Segmentation I

Jan Kybic

2013-10-14

Outline

Introduction

Individual pixel classification

Region-based algorithms

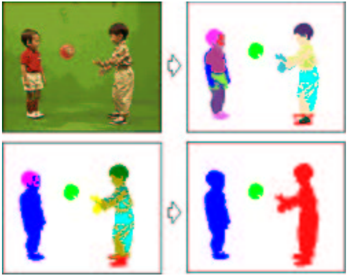
Topic

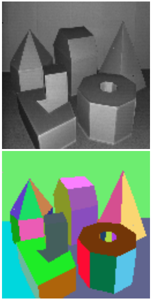
Introduction

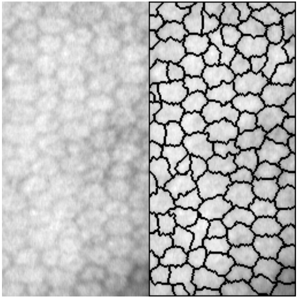
Individual pixel classification

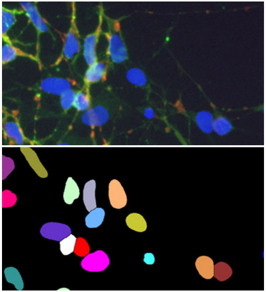
Region-based algorithms

What is image segmentation









Segmentation definition

- ... image dependent
- Image: $f : (\Omega \subseteq \mathbb{Z}^d) \rightarrow \mathbb{R}^m$,
  Segmentation: $s : (\Omega \subseteq \mathbb{Z}^d) \rightarrow \mathcal{Y}$ with $\mathcal{Y} = \{1, \dots L\}$
- Divide image pixels into L classes
- usually $L - 1$ objects and background
- usually objects are spatially compact (regularization)

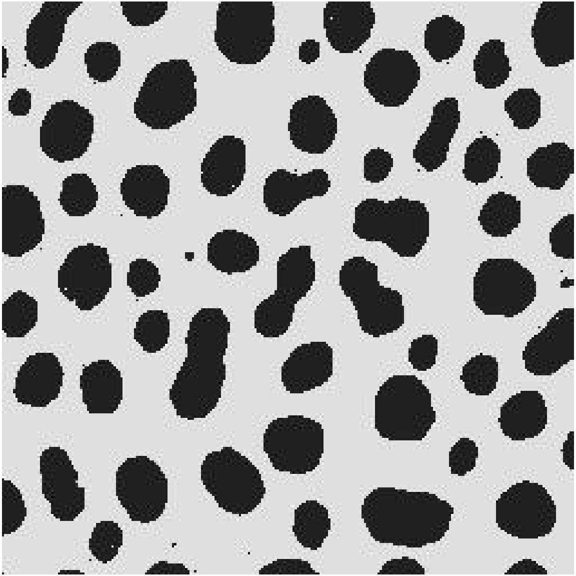
Segmentation approaches

- local information
  - intensity, colour, texture, ...
- global information
  - edges, shape, position, mutual position, ...
- complete × partial segmentation (parts not classified)
- a priori information (e.g. shape, position, relative position)
- multilevel segmentation (semantic feedback)

Segmentation difficulties

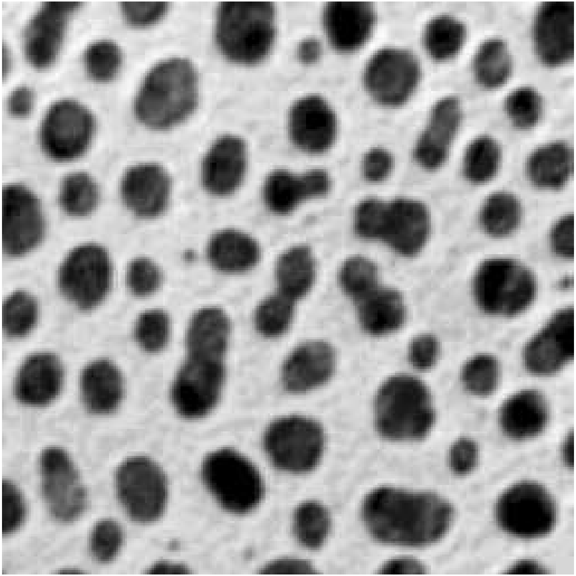
- Easy for humans $\times$ not easy for computers
- Humans cannot explain how they do it
- Image dependent; methods not universally applicable

Examples



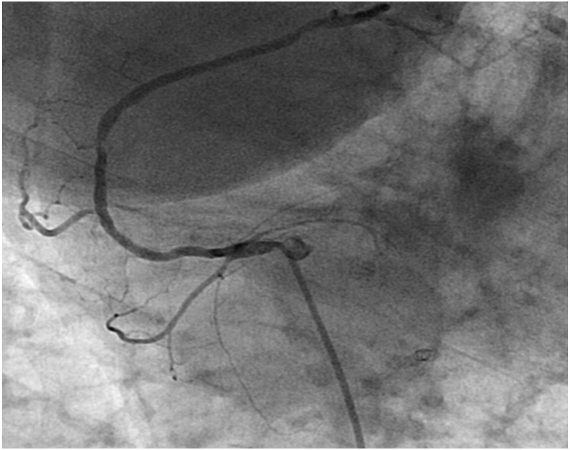
Cells are black

Examples



Cells are homogeneous connected regions

Examples



Arteries are dark and elongated, may have branches

Examples

Nose?

Taxonomy of segmentation methods

- individual pixel classification
- region-based methods — homogeneity, compactness
- edge-based methods
- active contours
- global methods (discrete optimization)

Topic

Individual pixel classification

- Calculate a descriptor vector for each pixel
  - colour, texture, wavelets, statistics…
- Classify each pixel based on the descriptor
  - linear classifier, SVM, AdaBoost …

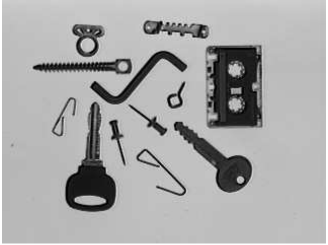
Thresholding

- For each pixel

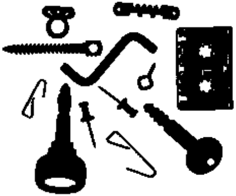
$$s(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad f(\mathbf{x}) \geq T \\ 0 & \text{if} \quad f(\mathbf{x}) < T \end{cases}$$

- Simple and frequently used technique
- Easy in hardware, intrinsically parallel.
- Threshold might be difficult to find
- Only works for some images

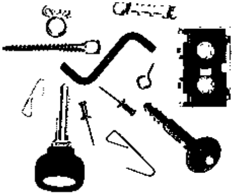
Dependence on threshold



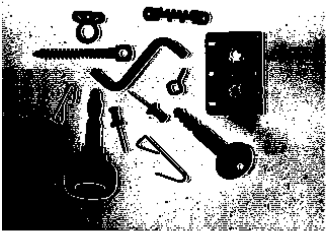
Original image.



