Vision and Image Processing: Fundamentals of Grouping and Segmentation

François Lauze

Department of Computer Science
University of Copenhagen

January 10. & 12., 2022

Plan for today and Monday

- A general introduction of segmentation, Gestalt Theory.
- Edge detection: Marr-Hildreth, Canny.
- Edge based segmentation: Closing gaps with Snakes.
- Watershed segmentation.
- Segmentation and Clustering.
- K-means.
- Notion of spatial regularization.
- Mean Shift.
- A few words on Deep Learning Based Segmentation.

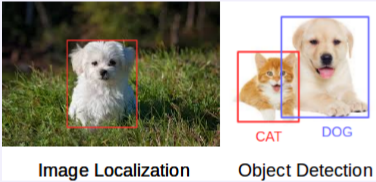
Outline

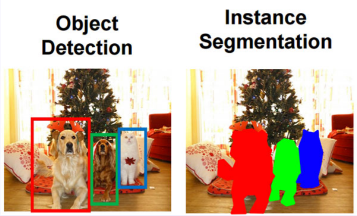
Image Segmentation

- An intelligible image is not formed of random pixels. There must fundamental consistencies between them.
- Image segmentation is the process of dividing an image into coherent regions of similarity, called segments, by grouping similar pixels:
  - Region: a group of connected pixels that share some common properties.
  - Property: intensity, colour, texture, motion (for sequences), boundary (edges)
  - Some of these properties can be defined in a pixel-wise manner: intensity, colour for instance. Some are related to pixel neighbourhoods, texture for instance.
  - Segments may be grouped together into complex objects.

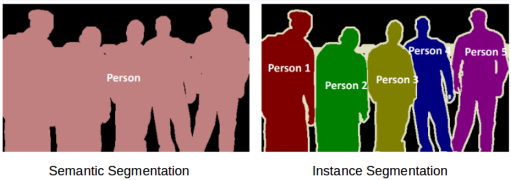
Localisation, Object Detection and Segmentation

Images from `analyticsvidhya.com`



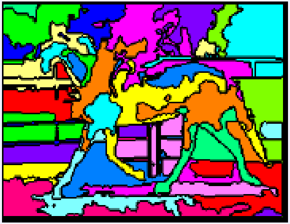


Images from `cs231n.stanford.edu`

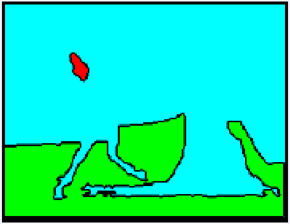


Actually very "high-level" type of segmentation using some form of cognition.

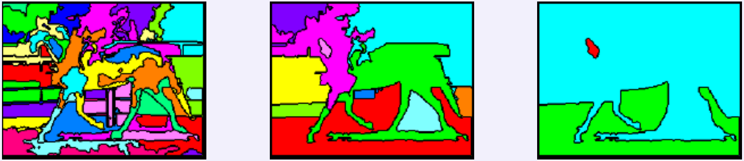
Levels of Segmentation[1]



oversegmented



undersegmented

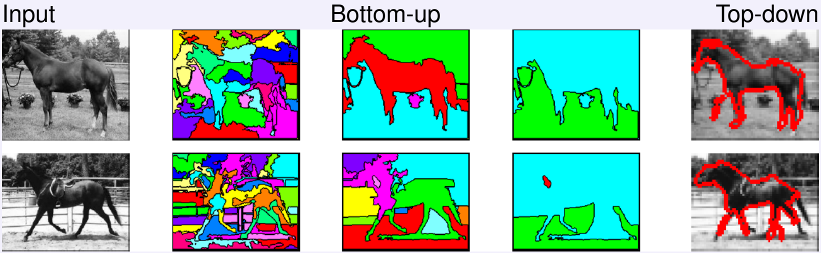


Multiple segmentations

[1] Slide adapted from Derek Hoiem

Major Segmentation types[2]

- Bottom up: grouping token with similar features
- Top-down: group tokens that likely belong to the same object



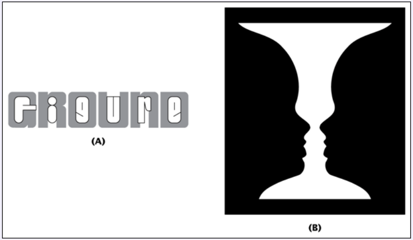
[2]Slide adapted from Hoiem, and paper from Levin-Weiss 2006

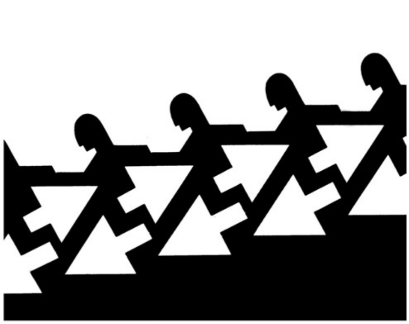
A bit of Gestalt Theory: Grouping

- Figure–Ground organization.
  - Perceptual apparatus picks out some objects to be figures, while other are less relevant in the background.
- Grouping
  - Inherent properties from the stimulus environment lead people to group them together.
  - Grouping principles
    - Proximity: group nearby figures together.
    - Similarity: group figures that are similar.
    - Continuity: perceive continuous patterns
    - Closure: fill-in gaps
    - Connectedness: spots, lines and areas are seen as unit when connected.
    - Synchronicity: occur at the same time
    - Common region: located within some boundary
    - Connectedness: connected by other elements.

