Algorithms for Image Analysis

The University of
Western
Ontario

Basic Image Segmentation and General clustering techniques

Basic Image Segmentation

- Segmentation examples
  - *unsupervised* (background subtraction, color quantization, superpixels)
  - *supervised* (photo-shop, medical image analysis)
- Basic low-level segmentation criteria and methods:
  - for segment region (appearance) – part I
  - for segment boundary – part II
  - thresholding, likelihood ratio test, region growing, watersheds

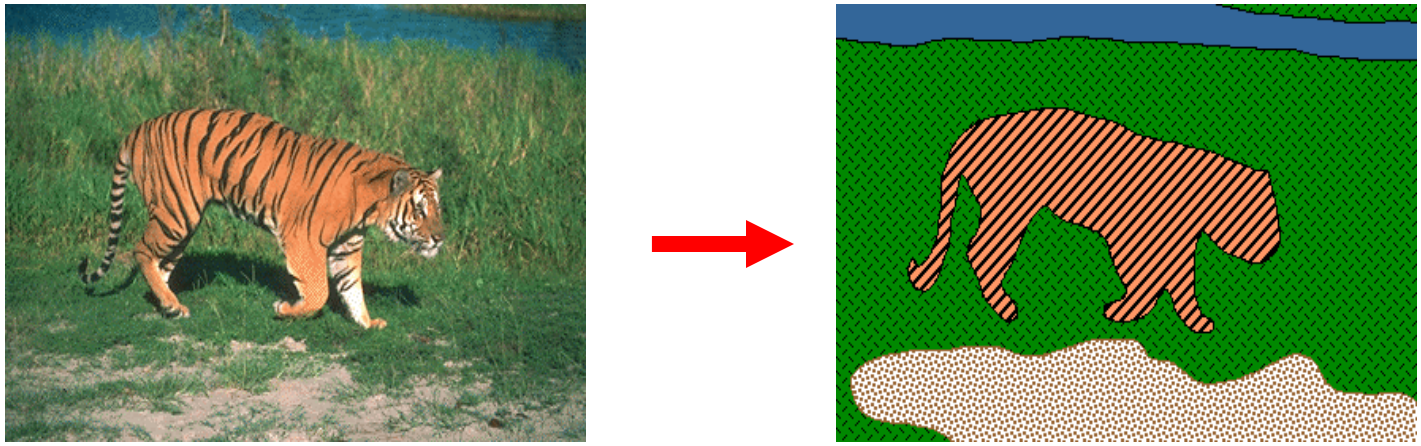
Szeliski, Sec 5.2

- General clustering methods (in the context of image analysis)
  - K-means, Normalized cut, mean-shift – part III

Szeliski, Sec 5.3

Other readings: Sonka at.al. Ch. 5
Gonzalez and Woods, Ch. 10

Segmentation



accurate boundary delineation is often required

Goal:

find coherent "blobs" or specific "objects"

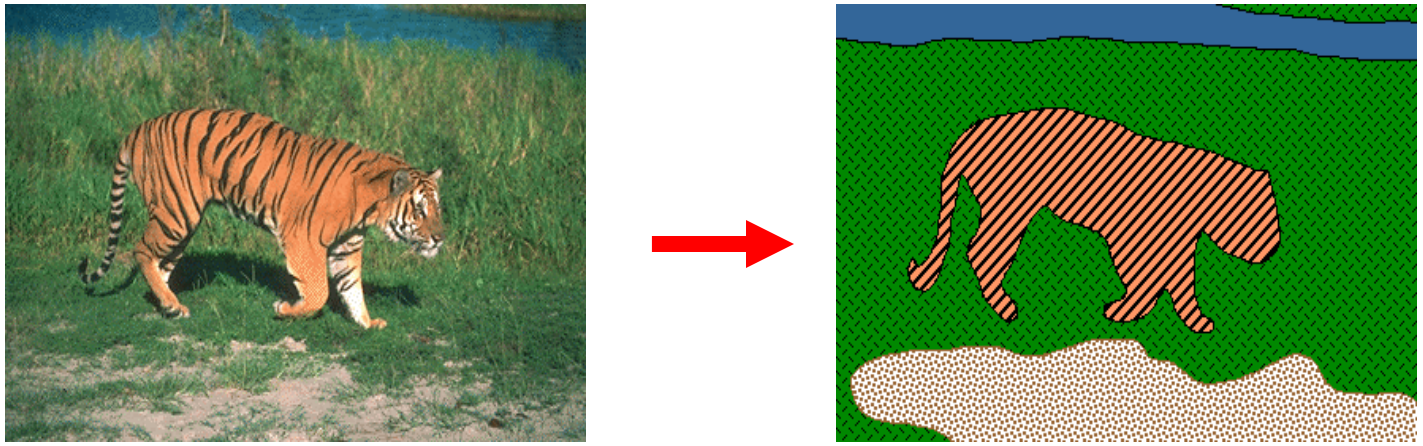
lower level tasks
(e.g. "superpixels")

large grey area
in-between

higher level tasks
(e.g. cars, humans, or organs)

Two major low-level criteria for "good" segmentation

The University of Western Ontario



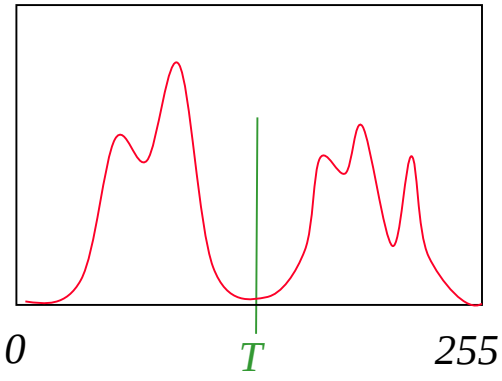
- Coherent **segment's appearance (region)**
  - e.g. consistent brightness, color, texture, etc
  - or agrees with known expected color distribution
- Coherent **segment's shape (boundary)**
  - e.g. alignment to contrast edges
  - or boundary regularity / smoothness (low-level "shape prior")
  - or agrees with known expected global shape (e.g. square, star - higher-level priors)

Part I

SEGMENT'S APPEARANCE

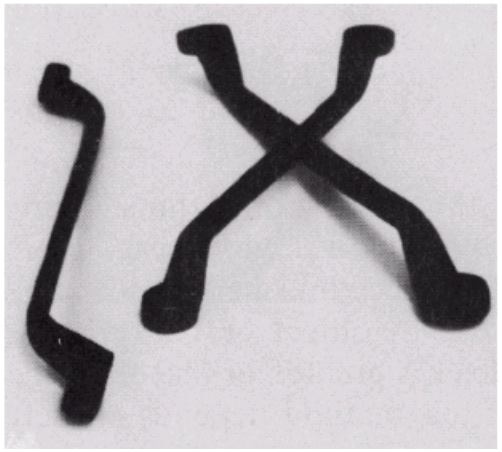
The most basic approach is to partition intensity histogram (**thresholding**)

- point processing, location is ignored



Coherent color "blobs"

- Simplest way to define blob coherence is as similarity in brightness or color:



The tools become blobs

The house, grass, and sky make different blobs

The University of Western Ontario

Why is this useful?

AIBO RoboSoccer (VelosoLab)

Ideal Segmentation

can recognize objects with

known ?

simple ?

color ?

models

Result of a segmentation method

(first learn how to get this, then how to get better results)

even
if

known
simple
color

Basic ideas

segment's region features		basic (naïve) methods
intensities, colors	←	thresholding, likelihood ratio test

(segmentation ← intensities/colors)

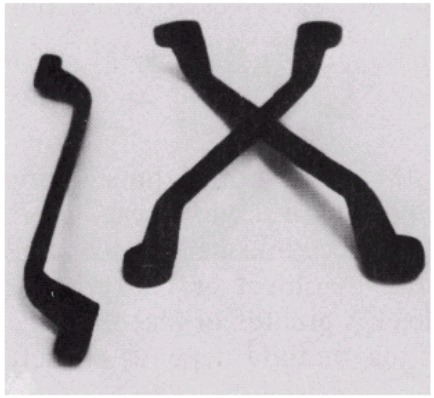
Thresholding

- Basic segmentation operation:

  $mask(x,y) = 1 \text{ if } im(x,y) > T$

  $mask(x,y) = 0 \text{ if } im(x,y) < T$

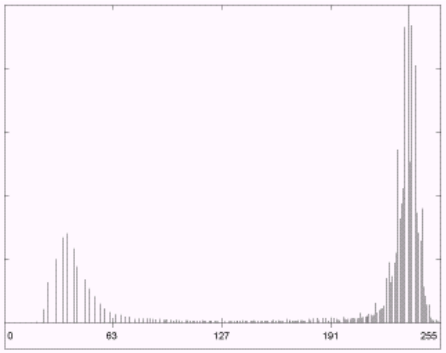
- T is threshold
  - user-defined
  - or automatic