Properly set threshold.



Threshold too low.



Threshold too high.

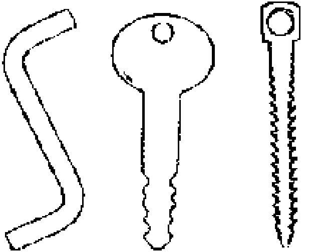
Thresholding, modifications

- Local adaptive threshold
- Band thresholding
- Multiple thresholds $\rightarrow$ multiple classes
- Thresholding to suppress background

$$s(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if} \quad f(\mathbf{x}) \geq T \\ 0 & \text{if} \quad f(\mathbf{x}) < T \end{cases}$$

Band thresholding example

Original image.

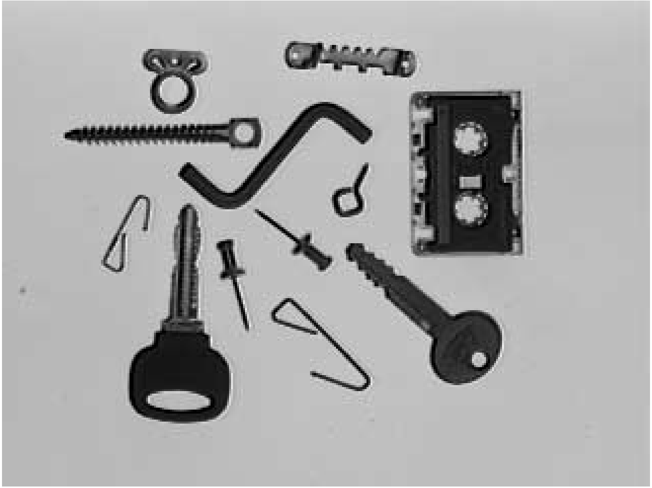


Border regions detected.

Threshold from a histogram

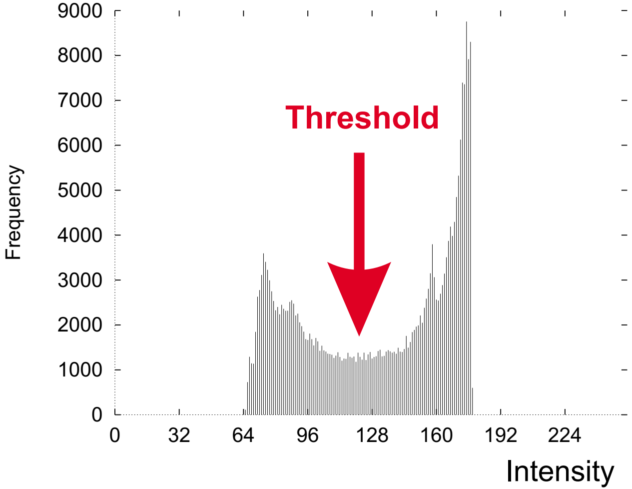
- Percentile method
  - object covering p percents $\rightarrow$ set T to p (or $1-p$) percentile
- Histogram shape analysis
  - put T between modes (maxima)

Histogram shape analysis



initial image

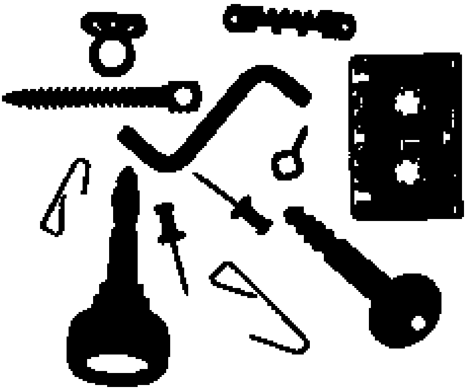
Histogram shape analysis



histogram

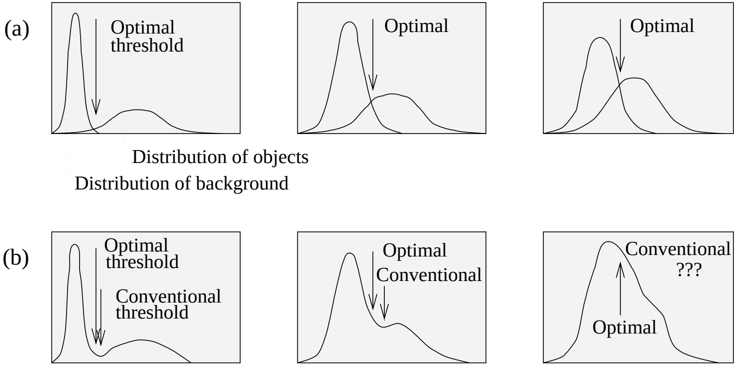
- histogram smoothing to avoid local minima — convolution

Histogram shape analysis



thresholded

Difficulty of finding an optimal threshold



Gaussian model

- Distribution of each class is Gaussian

$$p(f_i|s_i) = \frac{1}{\sigma_{s_i}\sqrt{2\pi}} \mathrm{e}^{-\frac{(f_i-\mu_{s_i})^2}{2\sigma_{s_i}^2}}$$

$$\text{with} \quad f_i = f(\mathbf{x}_i), \quad s_i = s(\mathbf{x}_i)$$

- All pixels are independent

$$p(F|S) = \prod_{i\in\Omega} p(f_i|s_i)$$

$$\text{with} \quad F = \big(f_i; i \in \Omega\big), \quad S = \big(s_i; i \in \Omega\big)$$