Slide adapted from “Sensation and Perception” at `http://college.cengage.com`

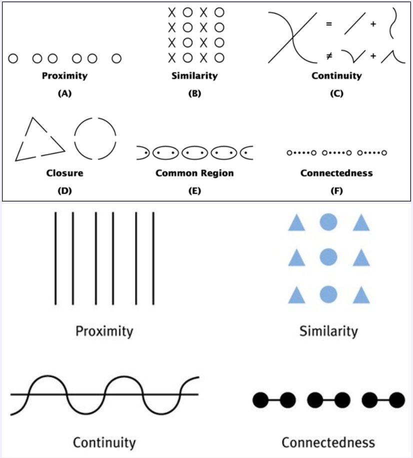
Figure/Ground





Slide adapted from “Sensation and Perception” at `http://college.cengage.com`

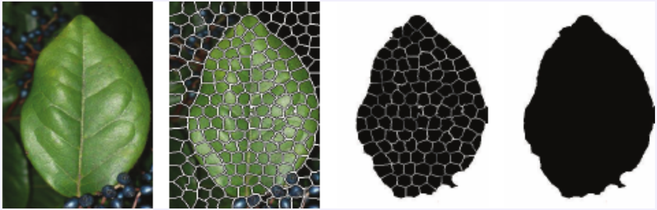
Grouping



Slide adapted from “Sensation and Perception” at `http://college.cengage.com`

Operationalization

- Similarity and Common Regions principles. Detect coherent regions, a.k.a segments. Group them further.



- Two different segments should have dissimilar properties. In particular, boundary between segments should present large variations.
- Some standard approaches:
  - Region based segmentation (generally bottom-up).
  - Edge / contour based segmentations (generally top-down).

Regions and Edges

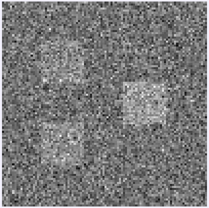
- A region should be bounded by a closed contour: edge detection.
- Regions may be obtained by “boundary filling”.

- However edges are not always well defined in observed images.

Regions and Edges

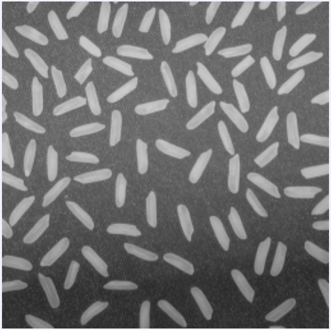
- A region should be bounded by a closed contour: edge detection.
- Regions may be obtained by "boundary filling".

- However edges are not always well defined in observed images.



Regions

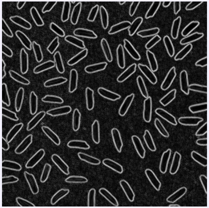
- Regions correspond generally to objects or pieces of objects in a scene.
- Scenes may contain several objects.



Outline

Edge Based Segmentation

- Boundary between region corresponds to sharp intensity / colour / texture change.
- Derivatives should indicate this:



- Edges should correspond to local maximum of gradient magnitude.
- Gradient of f : $\nabla f = (f_x, f_y)^T$, Gradient magnitude = $\sqrt{f_x^2 + f_y^2}$.
- Zero-crossing of Laplacian $\Delta f = \nabla^2 f := f_{xx} + f_{yy}$ is also an indicator.
- Canny edge detector.
- others

Scale, Noise, Derivatives and Gaussian –

- Convolution by Gaussian smooths: noise removal. Standard deviation of Gaussian directly linked to scale of features.
- Derivative of Gaussian:

$$g_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad \frac{\partial g_\sigma}{\partial x} = -\frac{x}{\sigma^2} g_\sigma, \quad \frac{\partial g_\sigma}{\partial y} = -\frac{y}{\sigma^2} g_\sigma$$

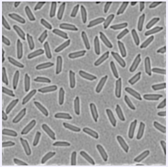
- Laplacian of Gaussian (LoG)

$$\Delta g_\sigma = \frac{\partial^2 g_\sigma}{\partial x^2} + \frac{\partial^2 g_\sigma}{\partial y^2} = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} g_\sigma$$

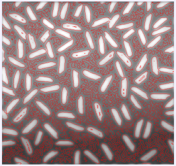
Marr-Hildreth Edge Detector

D. Marr and E. Hildreth, 1980.

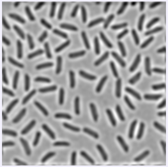
- Observe zero-crossing of Laplacian of Gaussian (the same use for SIFT for instance)
- Convolve image f with Laplacian of Gaussian filter: $\Delta_\sigma f = \Delta g_\sigma * f$
- Detect the zero-crossings of $\Delta_\sigma f$



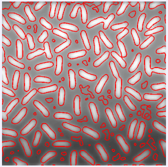
LoG $\sigma = 1.4$



detected edges

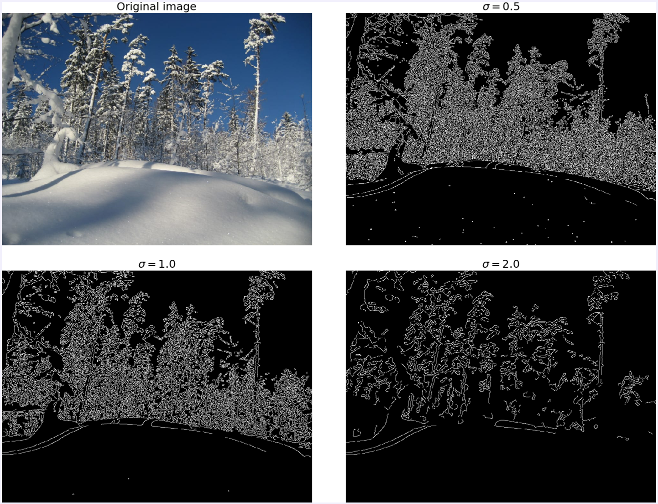


LoG $\sigma = 3.0$



detected edges

Canny Example

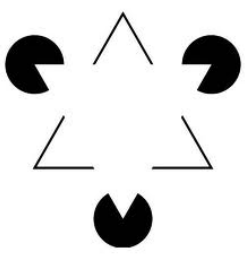


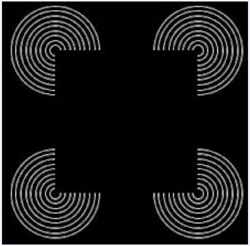
- Edges are not closed: good but might be insufficient for segmentation.

Outline

Contours and Perception: Amodal Completion

A Classical example from Italian psychologist G. Kanisza[3]



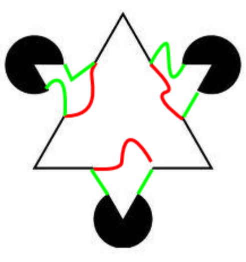


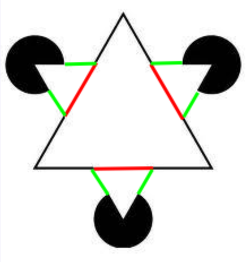
The brain is “closing the gaps” – Gestalt’s closure principle.

