1 σ=.2 σ=1

Global optimization

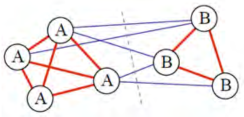
- In this formulation, the segmentation becomes a global process.

Boundaries of image regions defined by a number of attributes

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

[Malik]

Some Terminology for Graph Partitioning



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 $\mathrm{A} \cap \mathrm{B} = \varnothing$

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

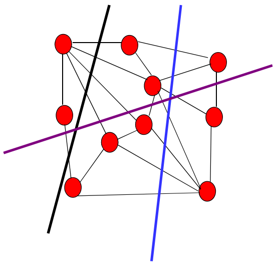
$$assoc(\mathrm{A},\mathrm{B}) = \sum_{u \in \mathrm{A}, v \in \mathrm{B}} \mathrm{W}(u,v)$$

A and B not necessarily disjoin

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

With V = the set of all nodes

Minimum Cut



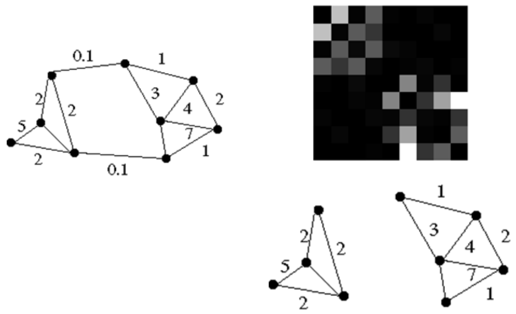
A cut of a graph G is the set of edges S such that removal of S from G disconnects G.

Minimum cut is the cut of minimum weight, where weight of cut <A,B> is given as

$$w(\langle A,B \rangle) = \sum_{x \in A, y \in B} w(x,y)$$

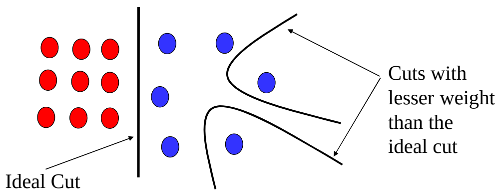
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Minimum Cut and Clustering



Drawbacks of Minimum Cut

- Weight of cut is directly proportional to the number of edges in the cut.



Normalized cuts

- 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

- Minimize

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

where cut(A,B) is sum of weights with one end in A and one end in B; assoc(A,V) is sum of all edges with one end in A.

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

Solving the Normalized Cut problem

- 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.

[Malik]

Normalized Cut As Generalized Eigenvalue problem

$$Ncut(\mathrm{A},\mathrm{B}) = \frac{cut(\mathrm{A},\mathrm{B})}{assoc(\mathrm{A},\mathrm{V})} + \frac{cut(\mathrm{A},\mathrm{B})}{assoc(\mathrm{B},\mathrm{V})} \qquad D_{ii} = \sum_j A_{ij}$$

$$= \frac{(1+x)^T(D-W)(1+x)}{k1^T D1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k)1^T D1}; \; k = \frac{\sum_{x_i>0} D(i,i)}{\sum_i D(i,i)}$$

$$= \ldots$$

after simplification, Shi and Malik derive

$$Ncut(A,B) = \frac{y^T(D-W)y}{y^T Dy}, \quad \text{with } y_i \in \{1,-b\}, y^T D1 = 0.$$

W = affinity matrix, "A"

[Malik]

Normalized cuts

- Instead, solve the generalized eigenvalue problem

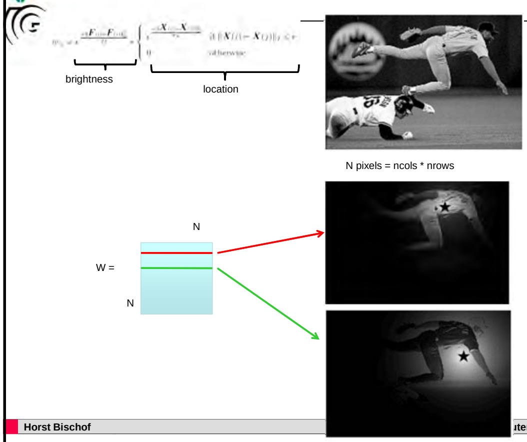
$$\max_y \left(y^T(D-W)y\right) \text{ subject to } \left(y^T Dy = 1\right)$$