- Log-likelihood

$$\ell(F|S) = \log p(F|S) = \sum_{i\in\Omega} -\frac{(f_i-\mu_{s_i})^2}{2\sigma_{s_i}^2} - \log\sqrt{2\pi\sigma_{s_i}^2}$$

Foreground/background means algorithm (1)

- Same variances $\sigma_1 = \sigma_2$
- Maximum likelihood

$$
\begin{aligned}
(S^*, \mu_1^*, \mu_2^*) &= \arg\max_{S,\mu_1,\mu_2} p(F|S) \\
&= \arg\max_{S,\mu_1,\mu_2} \ell(F|S) \\
&= \arg\max_{S,\mu_1,\mu_2} -\sum_i \frac{(f_i - \mu_{s_i})^2}{2\sigma_{s_i}^2} - \log\sqrt{2\pi\sigma_{s_i}^2} \\
&= \arg\min_{S,\mu_1,\mu_2} \sum_i \left(f_i - \mu_{s_i}\right)^2
\end{aligned}
$$

- Solve by alternate minimizations

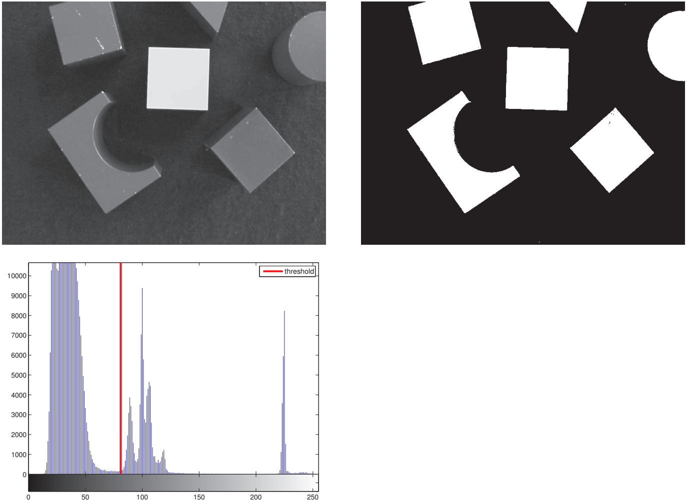
Foreground/background means algorithm (2)

1. Precalculate cumulative histograms
2. Estimate initial means μ_1, μ_2
3. Set threshold
$$T = \frac{\mu_1 + \mu_2}{2}$$
and calculate pixel labels s_i
4. Calculate new means μ_1, μ_2
5. Iterate 3–5 until convergence

- Very fast thanks to the cumulative histograms

Foreground/background means algorithm

example



k-Means

- Popular unsupervised learning algorithm
- ML estimation, equal variances

$$(S^*, \boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_L^*) = \arg \min_{S, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L} \sum_{i \in \Omega} \|\mathbf{f}_i - \boldsymbol{\mu}_{s_i}\|^2$$

- L classes , vector features $\mathbf{f}_i$

k-Means algorithm

1. Calculate class labels for all pixels

$$s_i = \arg\min_{j \in \mathcal{Y}} \|\mathbf{f}_i - \boldsymbol{\mu}_j\|^2$$

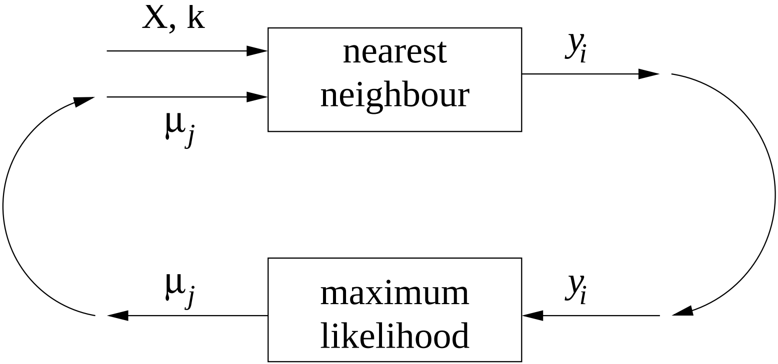
2. Calculate new means for all classes

$$\begin{aligned}\boldsymbol{\mu}_j &= \arg\min_{\boldsymbol{\mu}_j} \sum_{i \in \Omega} \|\mathbf{f}_i - \boldsymbol{\mu}_{s_i}\|^2 \\ &= \frac{1}{|\{i; s_i = j\}|} \sum_{s_i = j} \mathbf{f}_i\end{aligned}$$

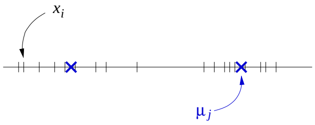
3. Repeat 1–2 until convergence

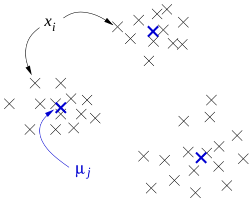
- Repeated random initializations

k-Means flowchart



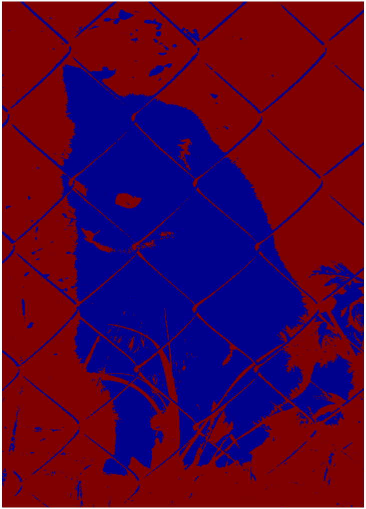
k-Means feature space





k-Means example

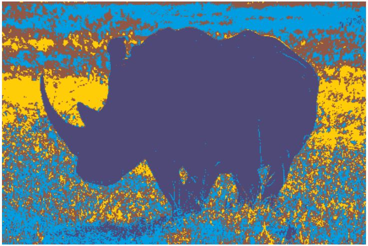
input



segmentation

k-Means example (2)

input



segmentation

Bayes classification (simplified)

- A priori class probabilities $P_s(1), P_s(2), \ldots, P_s(L)$
- Maximize a posteriori probability

$$S^* = \arg\max_S p(S|F)$$

- Independent pixels

$$s_i^* = \arg\max_{s_i} P(s_i|f_i)$$

- Bayes formula

$$P(s_i|f_i) = \frac{p(f_i|s_i)P_s(i)}{p(f_i)}$$

- Normalization factor $p(f_i)$
  - can be found by integration (marginalization)
  - sometimes not needed since independent of s_i