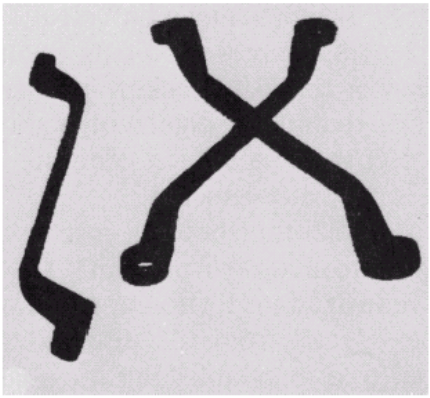


a
b c

FIGURE 10.28 (a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

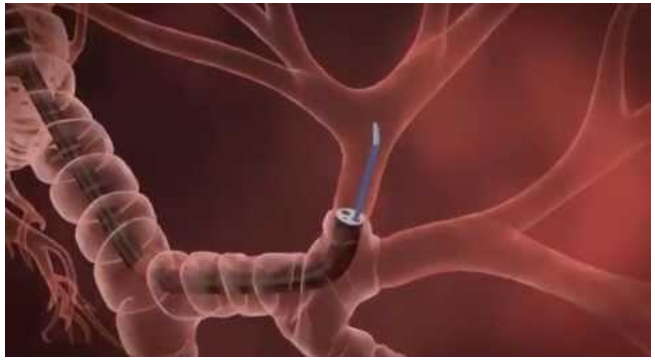
- Same as histogram partitioning:





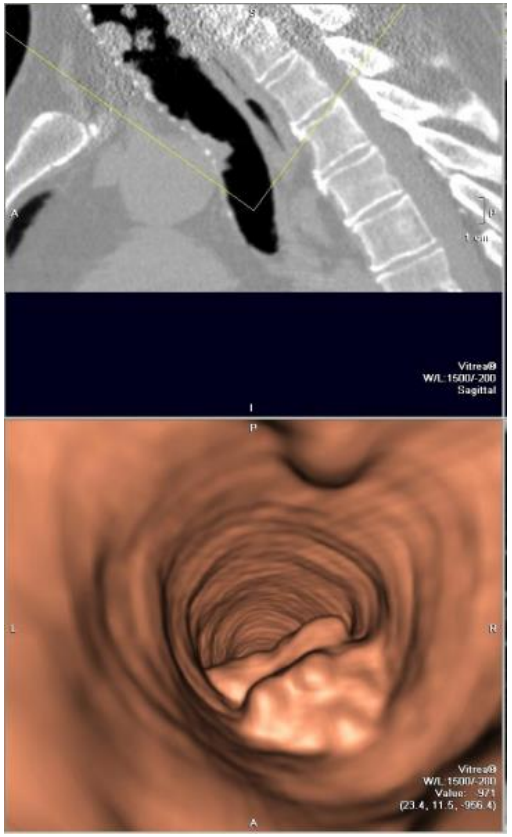
Sometimes works well…

Virtual colonoscopy, bronchoscopy, etc.



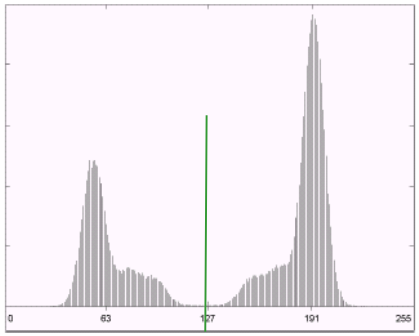
from real device to non-invasive virtual test

a) threshold CT volume -> binary mask



b) extract surface mash from binary mask (*fast marching cubes* method)

Sometimes works well…



a b
c

FIGURE 10.29
(a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

Thresholding could be derived as statistical decision: **likelihood ratio test**

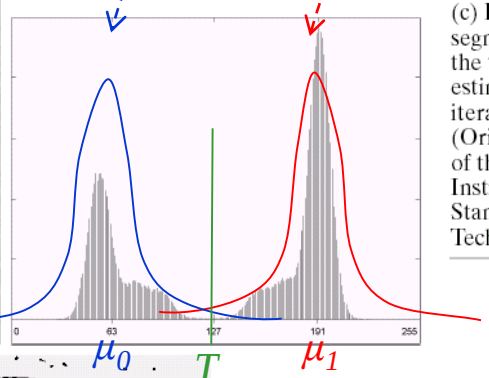
$$r_p := \log \frac{P_1(I_p)}{P_0(I_p)}$$

P_1 and P_0 are object and background known color models

$r_p \geq 0 \Rightarrow$ pixel p is object

$r_p < 0 \Rightarrow$ pixel p is background

Sometimes works well…



a b c

FIGURE 10.29 (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

Example: assume known probability distributions

$$P_1 = N(\mu_1, \sigma)$$

$$P_0 = N(\mu_0, \sigma)$$

$$r \geq 0 \quad \Leftrightarrow \quad I \geq T$$

$$T = \frac{\mu_1 + \mu_0}{2}$$

Thresholding could be derived as statistical decision: **likelihood ratio test**

$$r_p \quad := \quad \log \frac{P_1(I_p)}{P_0(I_p)}$$

P_1 and P_0 are object and background known color models

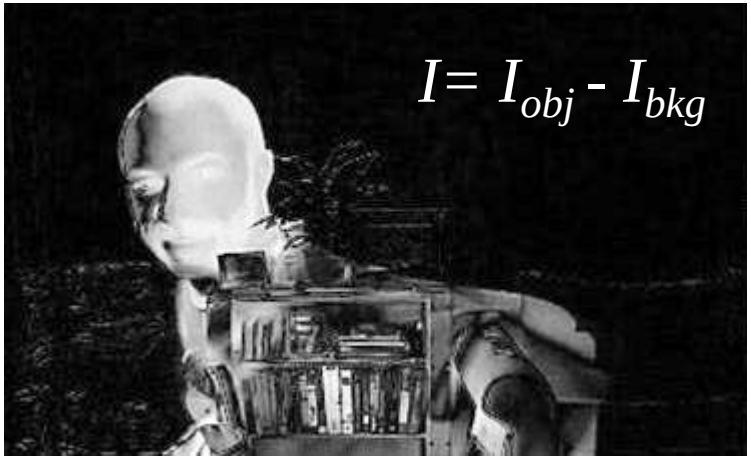
$$r_p \geq 0 \quad \Rightarrow$$ pixel p is object