[3] Grammatica del vedere / Organization in Vision

But how...

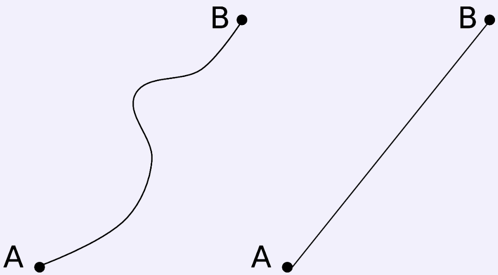
What is the best continuation?





- We reconstruct the gap with an implicit assumption of simplicity / regularity for the contour.
- Variational segmentation algorithms operationalize it.

Simplicity / Regularity



- The first curve is longer and winds more than the second one.
- The second one is simpler.

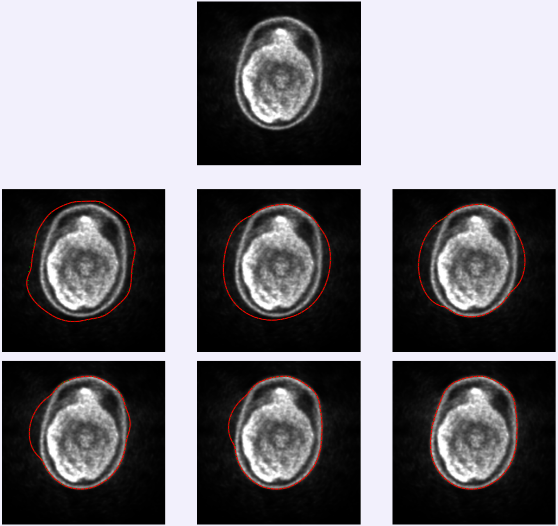
Snakes, Kass, Witkin, Terzopoulos 1988

- Curves evolving (as snakes) to catch an object. Also called Active Contours.
- How to define the"Cost of a Segmentation":
- Cost of a curve lying on an image such that
  - The more a curve follows edges of the image, the "cheaper" it is.
  - The more a curve is bending / winding, the "more expensive" it is.
  - Design an algorithm to compute the "minimal cost", or a cost low enough.
  - Kass, Witkin. Terzopoulos: minimize

$$\mathcal{E}(C) = \int \alpha|\dot{C}|^2 + \beta|\ddot{C}|^2\, dt + \int F_{\text{edge}}(C(t))\, dt$$

  - Solution via Partial Differential Equation (PDE).
  - Start with initial contour. Evolve it so as to decrease cost as fast as possible (steepest descent).
  - Thousands of derived methods!

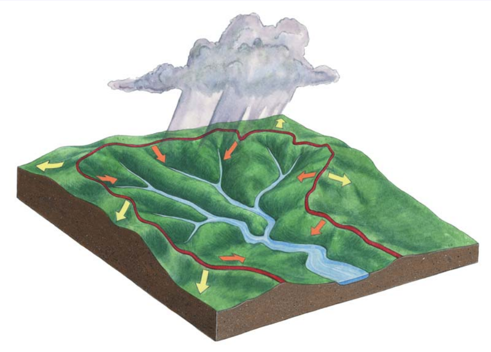
Active Contour Example



Outline

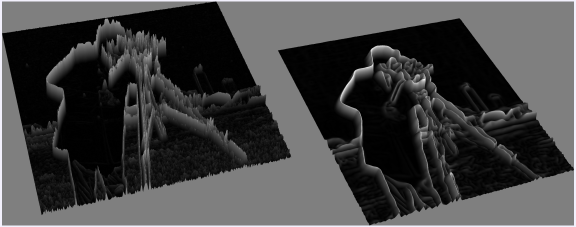
Watersheds

- Watersheds = Water separation lines



Images and Topographic maps

- Use Gaussian gradient magnitude image to obtain “topographic” image

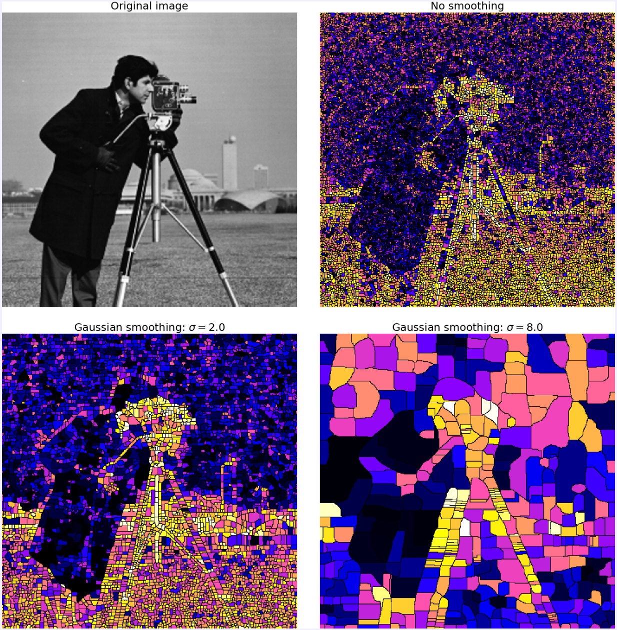


$\sigma = 2.0$ $\sigma = 4.0$

Watershed Algorithm

- Find ridges by “flooding”:
- Start at local minima – lowest catchment basin points
- flood: mark neighbouring points as belonging to local minima if not already marked.
- Progress by processing each times non marked lowest points.
- when a point belongs / at boundary of two 2 catchment basins: mark as watershed.