    #ML estimation of parameters

Two class Bayes decision

- Given P_S, μ_1, μ_2, σ_1, σ_2...
- Classify to class 1 iff

$$P(s_i = 1|f_i) > P(s_i = 2|f_i)$$
$$\frac{p(f_i|s_i = 1)P_s(1)}{p(f_i)} > \frac{p(f_i|s_i = 2)P_s(2)}{p(f_i)}$$
$$p(f_i|s_i = 1)P_s(1) > p(f_i|s_i = 2)P_s(2)$$

- For $P_s(1) = P_s(2)$ equivalent to the ML estimate

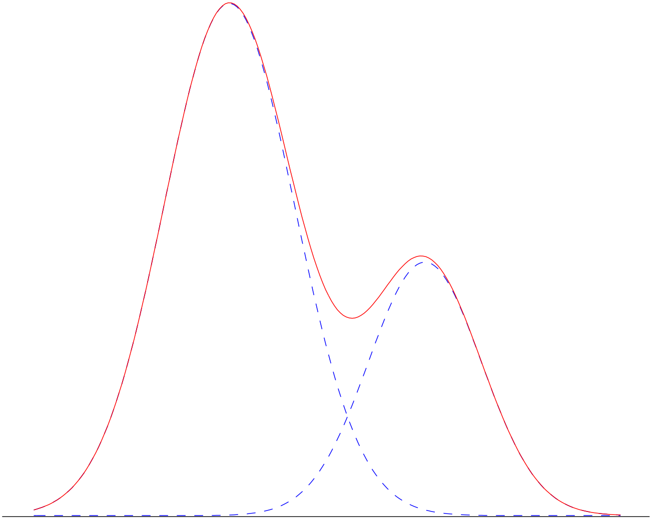
Gaussian mixture model (GMM)

- Probability of observing an intensity f

$$p(f) = \sum_{j=1}^{L} P_S(j) p(f|\mu_j, \sigma_j)$$

  - class unknown
  - weighted sum of Gaussians

Gaussian mixture model example



E-M algorithm (for GMM) — problem definition

- **Model:**
  - Independent pixels
  - Unknown classes $s_i \in \{1, \dots, L\}$
  - known intensities f_i

$$p(f_i|s_i) = \frac{1}{\sigma_{s_i}\sqrt{2\pi}} \mathrm{e}^{-\frac{(f_i - \mu_{s_i})^2}{2\sigma_{s_i}^2}}$$

- **Output:**
  - find parameters $\theta = \left\{\mu_1, \sigma_1, \dots, \mu_L, \sigma_L\right\}$
  - and probabilities $\alpha(i, j) = P(s_i = j)$

E-M algorithm

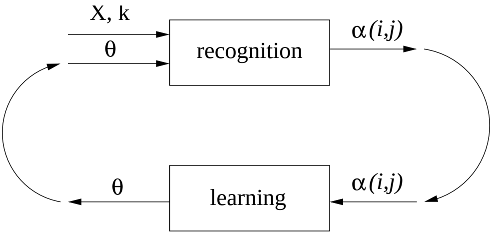
Alternatively maximize likelihood

$$p(F|\theta) = \prod_{i \in \Omega} p(f_i|\theta)$$

$$p(f_i|\theta) = \sum_j \underbrace{P(s_i = j)}_{\alpha(i,j)} p(f_i|s_i, \theta)$$

1. **E-step:** maximize with respect to α
2. **M-step:** maximize with respect to θ
3. Repeat 1–2 until convergence

E-M algorithm flowchart



- Similar to k-means with soft decisions
- A special case of a more general technique
- Can be extended to $\mathbb{R}^d$
- Likelihood increases monotonously
- Only local convergence guaranteed

E-M algorithm

explicit formulas

- **E-step:**

$$p(f_i|s_i = j) = \frac{1}{\sigma_{s_i}\sqrt{2\pi}} \mathrm{e}^{-\frac{(f_i - \mu_{s_i})^2}{2\sigma_{s_i}^2}}$$

$$P(s_i = j|f_i) = \alpha(i, j) = \frac{P_S(j)p(f_i|s_i = j)}{\sum_k P_S(k)p(f_i|s_i = k)}$$

$$P_S(j) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \alpha(i, j)$$

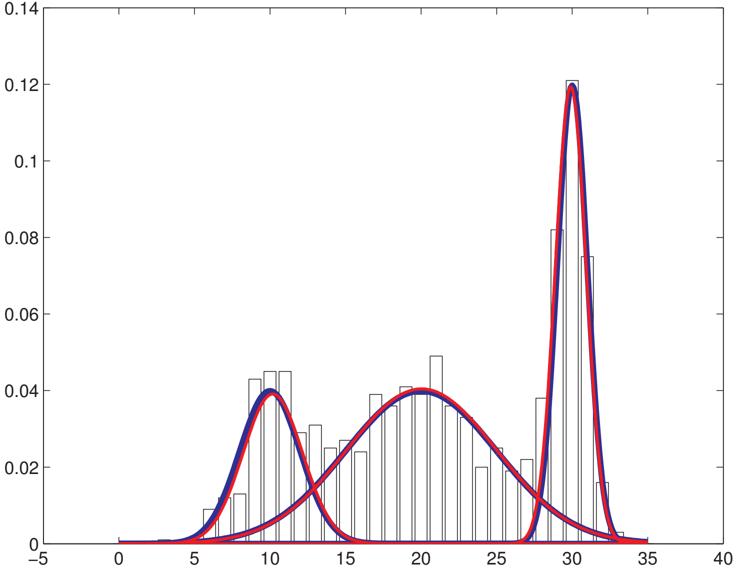
- **M-step:**

$$\mu_k = \frac{\sum_{i \in \Omega} \alpha(i, k) x_i}{\sum_{i \in \Omega} \alpha(i, k)}$$

$$\sigma_k^2 = \frac{\sum_{i \in \Omega} \alpha(i, k)(x_i - \mu_k)^2}{\sum_{i \in \Omega} \alpha(i, k)}$$