$$r_p < 0 \quad \Rightarrow$$ pixel p is background

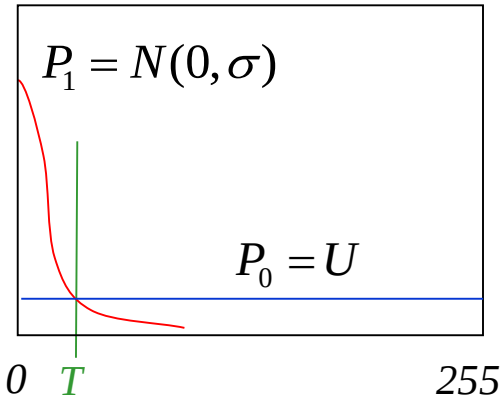
Sometimes works well…

-

=



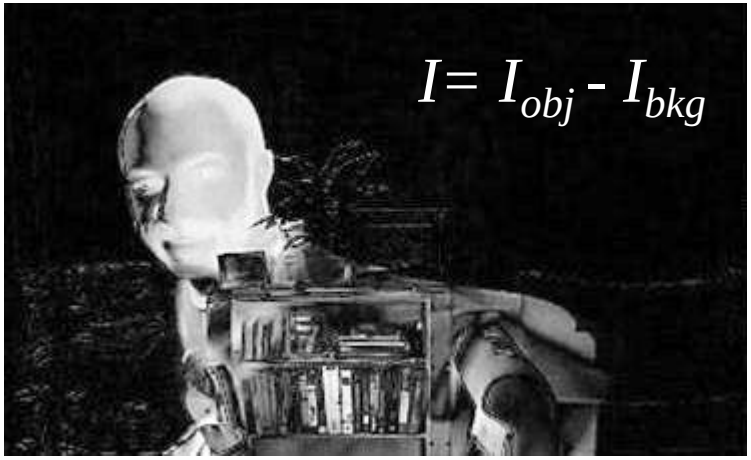
background subtraction



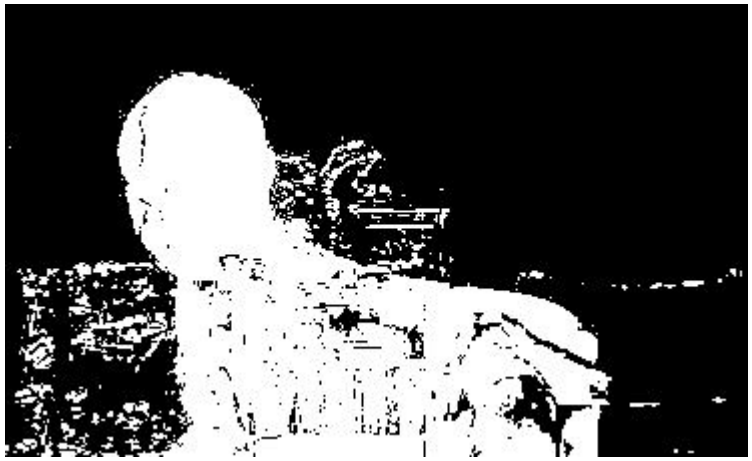
Sometimes works well… more often not

-

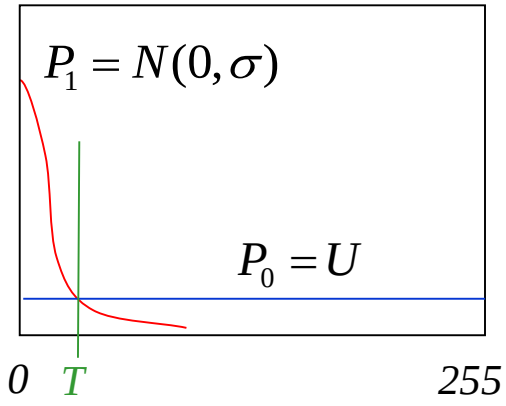
=



background subtraction



Threshold intensities below T



problems when color models have overlapping support

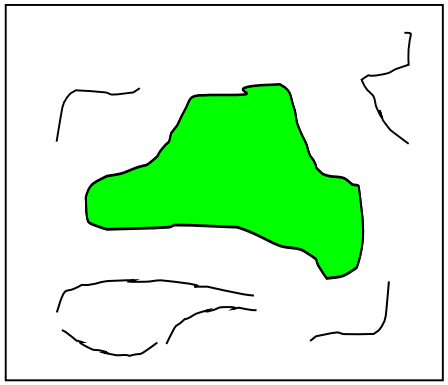
Part II

SEGMENT'S BOUNDARY

The most basic approach finds subset of pixels completely surrounded by strong intensity edges (can use **BFS** over grid)

- ignores segment appearance
- a single "hole" on the boundary ruins the result

Canny edges

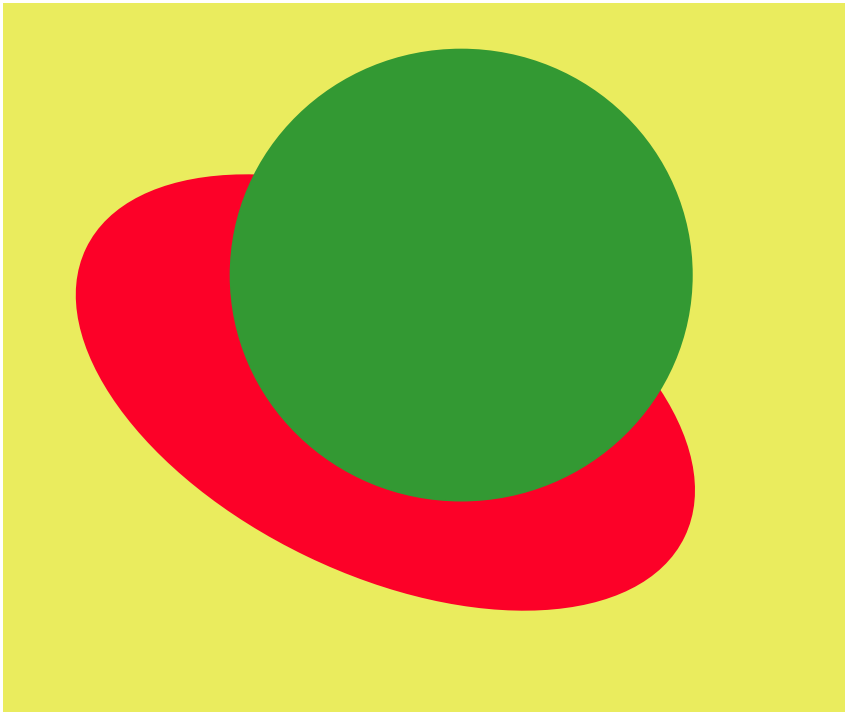


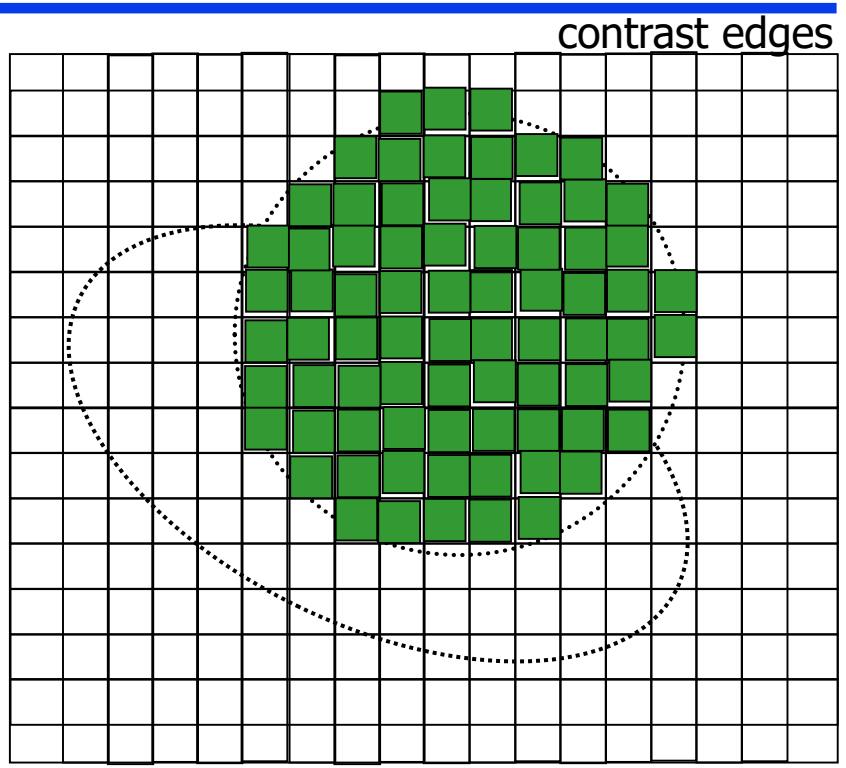
Basic ideas

segment's boundary features		basic (naïve) methods
contrast edges	←	region growing, watersheds

(segmentation ← contrast edges)

Region growing





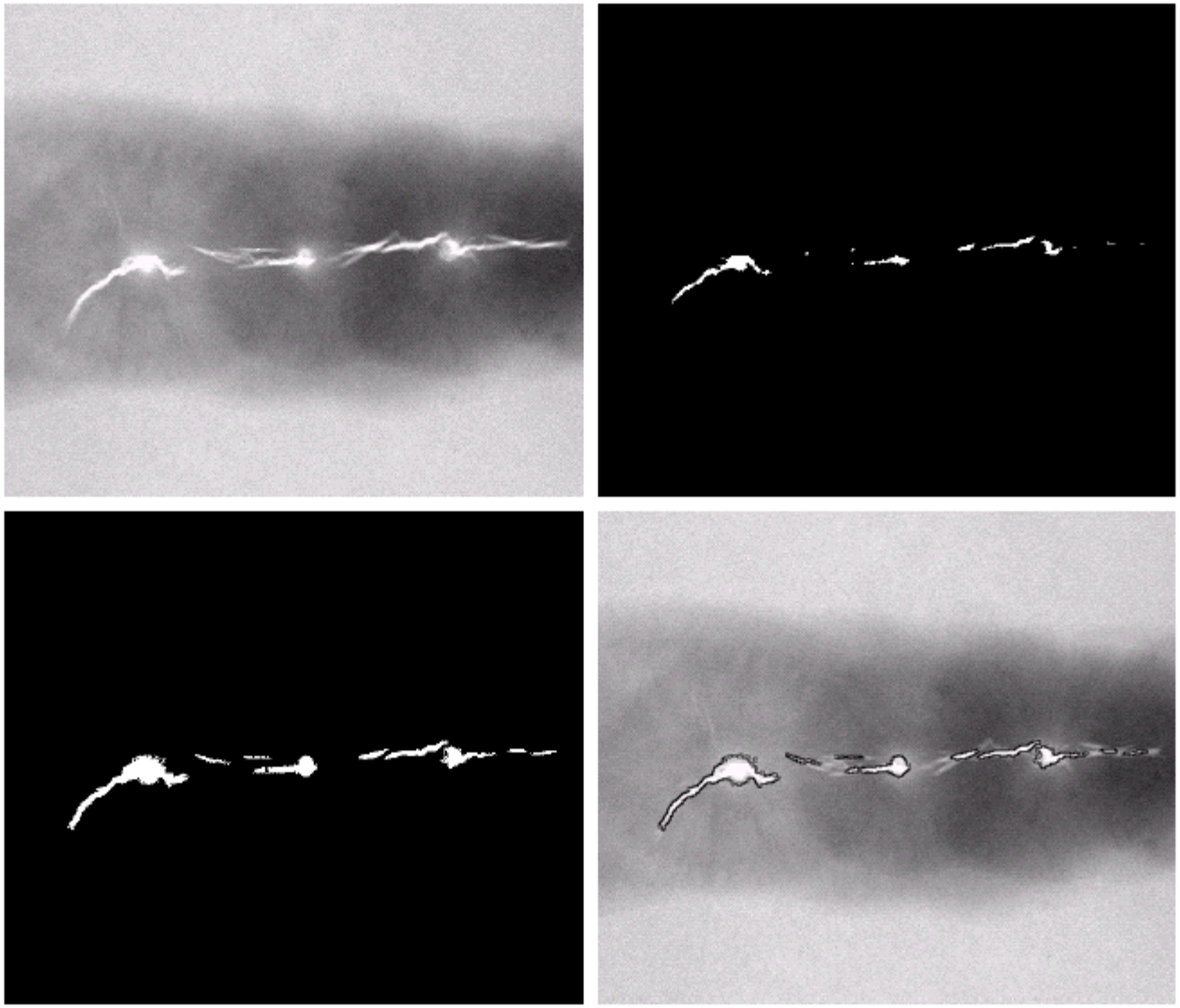
- Start with initial set of pixels K (initial seed(s))
- Add to pixels p in K their neighbors q if $|Ip-Iq| < T$
- Repeat until nothing changes