- which gives

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

- They show that the 2nd smallest eigenvector solution y is a good real-valued approx to the original normalized cuts problem. Then you look for a quantization threshold that maximizes the criterion --- i.e all components of y above that threshold go to one, all below go to -b

http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf



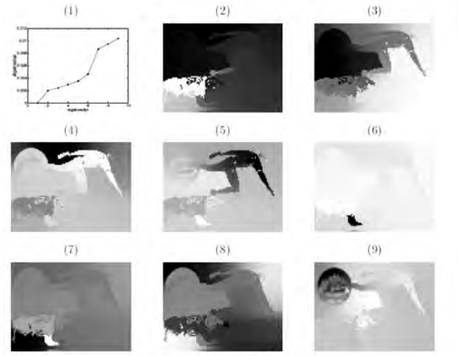


Figure 12: Subplot (1) plots the smallest eigenvectors of the generalized eigenvalue system (11). Subplot (2) – (9) shows the eigenvectors corresponding the 2nd smallest to the 9th smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

Edge-based Segmentation

Describe the border of region
- Dual to region

Edge Detector (+ Threshold selection)

How to obtain closed curve (border tracing)
How to fill holes on contour

Edge-based Segmentation

- Large group of methods based on information about edges in the image
- Rely on edges found in an image by edge detecting operators
- Edges mark image locations of discontinuities in gray level, color, texture, ..
- Edge detection results cannot be used as a segmentation result
- Edge segments → borders

Edge Thresholding

- Global Threshold
- Histogram Analyis
- p-tile Threshold
- Non maximum suppression (directional data)
- Hysteresis to filter output

cf. Canny Edge Detector

Edge-Aggregation

- Local edges (edgels) shall be aggregated to longer segments
- Global Method
  - Formulate Problem as search problem for minimal cost path (A*-Algorithm)
  - Costs based on Magnitude, Distance, Orientation
- Local Methods
  - Iteratively add edgels
  - Take best among neighbors

Parametric Model Fitting

- Fit a parametric representation to edgels
  - Line, Circle, etc.

- Hough Transform
  - Aggregation of local information
  - Lines, Circles

Hough transform

- An early type of voting scheme
- General outline:
  - Discretize parameter space into bins
  - For each feature point in the image, put a vote in every bin in the parameter space that could have generated this point
  - Find bins that have the most votes

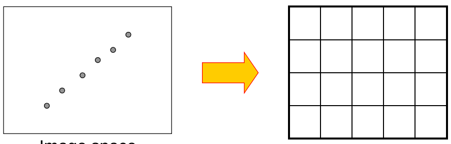


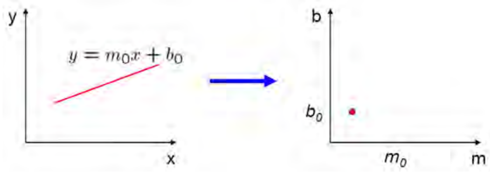
Image space Hough parameter space

P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Parameter space representation

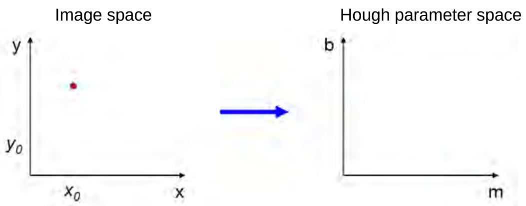
- A line in the image corresponds to a point in Hough space

Image space Hough parameter space



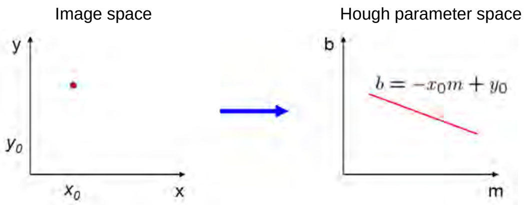
Parameter space representation

- What does a point (x_0, y_0) in the image space map to in the Hough space?