Example

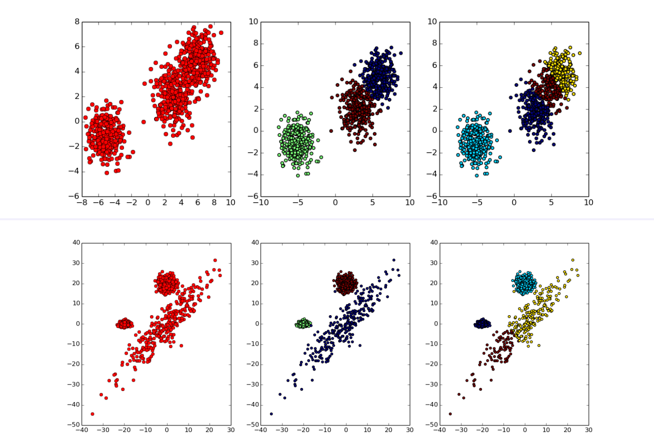

Original image
No smoothing
Gaussian smoothing: $\sigma = 2.0$
Gaussian smoothing: $\sigma = 8.0$

Outline

Clusters of Features

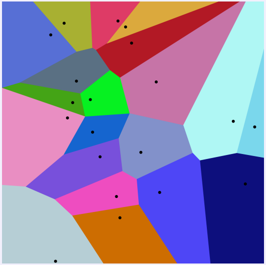
- Idea: extract features for each pixel of the image.
- Group features by similarity.
- Regions with similar features = segments.
- Need to define what features are.
- Need to define what feature similarity is.
- Automatic determination of clusters? Choice of the number of clusters?

Clustering In General



Up: k-means clustering, with $k = 3, 4$. (3 = optimal) Down: Gaussian Mixture models with 3 and 4 (3 optimal too).

The k-Means Algorithm



- Plan divided into cells.
- Proximity to centroid.
- k-means: compute centroids, implicit Voronoï tessellation, from data.
- Remember CBIR lecture before Xmas!

- $X = \{x_1, \ldots, x_n\}$ set of data points. Find a partition $S_1, \ldots, S_k$ of X and points $\mu_1, \ldots, \mu_k$ such that

$$\mathcal{D}(S_1 \ldots, S_k) = \sum_{j=1}^{k} \sum_{x \in S_j} \|x - \mu_j\|^2 = \text{ minimum.}$$

- The μ_js are the centroids of the S_js.

Lloyd's Algorithm

- Start with k candidate centroids. How to choose them?
- Repeat:
  1. Label pixels / assign them to clusters from their feature distance to centroids.
  2. Recompute centroids of features as average of clusters
- until regions or centroids don't change.

- What are good features for segmentation?
- Depends on image content.
  - Intensity very often used – **and is the one that has to be used in the assignment!**
  - But intensity is local and noisy. Gaussian features can be a good guess.
  - Some image modalities have more complicated image values. Synthetic Aperture RADAR (SAR), Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI) etc.

Gaussian Scale Space Features

- 4 scales, $\sigma = 0.5, 1.0, 2.0, 4.0$
- Gaussian derivatives from order 0 (no derivative) to order 3

$$\frac{\partial^0}{\partial x^0 \partial y^0}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial x \partial y}, \frac{\partial^3}{\partial x^3}, \frac{\partial^3}{\partial y^3}, \frac{\partial^3}{\partial x^2 \partial y}, \frac{\partial^3}{\partial x \partial y^2}$$

- In all, 40 features per pixel!

Variation

- Same features, but without order 0 (intensity)

- Conclusion: a lot of information in intensity alone!

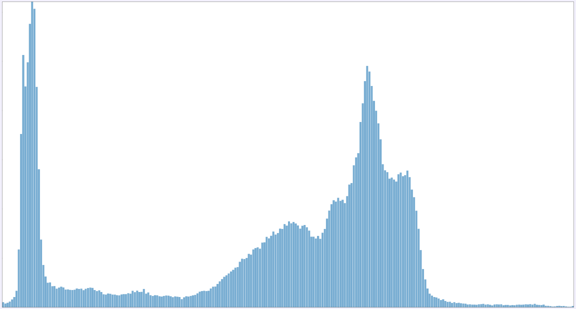
Variation

- Same features, but without order 0 (intensity)

- Conclusion: a lot of information in intensity alone!

Choice of k

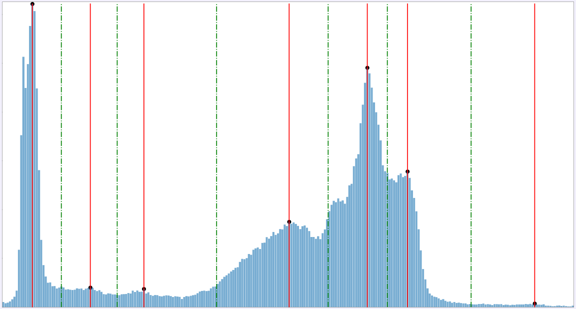
- Previous image clearly over segmented.
- Histogram of Cameraman Image $\pm$ 7 “bumps”:



- For 1D features: k-means tries to separate the bumps in the histogram!
- What if k is far from the number of bumps,
- if bumps are unclear?
- if bumps are very close?
- What makes a bump? ML course?

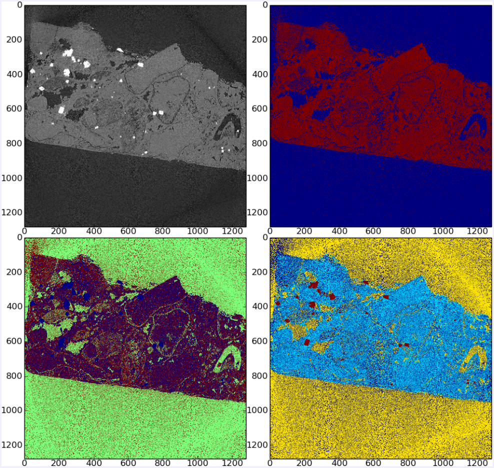
Choice of k

- Previous image clearly over segmented.
- Histogram of Cameraman Image $\pm$ 7 “bumps”:



- For 1D features: k-means tries to separate the bumps in the histogram!
- What if k is far from the number of bumps,
- if bumps are unclear?
- if bumps are very close?
- What makes a bump? ML course?

Example



Original image and *k*-means clustering for *k*=2,3,4.