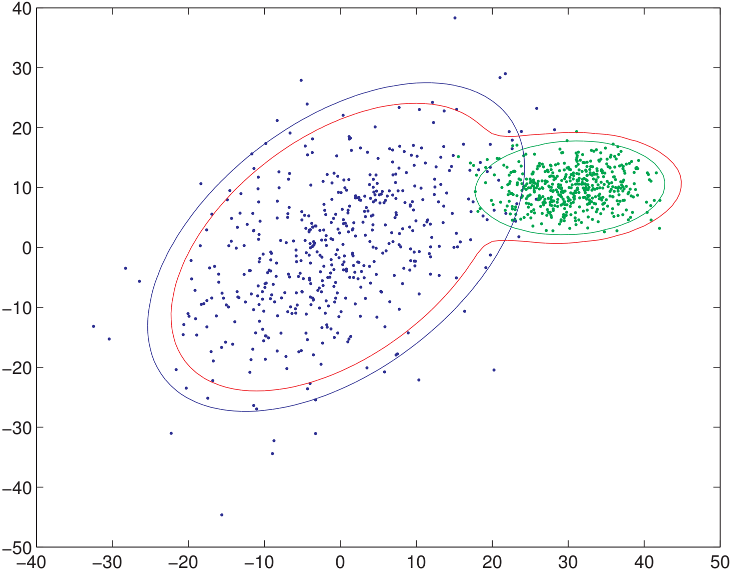
E-M algorithm

1D example



E-M algorithm

2D example

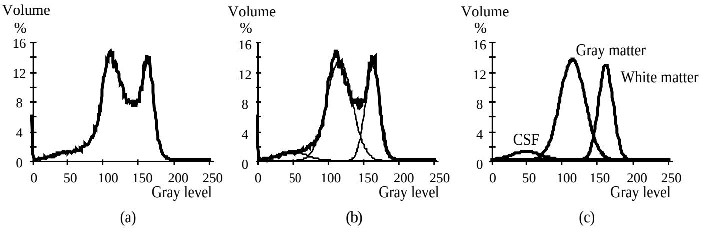


Confidence regions (95%) shown

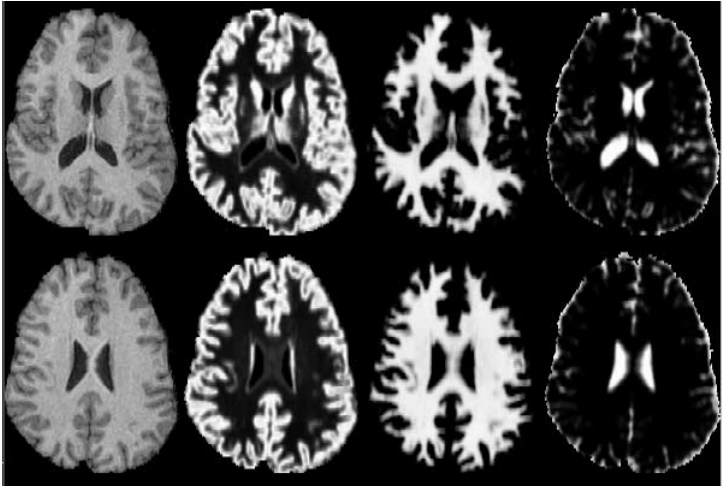
Brain MRI segmentation

Thresholding example

- **Input:** T1-weighted images
- **Desired classes:** white matter, grey matter, celebro-spinal fluid (CSF), background



Brain MRI segmentation results



Topic

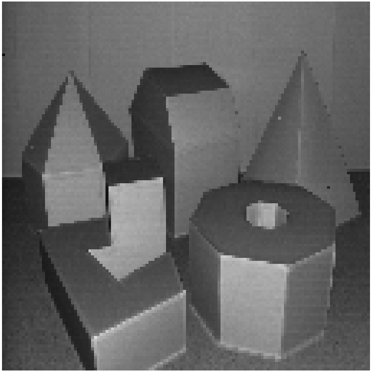
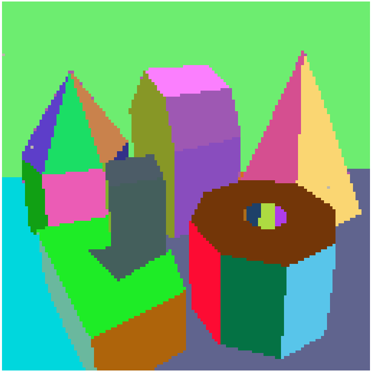
Introduction

Individual pixel classification

Region-based algorithms

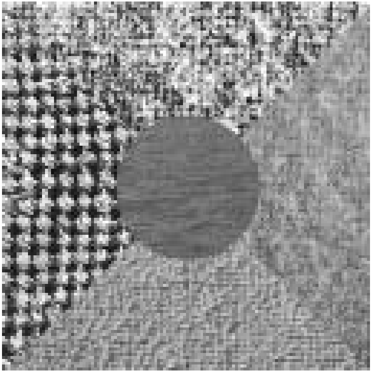
Region-based segmentation

- find homogeneous regions
- with respect to texture, intensity, color...

Region-based segmentation

- find homogeneous regions
- with respect to texture, intensity, color...



Region-based approaches

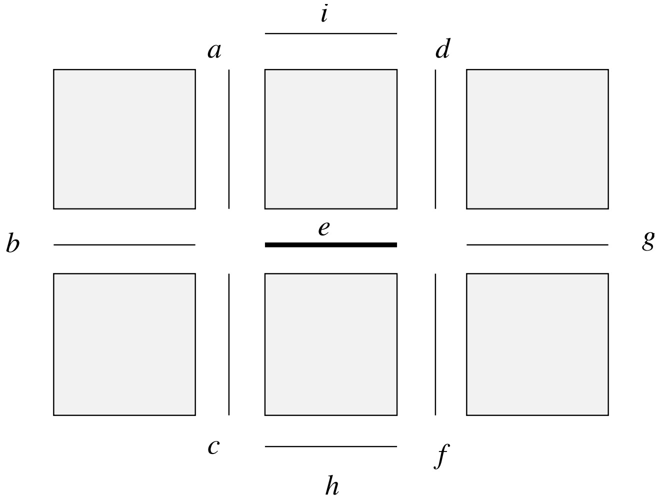
- region growing
- hierarchical image splitting
- watershed segmentation
- mean shift
- active contours (region based)
- graph-based algorithms
- ...