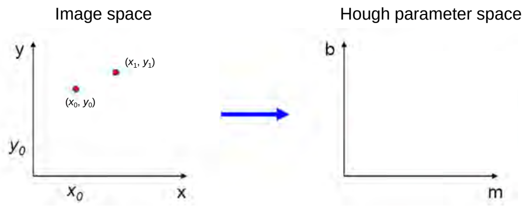
Parameter space representation

- What does a point (x_0, y_0) in the image space map to in the Hough space?
  - Answer: the solutions of $b = -x_0 m + y_0$
  - This is a line in Hough space



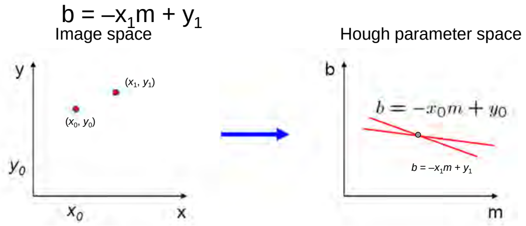
Parameter space representation

- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?



Parameter space representation

- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?
  - It is the intersection of the lines $b = -x_0 m + y_0$ and $b = -x_1 m + y_1$

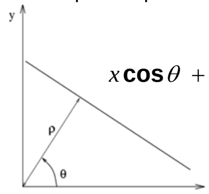


Parameter space representation

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m

Parameter space representation

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m
- Alternative: polar representation

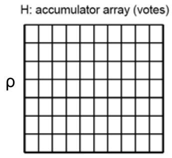


$$x \cos\theta + y \sin\theta = \rho$$

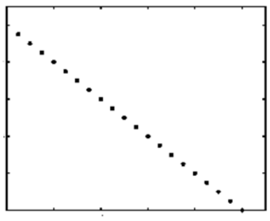
Each point will add a sinusoid in the (θ,ρ) parameter space

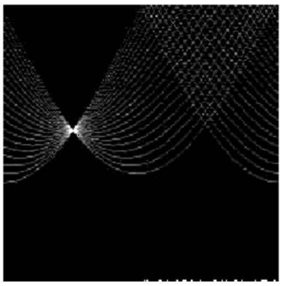
Algorithm outline

- Initialize accumulator H to all zeros
- For each edge point (x,y) in the image
  For θ = 0 to 180
  $\rho = x \cos \theta + y \sin \theta$
  $H(\theta, \rho) = H(\theta, \rho) + 1$
  end
  end
- Find the value(s) of (θ, ρ) where H(θ, ρ) is a local maximum
  – The detected line in the image is given by $\rho = x \cos \theta + y \sin \theta$