- **Breadth-First Search** (BFS) over pixel grid edges (p,q) s.t. $|Ip-Iq|<T$
- Method stops at high-contrast edges (p,q) such that $|Ip-Iq|>T$

Region growing

a b
c d

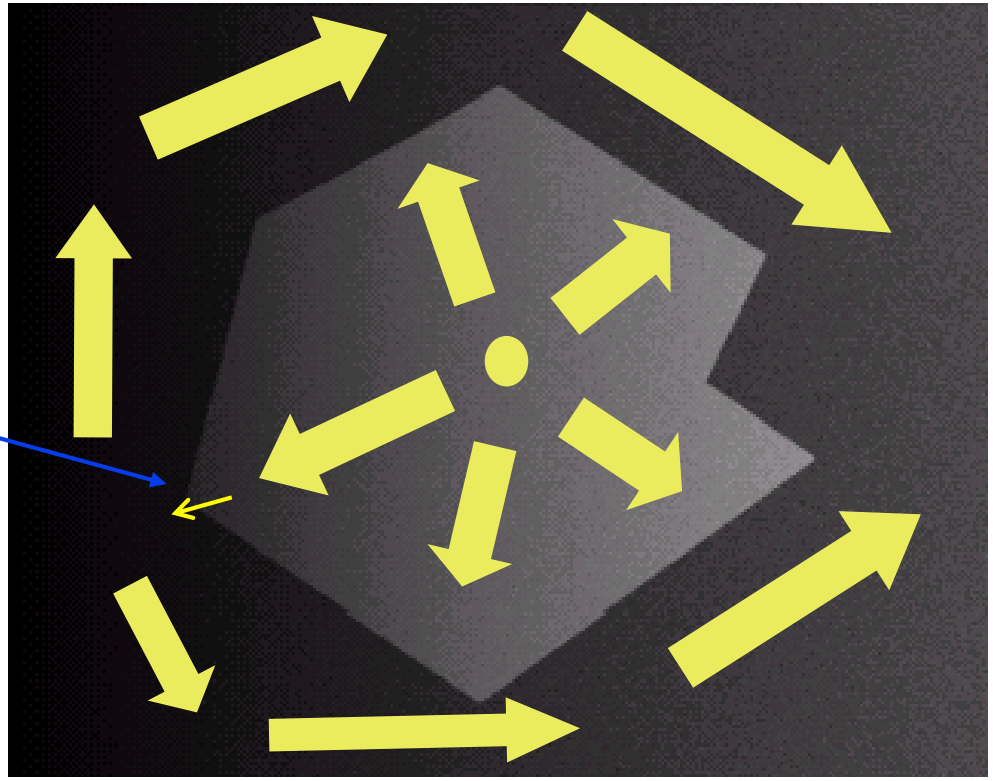
FIGURE 10.40
(a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).



What can go wrong with region growing ?

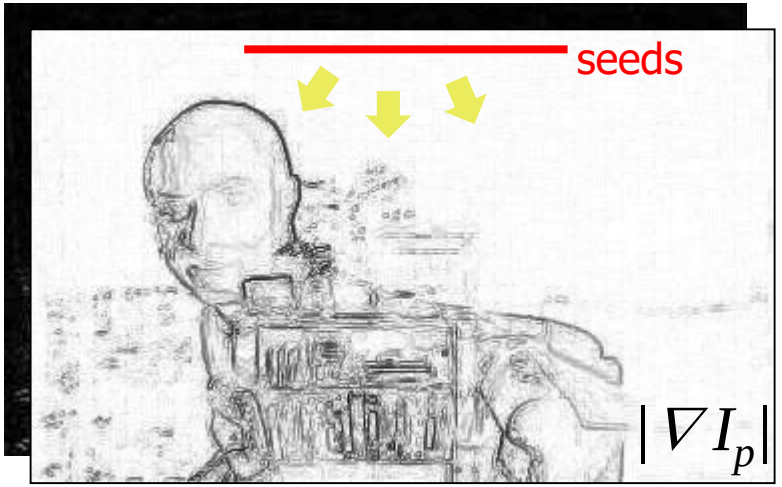
The University of Western Ontario

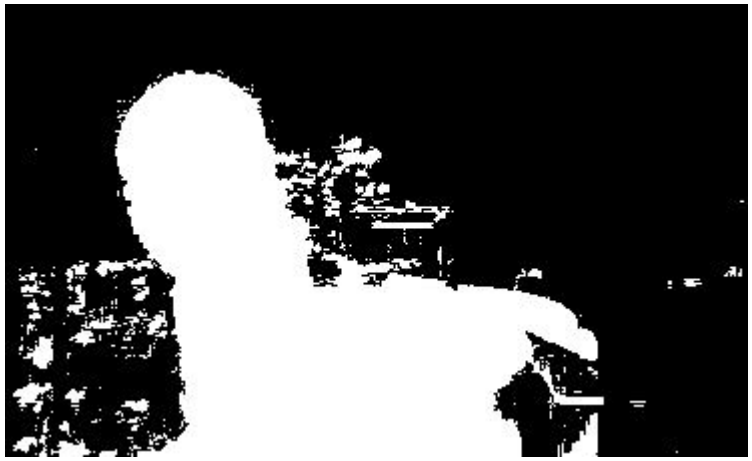
Region growth may “leak” through a single week spot in the boundary



Region growing


-

=




Breadth First Search (seeds) :

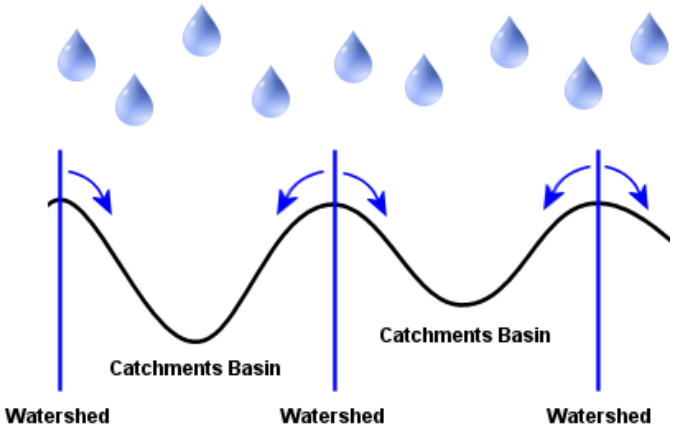
$$|\nabla I_p| < T$$

Region growing

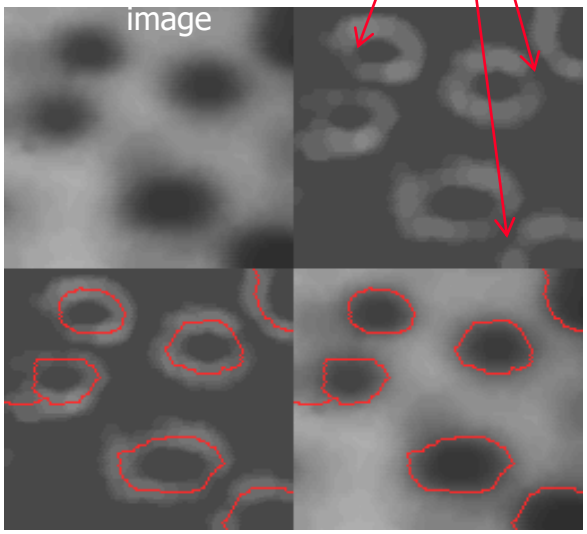
See region *leaks* into sky due to a weak boundary between them