Region merging
edge relaxation

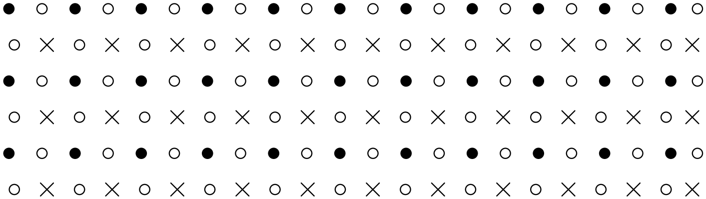
1. Initialization *(for speed)*
   - Neighboring pixels are *connected* if $|f_i - f_j| < T_0$
   - Connected components $\rightarrow$ initial regions
2. Weak edges: $|f_i - f_j| < T$
3. Merge regions recursively if $W \geq T_2 \min(l_1, l_2)$
4. Merge regions recursively if $W \geq T_3 l$

- Maintain list of regions and their boundaries

Region merging — structures

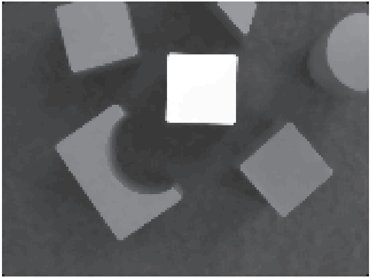


Region merging — structures



×,image data; ○, edges; •, unused.

Region merging — example



Removal of small regions

1. Find the smallest region R. If it is big enough $\rightarrow$ finish.
2. Consider neighbors of R and merge with the most similar one.
3. Repeat 1–2.

Removal of small regions — example

15 regions

7 regions

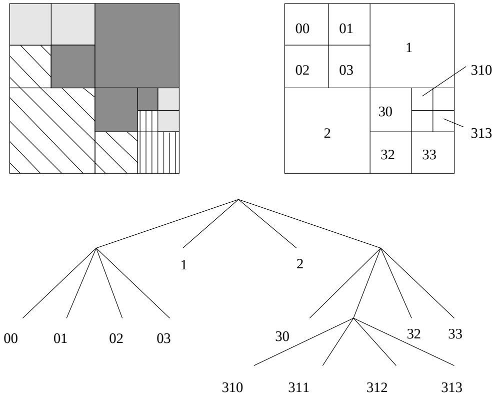
Region splitting

1. Image = region
2. Split each inhomogeneous region
3. Repeat recursively

-
    - Fast ($O(n \log n)$)
    - Unnatural region boundaries

Oct-tree splitting

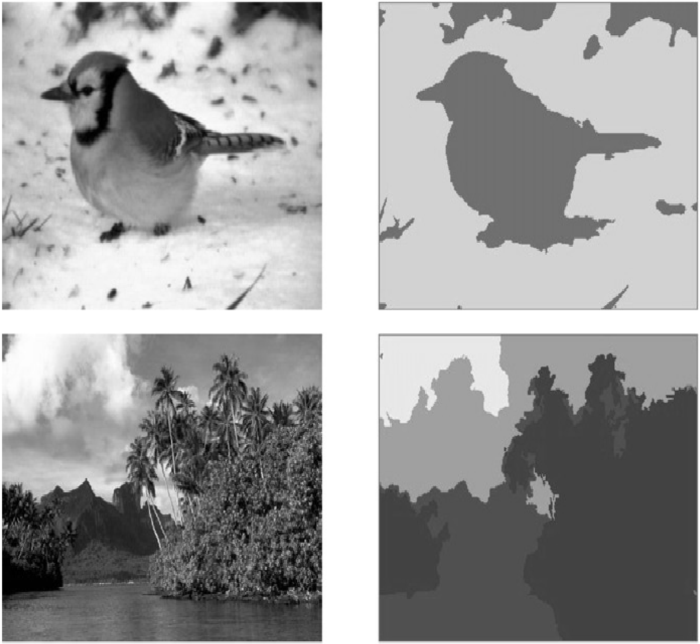
- Split each inhomogeneous square into four equal ones



Split and merge

1. Region splitting
2. Region merging

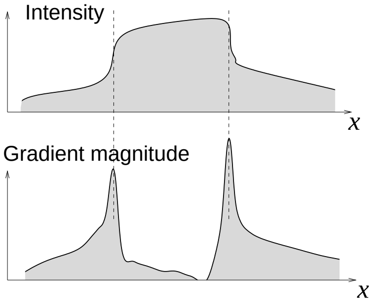
Split and merge example

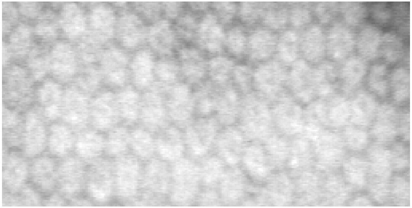


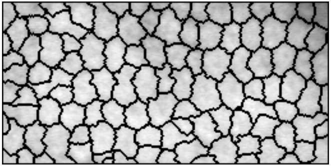
Watershed segmentation

- Interpret image in 3D
  - or gradient of the image, or distance function ...
- 'flood' the image to height T
- each 'drop' falls into a 'catchment basin'
- catchment basins define the segmentation
- basins meet as water rises $\rightarrow$ hierarchical segmentation