Basic illustration

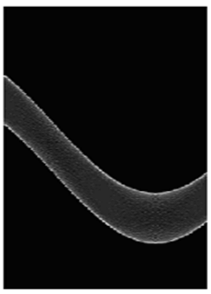

features


votes

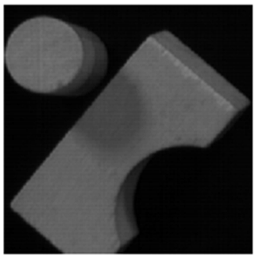
Other shapes

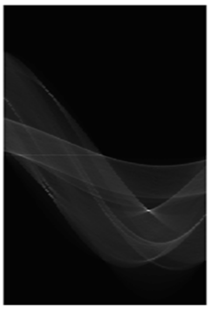
Square

Circle

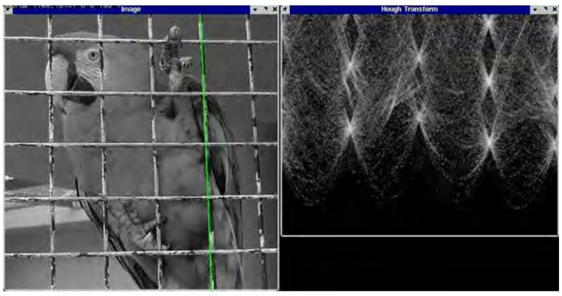


Several lines



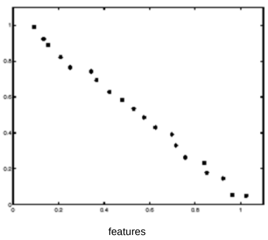


A more complicated image

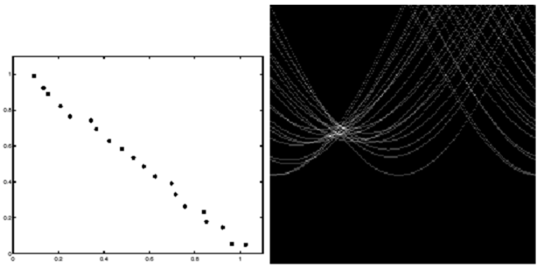


http://ostatic.com/files/images/ss_hough.jpg

Effect of noise



Effect of noise

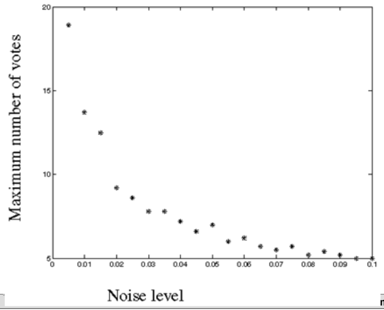


features votes

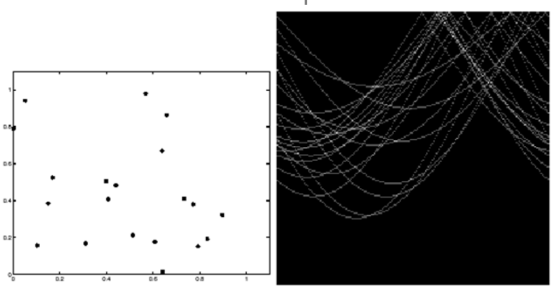
- Peak gets fuzzy and hard to locate

Effect of noise

- Number of votes for a line of 20 points with increasing noise:



Random points

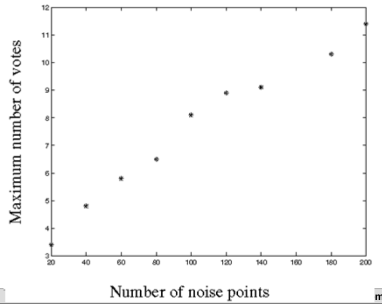


features votes

- Uniform noise can lead to spurious peaks in the array

Random points

- As the level of uniform noise increases, the maximum number of votes increases too:



Dealing with noise

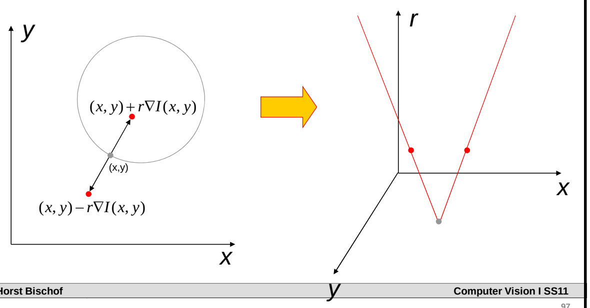
- Choose a good grid / discretization
  - Too coarse: large votes obtained when too many different lines correspond to a single bucket
  - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets
- Increment neighboring bins (smoothing in accumulator array)
- Try to get rid of irrelevant features
  - Take only edge points with significant gradient magnitude

Hough transform for circles

- How many dimensions will the parameter space have?
- Given an oriented edge point, what are all possible bins that it can vote for?