Related "flood-fill" idea: *Watersheds*

The University of Western Ontario



need tricks to build dambs
closing gaps (leaks)



1. gradient magnitudes

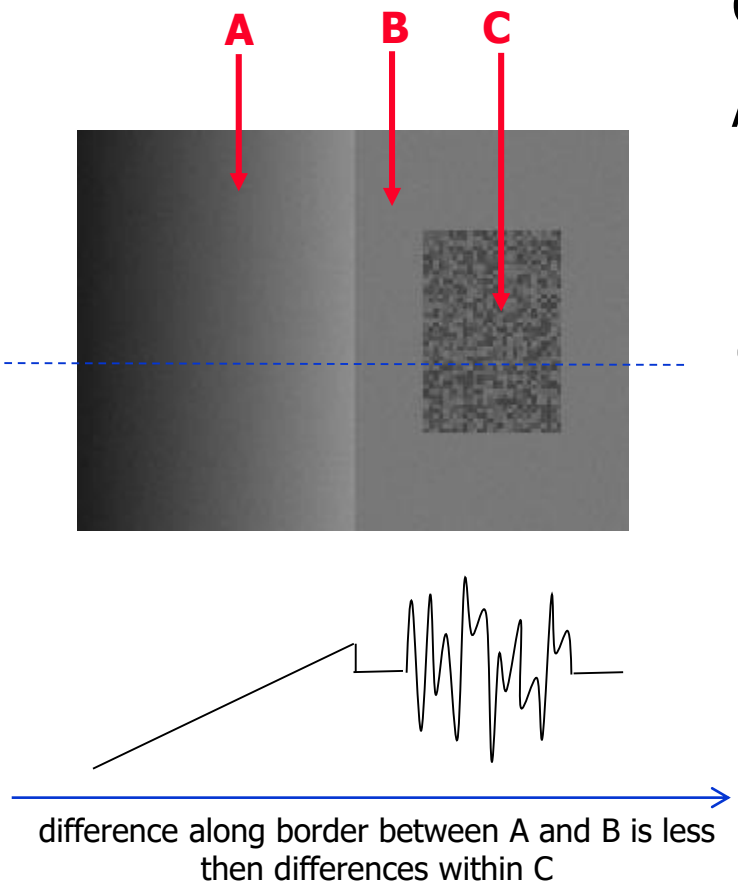
2. find catchment basins

3. copy over original image

Due to Pedro Felzenszwalb and Daniel Huttenlocher

Motivating example

This image has three perceptually distinct regions



Q: Where would **image thresholding** fail?

A: Region A would be divided in two sets and region C will be split into a large number of arbitrary small subsets

Q: Where would **region growing** fail?

A: Either A and B are merged or region C is split into many small subsets
Also, B and C are merged

Towards better segmentation

- Color/appearance criteria (part I)
  - Can we replace (manually-selected) thresholding with automatic "consistent" partitioning of features?
- Boundary criteria (part II)
  - Can we fix leaky boundaries?
  - Can we impose shape priors?

- Can we combine these and other criteria?

How?

Towards better segmentation

- Formulate segmentation quality (objective or energy) functions.
- Combining terms evaluating different properties.

Objectives for segment boundaries (topic 5).
We will learn about measures of **boundary "regularity"**.

$$E(S) = \gamma_1 C(S) + \gamma_2 B(S) + \ldots$$

Consistency of appearance (e.g. color).
We will learn how to evaluate
"consistency" of features partitioning
(clustering objectives, part III)

relative importance (weight)
for each property

- Optimize = find "objectively" the best solution S

Part III:

General clustering techniques

(feature consistency objectives)

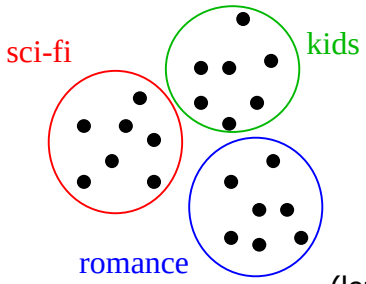
in the context of image analysis problems

- color quantization (RBG features)
- superpixels (RGBXY features)
- unsupervised segmentation (RGBXY features)

The University of Western Ontario

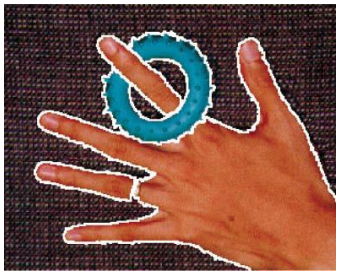
"Terminological" comments on clustering and segmentation

Clustering – general techniques for partitioning arbitrary data set $\{ f_i \} \subset R^N$ (of any dimensions) where i are data points' indices



e.g. movies in some feature space (length, director, tags,…)

(Image) Segmentation – methods for partitioning specifically image features $\{ f_p \} \subset R^N$ where p are image grid pixels $p = \{ x_p , y_p \}$



Comment: in practice, the boundary between terms **clustering** and **segmentation** is very fuzzy.

Different approaches to image segmentation

The University of Western Ontario

f_p (e.g. I_p)

- *e.g. intensity or color feature at pixel p = (x,y)*

I_p is a triple $\{I_p, x_p, y_p\}$
i.e. data point in RGBXY space (R^5)
where p represents pixel's index

general clustering techniques (part I)
in "joint" feature space (R^5) combining pixel location (XY) and color (e.g. RGB).

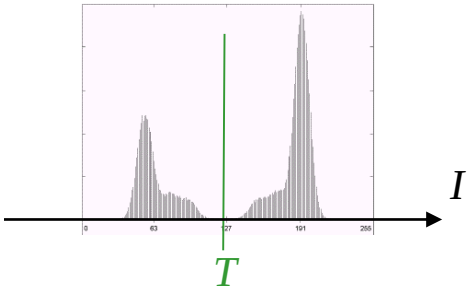
I_p is a mapping or function $I(p)$
from grid points XY to intensities RGB

e.g. **graph partitioning techniques** (part II)
where pixel location (XY) can be treated differently from pixel's feature (RGB).
e.g. shape (XY) + feature consistency (RGB)

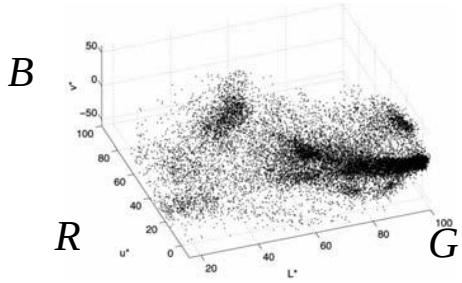
In general: clustering techniques are used in image analysis for feature spaces like: RGB, RGBXY, RGBD, etc.

Motivation

- In 1D feature spaces (gray-scale intensities) it is possible to set decision boundaries manually.

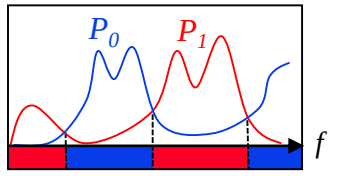


Not so easy in RGB space.



- It is possible to use log-likelihood ratio test if (color) distributions for segments, *e.g.* P_1 and P_0, are known.

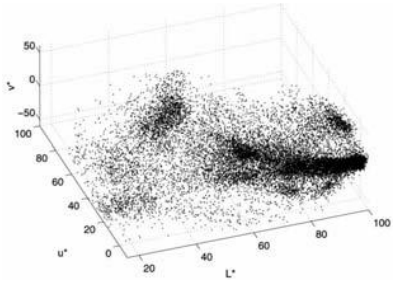
$$\log \frac{P_1(f)}{P_0(f)} \geq 0$$



can yield complex decision boundaries in space of any features f

Does not work if distributions are not known.