Otsu Clustering Method – 1979

- For 2 regions:
  - Find threshold such that intraclass variance is minimized
    $$\sigma_w(t)^2 := \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$
  - $\omega_1(t) = \frac{\#\text{pixels} < t}{\#\text{pixels in image}}$ = probability that pixel value is $< t$,
    $\omega_2(t) = \frac{\#\text{pixels} \geq t}{\#\text{pixels in image}}$ = probability that pixel value is $\geq t$.
  - $\sigma_1^2(t) =$ variance of the class of pixels $< t$,
    $\sigma_2^2(t) =$ variance of the class of pixels $\geq t$,
  - Search for t in range $[\min(\text{image value}), \max(\text{image value})]$.

Means, Variances, classes

- For a greyscale image I:

Means, Variances, classes

- For a greyscale image I:
- Total mean μ_T and total variance σ_T^2 defined via histogram of intensities

$$h(i) = \text{number of pixels of } I \text{ with intensity } i,$$

Means, Variances, classes

- For a greyscale image I:
- Total mean μ_T and total variance σ_T^2 defined via histogram of intensities

$$h(i) = \text{number of pixels of } I \text{ with intensity } i,$$

- $S = \sum_{i=i_{\min}}^{i_{\max}} h(i)$ number of pixels of I, $\frac{h(i)}{S} := p(i)$ frequency / probability of intensity value i in image I.

$$\mu_T = \frac{\sum_{i=i_{\min}}^{i_{\max}} ih(i)}{S} = \sum_{i=i_{\min}}^{i_{\max}} ip(i)$$

$$\sigma^2 = \frac{\sum_{i=i_{\min}}^{i_{\max}} h(i)(i-\mu_T)^2}{S} = \sum_{i=i_{\min}}^{i_{\max}} p(i)(i-\mu_T)^2$$

Means, Variances, classes

- For a greyscale image I:
- Total mean μ_T and total variance σ_T^2 defined via histogram of intensities

$$h(i) = \text{number of pixels of } I \text{ with intensity } i,$$

- $S = \sum_{i=i_{\min}}^{i_{\max}} h(i)$ number of pixels of I, $\frac{h(i)}{S} := p(i)$ frequency / probability of intensity value i in image I.

$$\mu_T = \frac{\sum_{i=i_{\min}}^{i_{\max}} ih(i)}{S} = \sum_{i=i_{\min}}^{i_{\max}} ip(i)$$

$$\sigma^2 = \frac{\sum_{i=i_{\min}}^{i_{\max}} h(i)(i-\mu_T)^2}{S} = \sum_{i=i_{\min}}^{i_{\max}} p(i)(i-\mu_T)^2$$

- Probability of having pixel value $\leq t$ (class 1) and $> t$ (class 2)

$$\omega_1(t) = \sum_{i=i_{\min}}^{t} p(i) \quad \omega_2(t) = \sum_{t+1}^{i_{\max}} p(i)$$

Intra- and inter-class variances

- Mean of class 1 and class 2

$$\mu_1(t) = \frac{\sum_{i=i_{\min}}^{t} ip(i)}{\omega_1(t)}, \quad \mu_2(t) = \frac{\sum_{t+1}^{i=i_{\max}} ip(i)}{\omega_2(t)}$$

Intra- and inter-class variances

- Mean of class 1 and class 2

$$\mu_1(t) = \frac{\sum_{i=i_{\min}}^{t} ip(i)}{\omega_1(t)}, \quad \mu_2(t) = \frac{\sum_{t+1}^{i=i_{\max}} ip(i)}{\omega_2(t)}$$

- Variances of class 1 and class 2

$$\sigma_1^2(t) = \frac{\sum_{i=i_{\min}}^{t} p(i)(i-\mu_1(t))^2}{\omega_1(t)}, \quad \sigma_2^2(t) = \frac{\sum_{t+1}^{i_{\max}} p(i)(i-\mu_2(t))^2}{\omega_2(t)}$$

Intra- and inter-class variances

- Mean of class 1 and class 2

$$\mu_1(t) = \frac{\sum_{i=i_{\min}}^{t} ip(i)}{\omega_1(t)}, \quad \mu_2(t) = \frac{\sum_{t+1}^{i=i_{\max}} ip(i)}{\omega_2(t)}$$

- Variances of class 1 and class 2

$$\sigma_1^2(t) = \frac{\sum_{i=i_{\min}}^{t} p(i)(i-\mu_1(t))^2}{\omega_1(t)}, \quad \sigma_2^2(t) = \frac{\sum_{t+1}^{i_{\max}} p(i)(i-\mu_2(t))^2}{\omega_2(t)}$$

- Intra-class variance (or *within class variance*)

$$\sigma_w^2(t) = \omega_1(t)\sigma_1(t)^2 + \omega_2(t)\sigma_2(t)^2$$

Intra- and inter-class variances

- Mean of class 1 and class 2

$$\mu_1(t) = \frac{\sum_{i=i_{\min}}^{t} ip(i)}{\omega_1(t)}, \quad \mu_2(t) = \frac{\sum_{t+1}^{i=i_{\max}} ip(i)}{\omega_2(t)}$$

- Variances of class 1 and class 2

$$\sigma_1^2(t) = \frac{\sum_{i=i_{\min}}^{t} p(i)(i-\mu_1(t))^2}{\omega_1(t)}, \quad \sigma_2^2(t) = \frac{\sum_{t+1}^{i_{\max}} p(i)(i-\mu_2(t))^2}{\omega_2(t)}$$

- Intra-class variance (or *within class variance*)

$$\sigma_w^2(t) = \omega_1(t)\sigma_1(t)^2 + \omega_2(t)\sigma_2(t)^2$$

- Inter-class variance (or *between class variance*)

$$\begin{aligned} \sigma_b^2(t) = \sigma_T^2 - \sigma_w^2(t) &= \omega_1(t)(\mu_1(t)-\mu_T)^2 + \omega_2(t)(\mu_2(t)-\mu_T)^2 \\ &= \omega_1(t)\omega_2(t)\left(\mu_1(t)-\mu_2(t)\right)^2 \quad \text{(why?)} \end{aligned}$$

Otsu clustering

- Computationally easier to maximise $\sigma_b^2(t)$ (pre calculations, easy updates).