Watershed segmentation — example

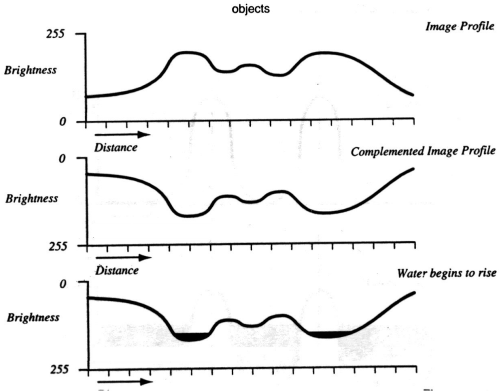






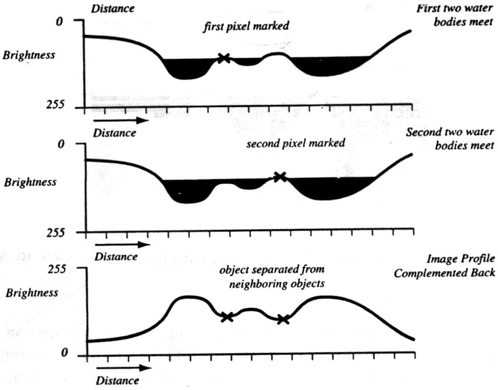
Watershed algorithm

- Pump-in water from local minima, build dams when basins meet



Watershed algorithm

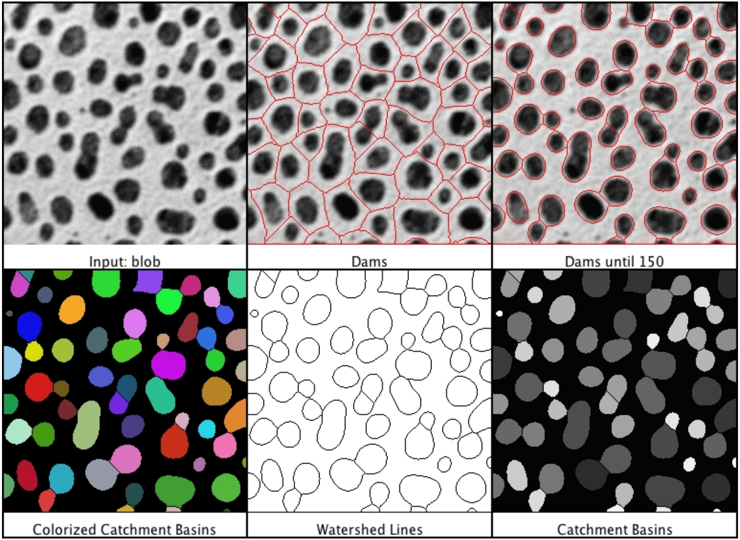
- Pump-in water from local minima, build dams when basins meet



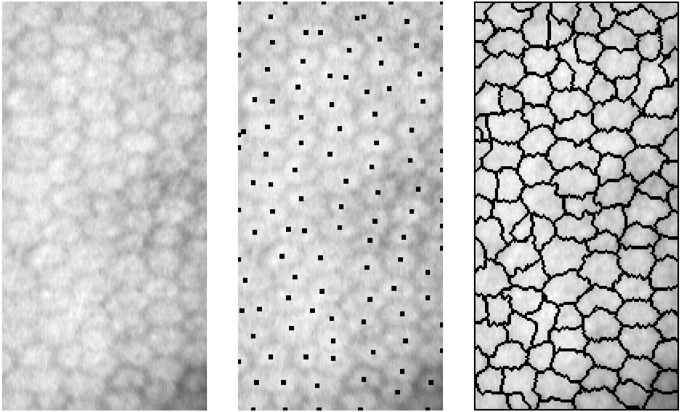
Watershed algorithm (2)

1. Insert local minima into a priority queue sorted by intensity
2. Repeatedly extract pixels from the queue
   - If adjacent to one region, add it to that region.
   - If adjacent to more regions, it is a boundary
   - Merge regions
   - Unprocessed neighbors are added to the queue.

Watershed algorithm — example



Watershed with markers



original markers cell boundaries

Mean shift

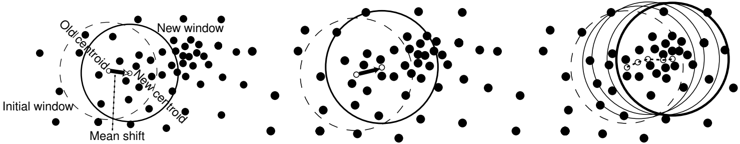
- Given samples
- Find global maxima of a probability density
- In segmentation - find regions corresponding to each maximum

Mean shift segmentation overview

- No assumptions about probability distributions — rarely known
- Spatial-range domain $\big(x, y, f(x, y)\big)$ — normally $f(x, y)$
- Find maxima in the $\big(x, y, f\big)$ space — clusters close in space and range correspond to classes.

Mean shift procedure

Goal: Find local maxima of the probability density *(density modes)* given by samples.



1. Start with a random region of interest.
2. Determine a centroid of the data.
3. Move the region to the location of the new centroid.
4. Repeat until convergence.

Kernel estimation

$$K(\mathbf{x}) = c_k k\left(\|\mathbf{x}\|^2\right) \qquad \text{(radial symmetry)}$$

Epanechnikov kernel (other choices possible)

$$k(r) = \begin{cases} 1 - r & \text{for } r \leq 1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{(profile, } r = \|\mathbf{x}\|^2\text{)}$$

Kernel density estimator

$$\widetilde{f}(\mathbf{x}) = \frac{1}{n\,h^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

Mean shift procedure

At density maxima $\nabla \widetilde{f} = 0$

$$\widetilde{f}(\mathbf{x}) = \frac{1}{n\,h^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

$$\nabla \widetilde{f}(\mathbf{x}) = \frac{2\,c_k}{n\,h^{(d+2)}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i)\, k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$$

$$= \frac{2\,c_k}{n\,h^{(d+2)}} \left(\sum_{i=1}^{n} g_i\right) \left(\mathbf{x} - \frac{\sum_{i=1}^{n} \mathbf{x}_i\, g_i}{\sum_{i=1}^{n} g_i}\right)$$

for $g(r) = -k'(r) > 0$, $g_i = g\big(\,\|(\mathbf{x} - \mathbf{x}_i)/h\|^2\,\big)$

Mean shift procedure

At density maxima $\nabla \widetilde{f} = 0$

$$\widetilde{f}(\mathbf{x}) = \frac{1}{n\,h^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

$$0 = \nabla \widetilde{f}(\mathbf{x}) = \frac{2\,c_k}{n\,h^{(d+2)}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i)\, k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)$$

$$= \frac{2\,c_k}{n\,h^{(d+2)}} \left(\sum_{i=1}^{n} g_i\right) \underbrace{\left(\mathbf{x} - \frac{\sum_{i=1}^{n} \mathbf{x}_i\, g_i}{\sum_{i=1}^{n} g_i}\right)}_{\textit{mean shift vector} \text{ — must be 0 at optimum}}$$

for $g(r) = -k'(r) > 0,\ g_i = g\big(\,\|(\mathbf{x} - \mathbf{x}_i)/h\|^2\,\big)$