Motivation



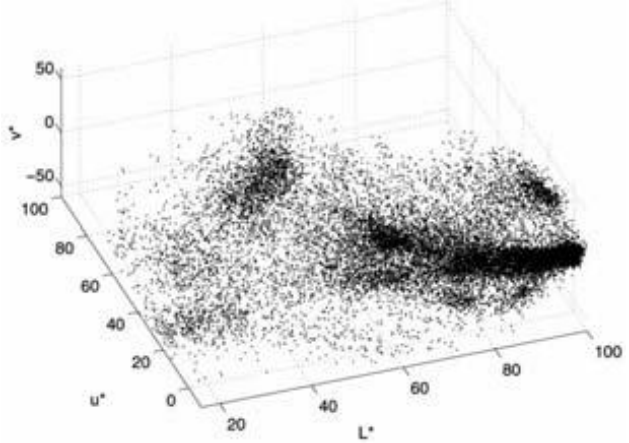
Need automatic data *clustering* methods

- parametric methods: K-means
- non-parametric: mean-shift, normalized cut

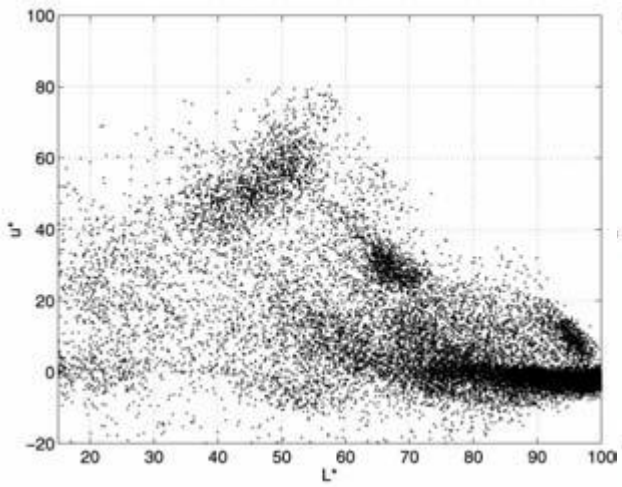
Szeliski, Sec 5.3

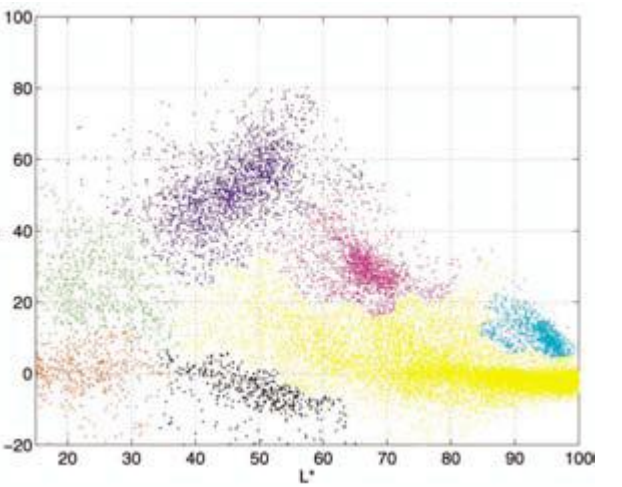
Decision boundaries in feature spaces

should break colors (in RGB or LUV space) into multiple clusters



color quantization
superpixels

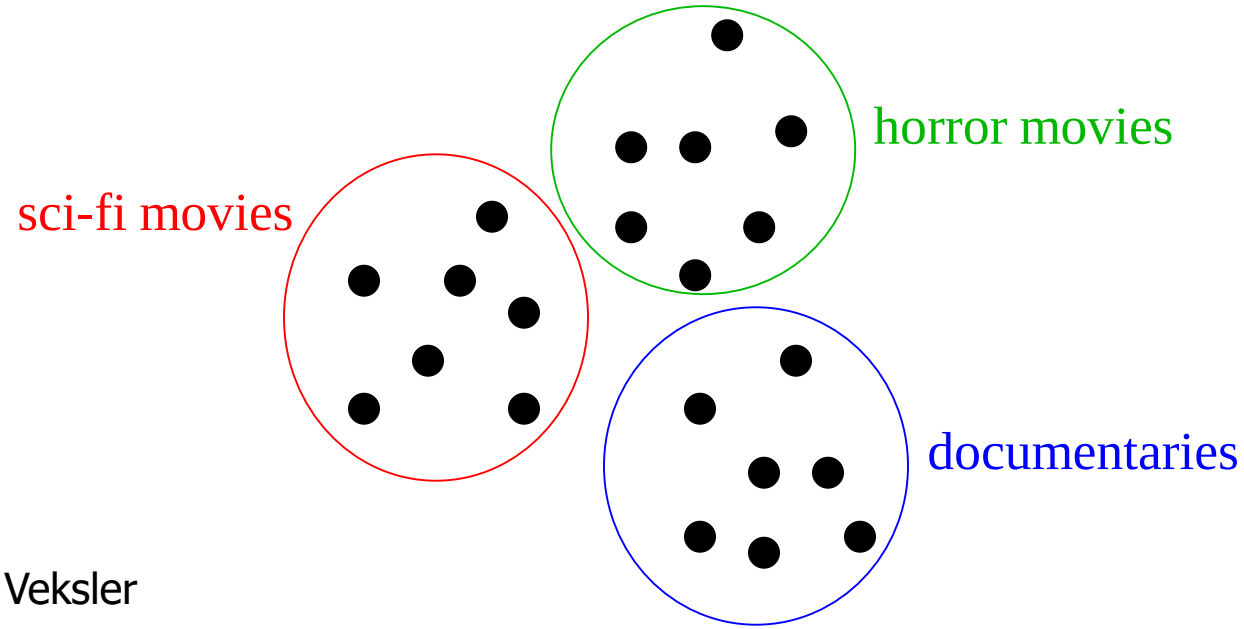




General Grouping or Clustering

(a.k.a. *unsupervised learning*)

- Have data points (samples, a.k.a. feature vectors, examples, etc.) $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_p,\ldots$
- Cluster similar points into groups
  - points are not pre-labeled
  - think of clustering as 'discovering' labels



slides from Olga Veksler

How does this Relate to Image Segmentation?

- Represent image pixels as feature vectors $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_p,\ldots$ or $\{ f_p \mid p \in \Omega \}$
  - For example, each pixel can be represented as
    - intensity, gives one dimensional feature vectors
    - color, gives three-dimensional feature vectors
    - color + coordinates, gives five-dimensional feature vectors

set of all pixels or indices

- Cluster them into ***K*** clusters, i.e. ***K*** segments

input image

9 4 2	7 3 1	8 6 8
8 2 4	5 8 5	3 7 2
9 4 5	2 9 3	1 4 4

feature vectors for clustering based on color

[9 4 2]	[7 3 1]	[8 6 8]
[8 2 4]	[5 8 5]	[3 7 2]
[9 4 5]	[2 9 3]	[1 4 4]

RGB (or LUV) space clustering

How does this Relate to Image Segmentation?

- Represent image pixels as feature vectors $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_p,\ldots$ or $\{ f_p \mid p \in \Omega \}$
  - For example, each pixel can be represented as
    - intensity, gives one dimensional feature vectors
    - color, gives three-dimensional feature vectors
    - color + coordinates, gives five-dimensional feature vectors

set of all pixels or indices

- Cluster them into ***K*** clusters, i.e. ***K*** segments

input image

9 4 2	7 3 1	8 6 8
8 2 4	5 8 5	3 7 2
9 4 5	2 9 3	1 4 4

feature vectors for clustering based on color and image coordinates

[9 4 2 0 0] [7 3 1 0 1] [8 6 8 0 2]

[8 2 4 1 0] [5 8 5 1 1] [3 7 2 1 2]

[9 4 5 2 0] [2 9 3 2 1] [1 4 4 2 2]

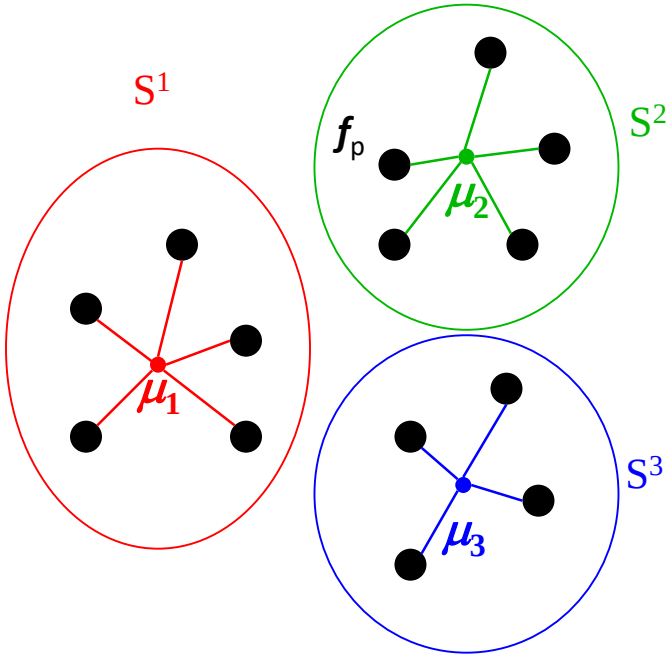
RGBXY (or LUVXY) space clustering

K-means Clustering: Objective Function

- Probably the most popular clustering algorithm
  - assumes the number of clusters is given - ***K***
- optimizes (approximately) the following objective function for variables S^k and μ_k