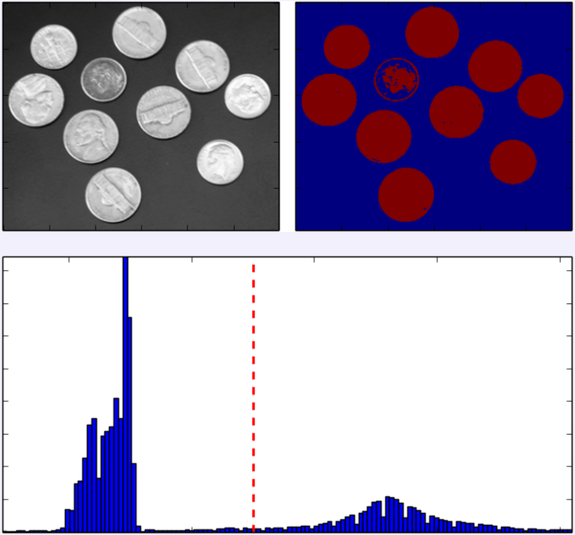
- Classically: exhaustive search: Try all thresholds t from $i_{\min}$ y $i_{\max}$.

- returns the one which maximises σ_b^2.

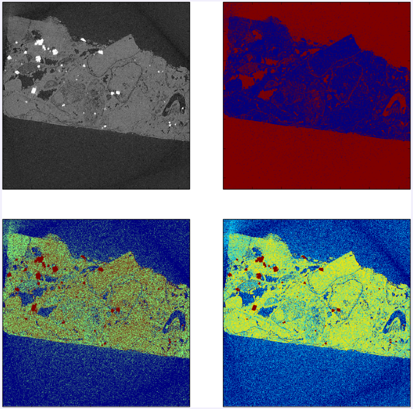
- Easily generalized to more classes/regions.

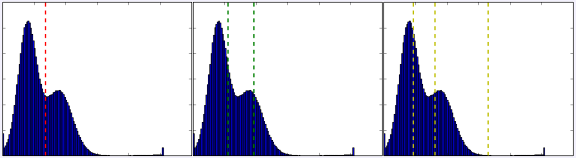
- Can search for local thresholds (on sub-images).

Example



Multiple Classes Otsu

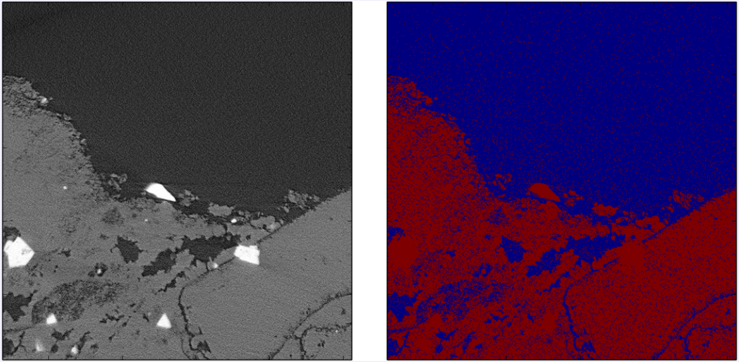




Outline

Clustering and Noise

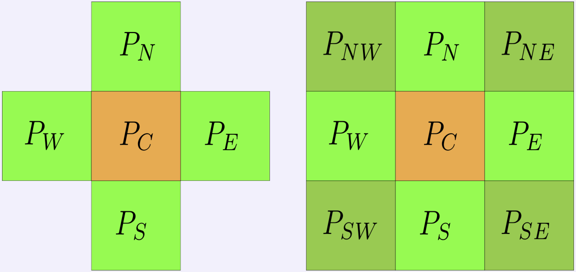
- Intensity clustering discard does not use spatial proximity grouping
- Noise disturbs clustering.



Computed X-Ray tomography from a sandstone sample. Relatively high noise level.

Local Spatial Coherence

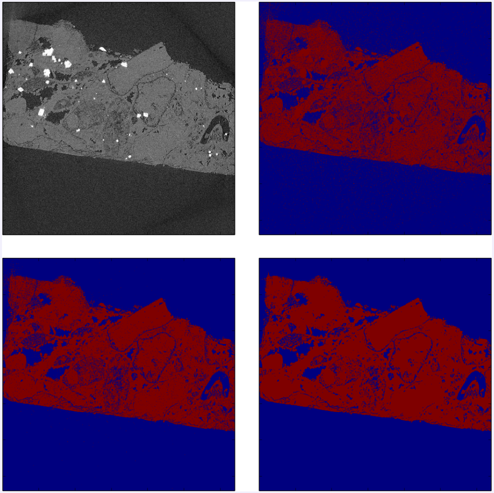
- Prevent small holes: pixel of class A surrounded by pixels of class B: remember proximity grouping.
- Local proximity defined in term of pixel neighborhoods



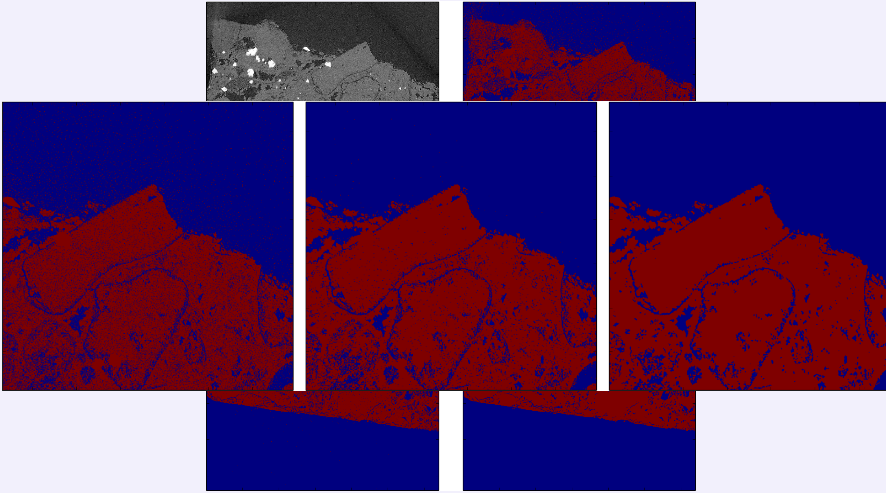
4 and 8 pixels neighbourhood systems.

- If neighbour pixels have same (or almost same) labels, replace centre pixel by dominant one.