Mean shift procedure (2)

Mean shift vector

$$m(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i \, g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}$$

$$g_i = g\big(\|(\mathbf{x} - \mathbf{x}_i)/h\|^2 \big)$$

$$g(r) = -k'(r)$$

Successive locations $\mathbf{y}_j$ of the kernel:

$$\mathbf{y}_{j+1} = \sum_{i=1}^{n} \mathbf{x}_i \, g\left(\left\| \frac{\mathbf{y}_j - \mathbf{x}_i}{h} \right\|^2 \right) \Bigg/ \sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{y}_j - \mathbf{x}_i}{h} \right\|^2 \right)$$

Mean shift procedure (2)

Mean shift vector

$$m(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i \, g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}$$

$$g_i = g\left(\|(\mathbf{x} - \mathbf{x}_i)/h\|^2 \right)$$

$$g(r) = -k'(r)$$

Successive locations $\mathbf{y}_j$ of the kernel:

$$\mathbf{y}_{j+1} = \sum_{i=1}^{n} \mathbf{x}_i \, g\left(\left\| \frac{\mathbf{y}_j - \mathbf{x}_i}{h} \right\|^2 \right) \Bigg/ \sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{y}_j - \mathbf{x}_i}{h} \right\|^2 \right)$$

Theorem: If k is convex and monotonically decreasing, the sequence $\{y_j\}_{j=1,2,\ldots}$ converges and $\{\widetilde{f}(\mathbf{y}_j)\}_{j=1,2,\ldots}$ increases monotonically.

For Epanechnikov kernel → convergence in finite number of steps.

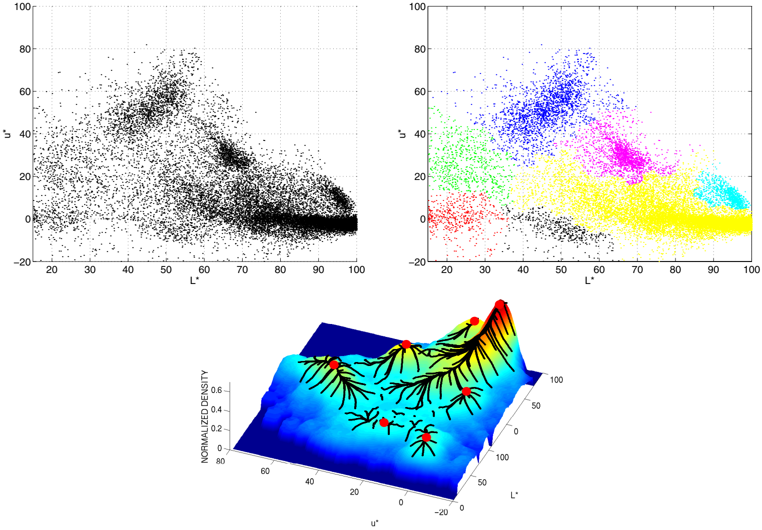
Mean shift mode detection

Points from a *basin of attraction* converge to the same mode.

Algorithm:

1. Using multiple initializations covering the entire feature space, identify modes (stationary points).
2. Using small random perturbation, retain only local maxima.

Mean shift mode detection example



Mean shift discontinuity preserving filtering

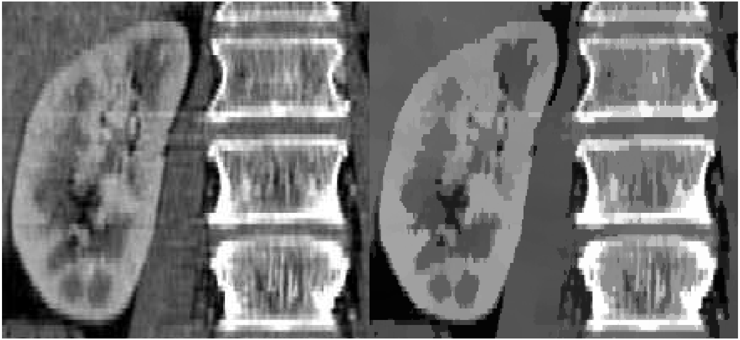
Combine spatial and range values

$$K(\mathbf{x}) = ([\mathbf{x}^s \; \mathbf{x}^r]) = \frac{c}{h_s^d \, h_r^p} \, k\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|^2\right) ,$$

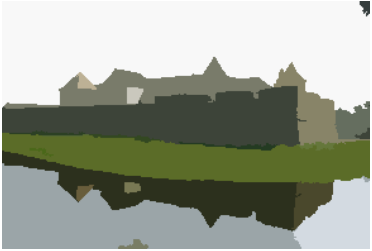
Algorithm:

1. For each image pixel $\mathbf{x}_i$, initialize $\mathbf{y}_{i,1} = \mathbf{x}_i$.
2. Iterate the mean shift procedure until convergence.
3. The filtered pixel values are defined as $\mathbf{z}_i = (\mathbf{x}_i^s, \mathbf{y}_{i,\text{con}}^r)$; the value of the filtered pixel at the location $\mathbf{x}_i^s$ is assigned the image value of the pixel of convergence $\mathbf{y}_{i,\infty}^r$.

Mean shift discontinuity preserving filtering

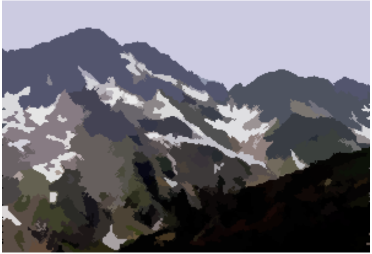


Mean shift filtering examples



Mean shift filtering examples

Mean shift filtering examples

Mean shift segmentation

1. Mean shift discontinuity preserving filtering
2. Determine the clusters $\{C_p\}_{p=1,\dots,m}$ by grouping all $\mathbf{z}_i$, which are closer than h_s in the spatial domain and h_r in the range domain, i.e. merge the basins of attractions.
3. Assign class labels to clusters
4. If desired, eliminate regions smaller than P pixels.

Mean shift segmentation examples

Mean shift segmentation examples