Hough transform for circles

image space | Hough parameter space



Hough-Transformation: Erweiterungen

- Das Prinzip der Geraden-Detektion kann erweitert werden zu höheren Merkmalen
  z.B. **Parabeln** **Kreise**
  $y = ax^2 + bx + c$ $(x-a)^2 + (y-b)^2 = r^2$
- Berechnungskomplexität und die Akkumulatorgröße steigen polynomiell mit der Anzahl der Parameter → klassische Hough-Technik ist nur für einfache Kurven praktisch einsetzbar
- **Beschleunigungen:**

Hierarchical Hough-Transform: Verarbeitung in mehreren Auflösungen
Randomized Hough-Transform: Auf Grund der Robustheit wird nur eine zufällig ausgewählte Untermenge der Features berücksichtigt.

Region-based Segmentation

- Edge-based segmentation: borders between regions
- Region-based segmentation: direct construction of regions
- Easy to construct regions from their borders and easy to detect borders of regions
- Combination of results may be a good idea
- Region growing techniques are generally better in noisy images where edges are extremely difficult to detect

Region-based Segmentation

- Homogeneity of regions is used as the main segmentation criterion in region growing
- The criteria for homogeneity:
  - gray level
  - color, texture
  - shape
  - model
  - etc.

Region Definition

- 4 Conditions

1. $I = \bigcup_{i=1}^{m} S_i$
2. $S_i \cap S_j = \emptyset$ for all $i \neq j$
3. $H(S_i) = TRUE$ for all i
4. $H(S_i \cup S_j) = FALSE$ neighboring S_i and S_j, $i \neq j$

- Resulting regions of the segmented image must be both homogeneous and maximal

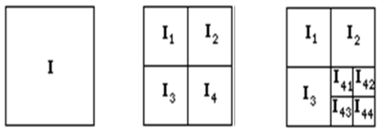
Region Merging

- Start at small segment (e.g. one pixel) satisfying homogeneity criterion
- Merge adjacent regions that satisfy merging criterion
- If no regions can be merged stop

- Different algorithms depending on starting and order and merging criterion

Region Splitting

Start with whole
Split region if not homogeneous
Stop if no region can be split anymore



Split-and-Merge Algorithm

- Combination of splitting and merging advantages of both approaches.
- Split-and-merge use pyramid image representations; regions are square-shaped and correspond to elements of the appropriate pyramid level.

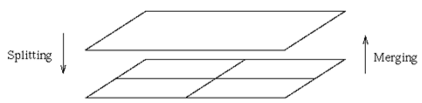
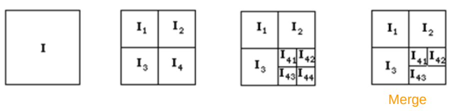


Figure 5.44 *Split-and-merge in a hierarchical data structure.*

Split-and-Merge Algorithm

Data Structure is Pyramid/Quadtree
If region is not homogeneous split (4 children)
If all 4 regions with same parent can be merged, then merge.
If no split or merge is possible, merge any two adjacent regions that are homogeneous (even on different pyramid levels)



Quadtree Representation

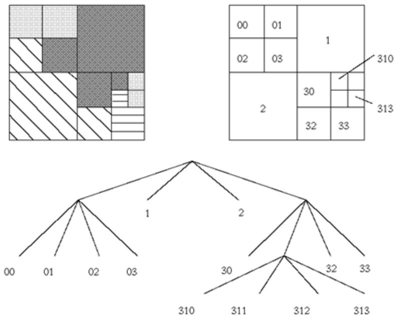


Figure 5.45 *Segmentation quadtree.*

Split-and-Merge Algorithm

Original

After Split

After merge

Edges