Example



Example



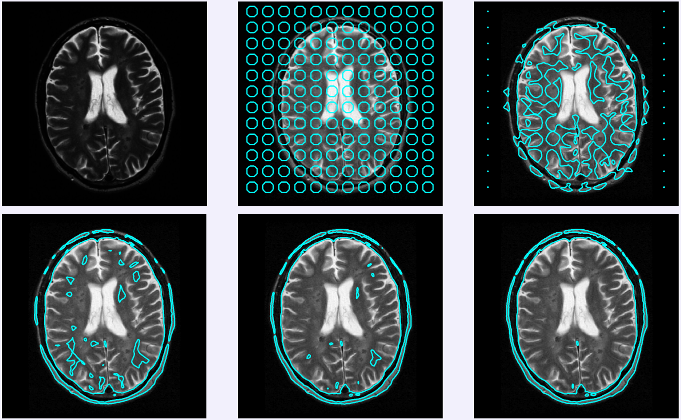
Clustering Snakes: Chan-Vese Algorithm

- Mixes ideas: Clustering (k-means), Snakes a.k.a Active contours, Spatial Regularization.
- Performs clustering while enforcing region coherence – remove small holes.
- Continuous formulation: find a curve C, numbers u_0, u_1 (class centroids) minimizing

$$\mathcal{E}(C, u_0, u_1) = \int_{\mathrm{int}(C)} (u - u_0)^2\, dx + \int_{\mathrm{ext}(C)} (u - u_1)^2\, dx + \lambda \mathrm{length}(C)$$

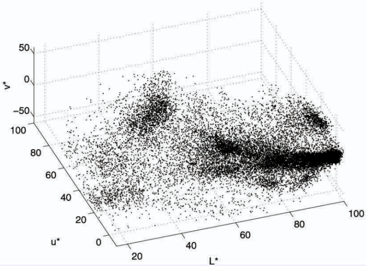
- Minimization by solving a Partial Differential equation for C.
- Complex, non linear.
- Very effective implementations – level sets, relaxations.
- Hundreds of derived methods.

Chan Vese Example



Outline

Exploring Feature Spaces

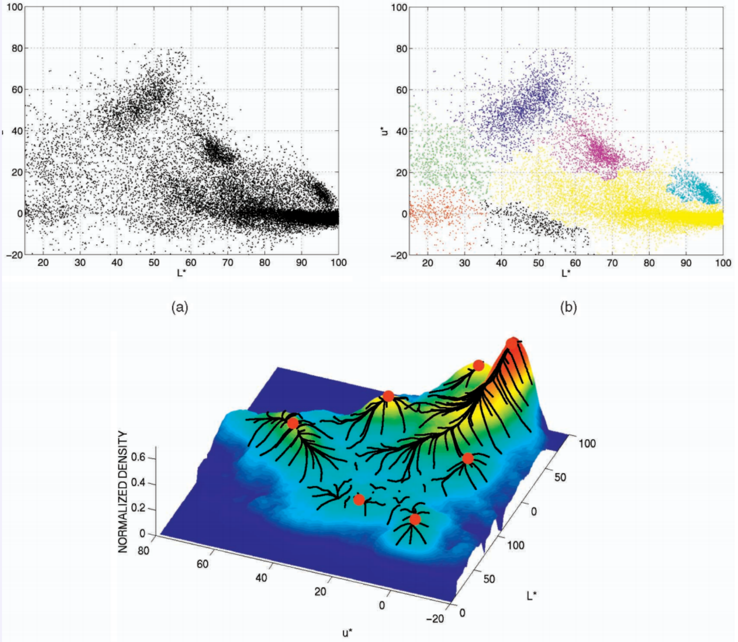


Exploration of the colour space of the image.[4]

---

[4]From D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE PAMI 2002.

Mean Shift Exploration



(a)

(b)

Mean Shift

Find the modes (local maxima) of a sampled density function. Given radius, kernel function K.

- Start with a (random or not) sample x.
- Do until convergence:
  1. Set $N(x)$ the set of sample points x_i such that $d(x, x_i) \leq$ radius.
  2. Compute weighted means of samples in a neighbourhood of x:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x, x_i) x_i}{\sum_{x_i \in N(x)} K(x, x_i)}$$

  3. if $\|m(x) - x\| >> 0$ set $x := m(x)$, go to 1. Else output x.

Mean Shift

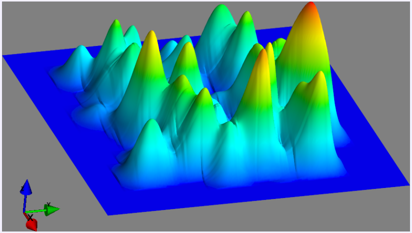
Find the modes (local maxima) of a sampled density function. Given radius, kernel function K.

- Start with a (random or not) sample x.
- Do until convergence:
  1. Set $N(x)$ the set of sample points x_i such that $d(x, x_i) \leq$ radius.
  2. Compute weighted means of samples in a neighbourhood of x:
  $$m(x) = \frac{\sum_{x_i \in N(x)} K(x, x_i) x_i}{\sum_{x_i \in N(x)} K(x, x_i)}$$
  3. if $\|m(x) - x\| >> 0$ set $x := m(x)$, go to 1. Else output x.

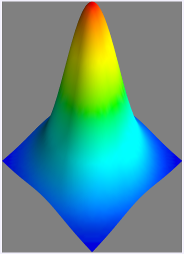
Kernel function: Positive function $K(x, y)$ which measures similarity between x and y:

- x and y very similar: $K(x, y) >> 0$.
- x and y very dissimilar: $K(x, y) << 1$.
- Very usual choice: Gaussian kernel $K(x, y) \propto e^{-\dfrac{\|x-y\|^2}{2\sigma^2}}$.

Mean Shift is Iterative Fitting



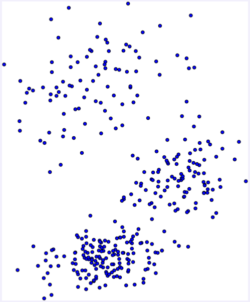
Density function



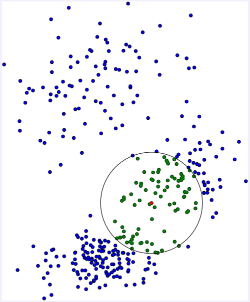
Kernel function

- Fit locally kernel to density function.
- Shifted kernel centre = local mode.
- Density function only known from samples
  sample density $\propto$ function value.