$$SSE = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|^2$$

sum of squared errors from cluster center μ_k



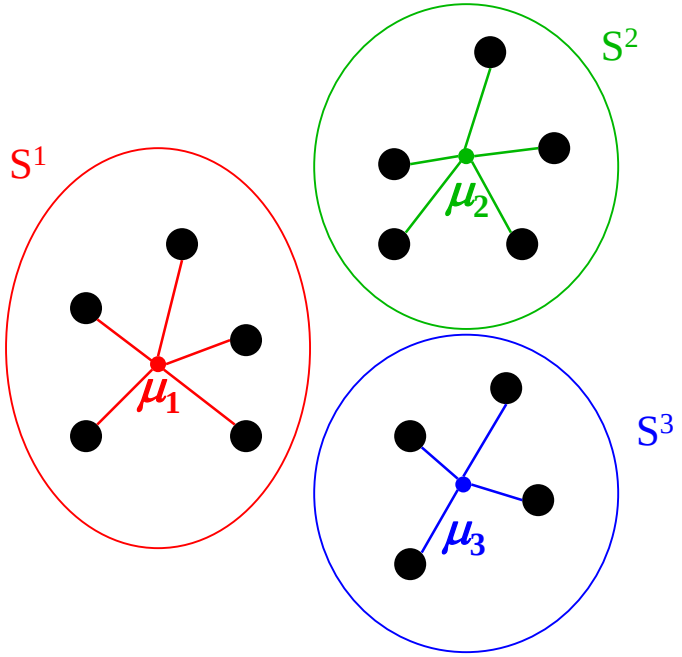
$$SSE = \quad + \quad +$$

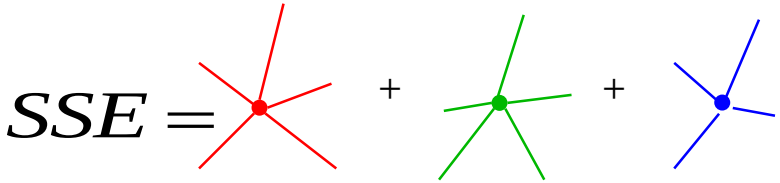
optimization "variables"

subsets $S = (S^1, \ldots, S^K)$

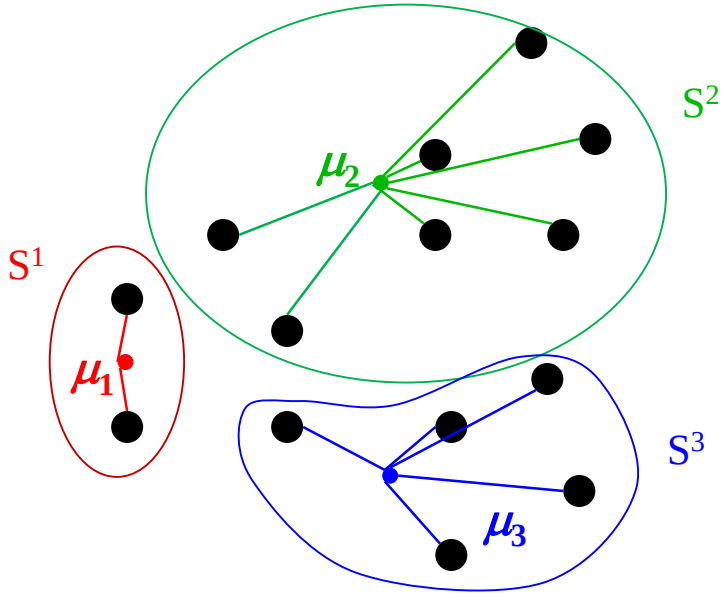
means $\mu = (\mu_1, \ldots, \mu_K)$

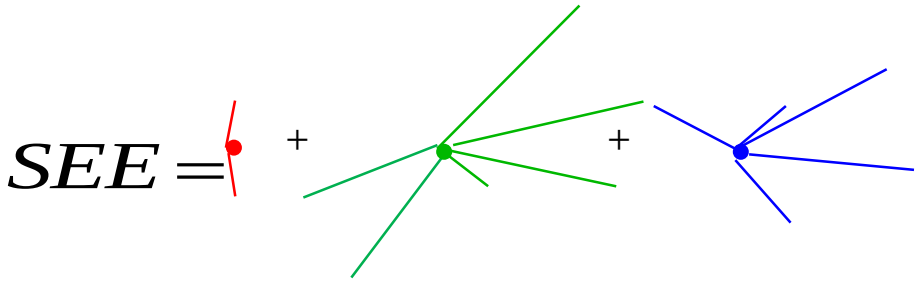
K-means Clustering: Objective Function





Good (tight) clustering
smaller value of SSE

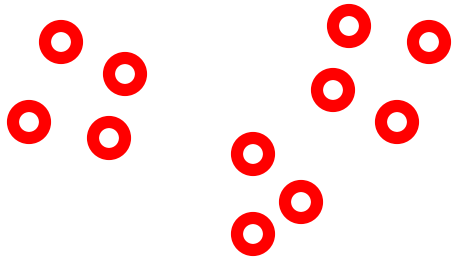




Bad (loose) clustering
larger value of *SSE*

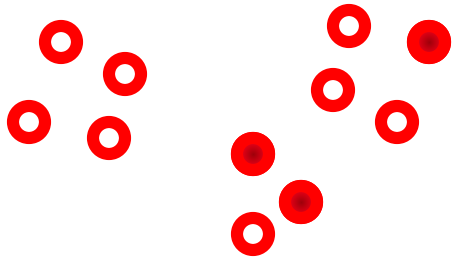
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly



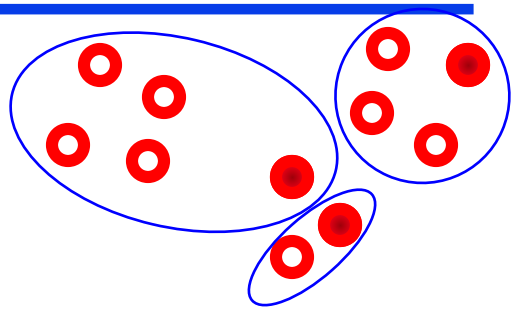
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly



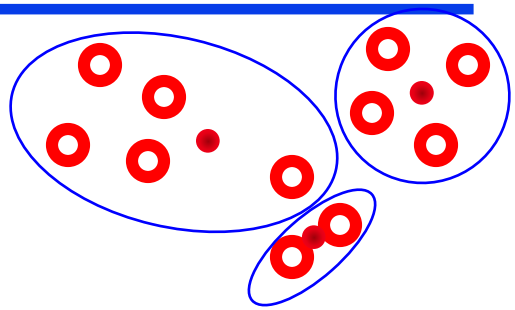
K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to closest center



K-means Clustering: Algorithm

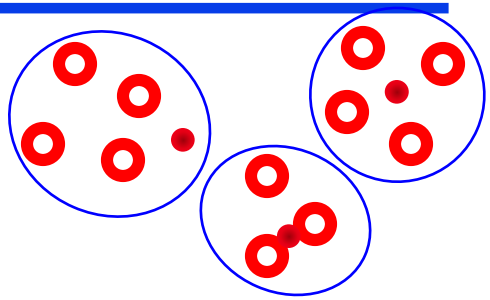
- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$

K-means Clustering: Algorithm

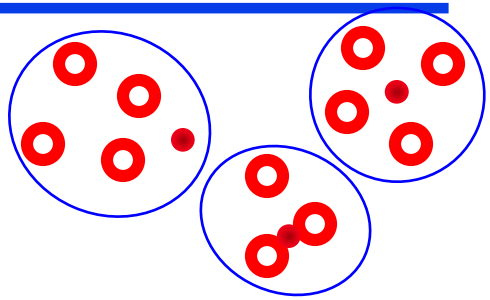
- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean

K-means Clustering: Algorithm

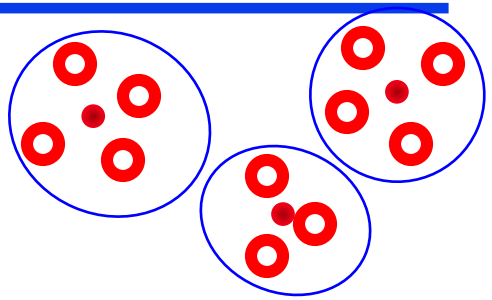
- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

K-means Clustering: Algorithm

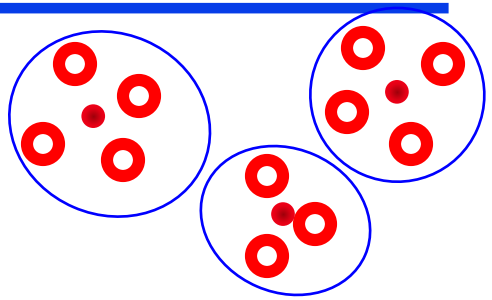
- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

K-means Clustering: Algorithm

- Initialization step
  1. pick ***K*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_k = \frac{1}{|S^k|} \sum_{p \in S^k} f_p$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

- This procedure decreases the value of the objective function

$$E_K(S, \mu) = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|^2$$

optimization variables

$$S = (S^1, \ldots, S^K)$$
$$\mu = (\mu_1, \ldots, \mu_K)$$

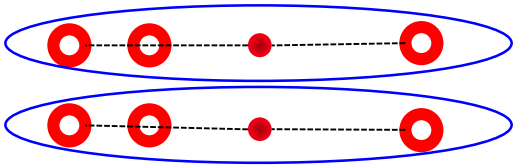
block-coordinate descent: step 1 optimizes μ, step 2 optimizes S