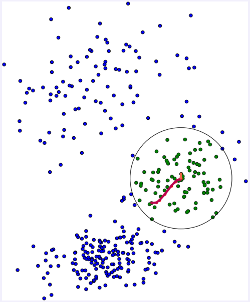
Example[5]



Dataset



Start of Mean shift

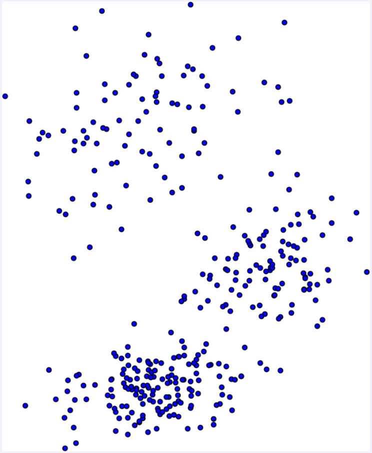


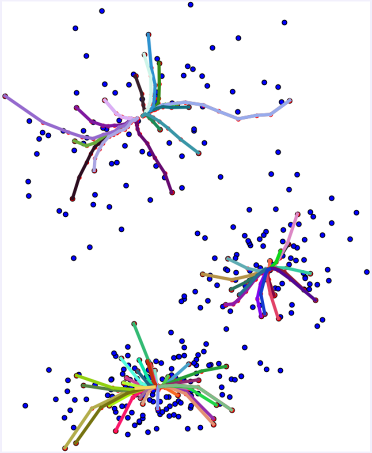
Shift to local mode.

[5] You can play with provided Python illustration code `ms_demo.py` and change some of its parameters.

Mean Shift and Grouping

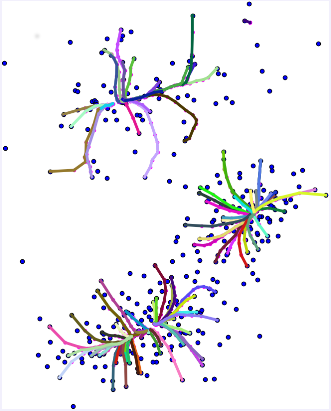
Group points with common modes.

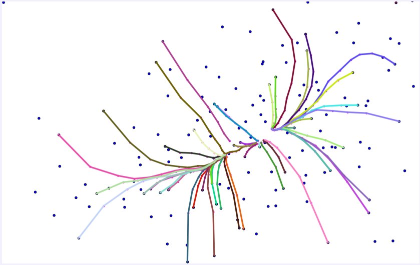




- Attraction Basin (cf. Watershed catchments bass in): the region made by feature vectors that share the same mode.
- Clusters/Groups: all data points in the attraction basin of a given mode.

Mean shift, radius, kernel





In practice, may obtain close but distinct modes. Depends on neighbourhood size, kernel...

Mean Shift and Segmentation

- Compute features vectors per pixel range and position features:
  - range features: grey value, colour, Gaussian features.
- Position feature helps spatial homogeneity.
- Define a kernel adapted to these features, and a feature space radius.
- Use Mean shift to group points with common modes.
- Beware: very expensive for high dimensional features.
- Need to set kernel and radius parameters, can be tricky.
- Amenable to series of optimizations.

Examples

[6]

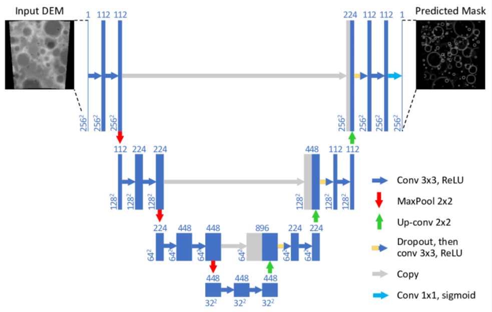
---

[6] From D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE PAMI 2002.

Outline

Deep Learning Approaches

- Nowadays many segmentation algorithms are Deep Learning based.
- Generally *supervised*: they learn segmentation from example.
- The most known is the U-net [Ronneberger, Fisher, Brox 2015].



From A. Silburt *et. al*, "Lunar Crater Identification via Deep Learning", Icarus, 2019

- This type of segmentation algorithm requires *training data*.
- In this case, input images and segmented images.
- But how are the training segmented images obtained?
  - Manual annotation: often used, but requires large time and human resources.
  - (semi-)automatic: use *classical* (i.e., pre-DL) segmentations algorithms: The good ol' methods are not dead!

- Some 3D segmentation tasks may require tremendous memory resources. Simple segmentation methods presented in these lectures are often better adapted (for now?)

- More on CNNs in the next lectures.

Outline

Summary

- Ideas from Gestalt theory: Figure-Ground, grouping: proximity, connectedness, similarity, continuity, closure, common regions.
- Segment as "small" semantic unit.
- Edges and segmentation: Marr-Hildreth edge detector, Canny Edge detector.
- Closing the gap. Perceptual aspects (amodal completion), operationalization: Snakes - a.k.a. Active contours.
- Soft edges and ridges of "topographic" maps: Watershed algorithm
- Feature proximity, clustering, K-Means, Otsu.
- Noise and Gap. Spatial regularization.
- Chan-Vese: Clustering Snakes.
- Mode analysis for grouping and Segmentation: Mean Shift.

francois@francois-vision: ~/temp/Segmentation

```
francois@francois-vision:~/temp/Segmentation$ cat bummer.c
/*
 * A very simple segmentation related program.
 * So universal that it does not even need an image!
 */
#include <stdlib.h>
int main(void) {
    int *a = 0;
    *a = 10;
    return EXIT_SUCCESS; /* no way...*/
}
francois@francois-vision:~/temp/Segmentation$ gcc -o bummer bummer.c
francois@francois-vision:~/temp/Segmentation$ ./bummer
Segmentation fault (core dumped)
francois@francois-vision:~/temp/Segmentation$
```