K-means: Approximate Optimization

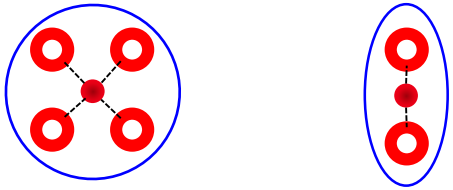
- K-means is fast and often works well in practice
- But can get stuck in a local minimum of objective E_K
  - not surprising, since the problem is NP-hard

initialization

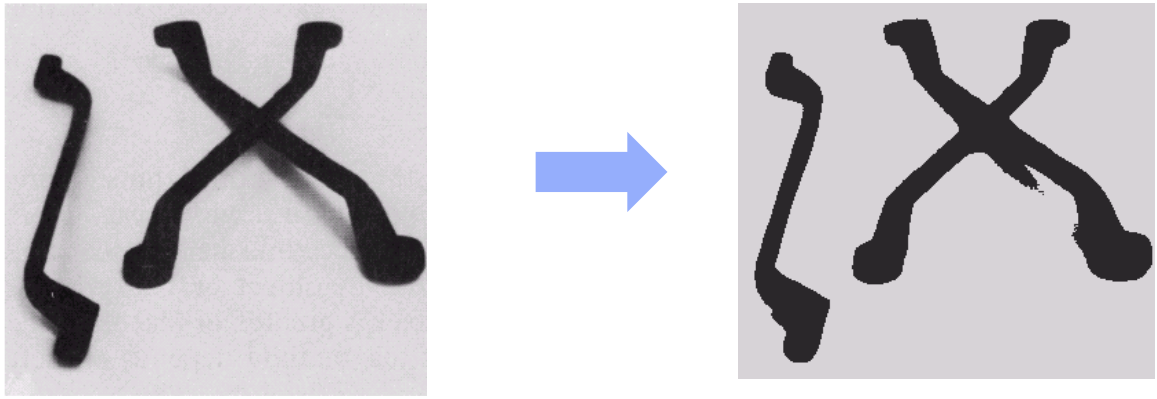
converged to local min

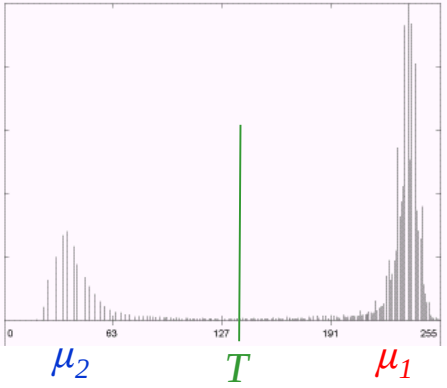


global minimum



K-means clustering examples: Segmentation



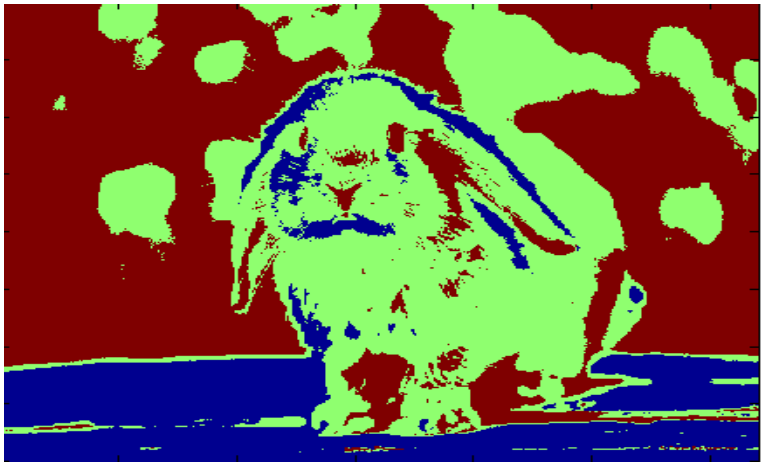


K-means finds compact clusters

In this case K-means (K=2) automatically finds good threshold (between 2 clusters)

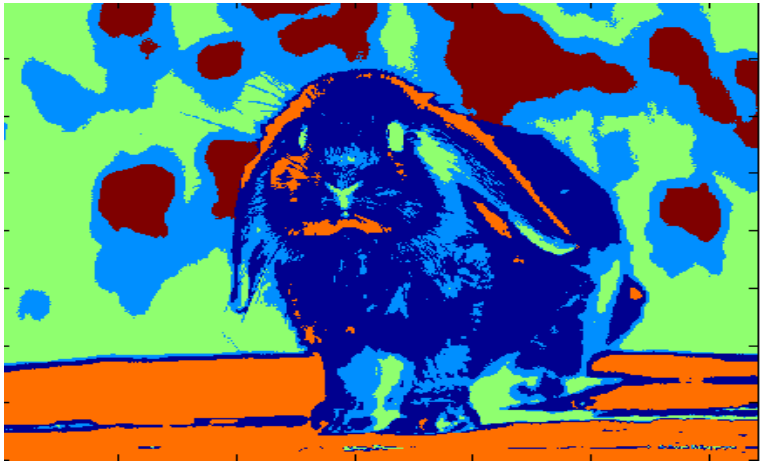
K-means clustering examples: Segmentation?

The University of Western Ontario

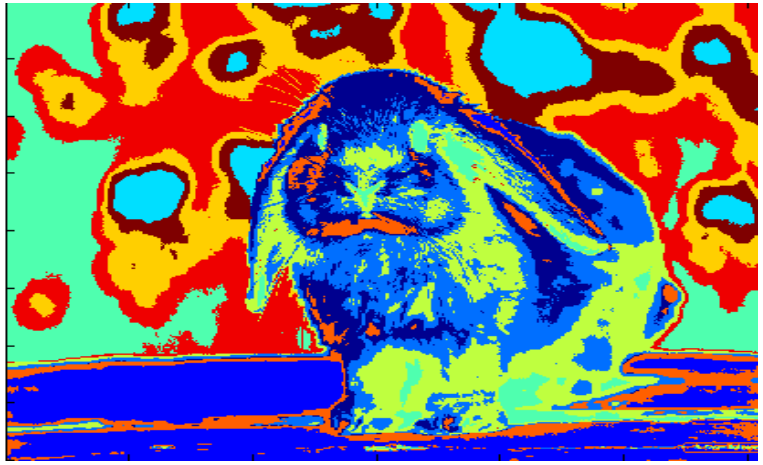


***k* = 3**

(random colors are used to better show segments/clusters)



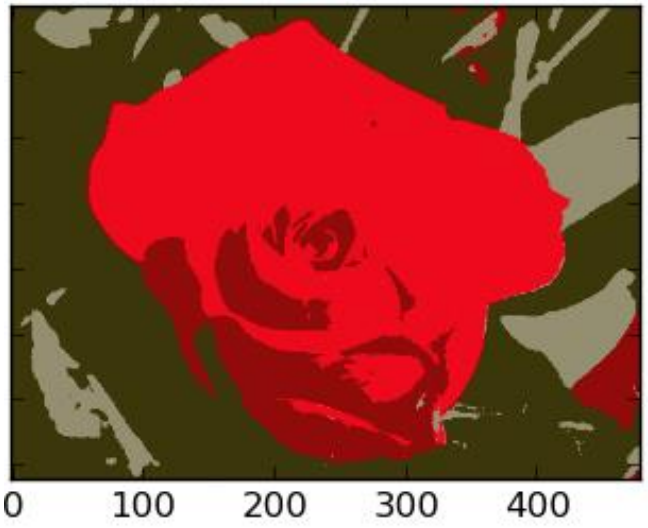
***k* = 5**

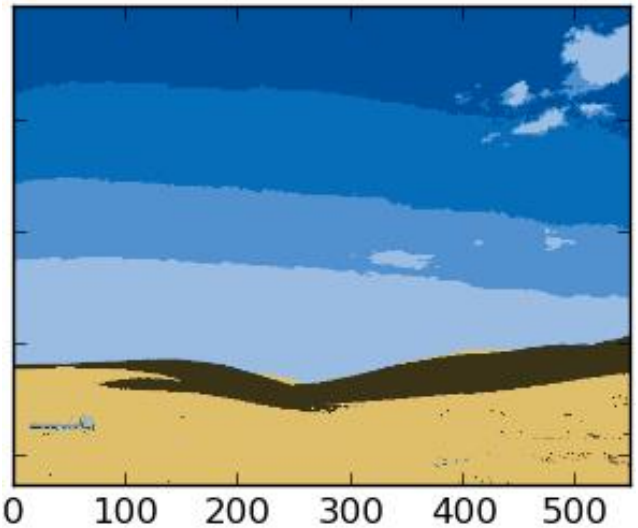


***k* = 10**

K-means clustering examples: Color Quantization

The University of Western Ontario





NOTE bias to equal-size clusters

K-means clustering examples: Adding XY features

The University of Western Ontario

color quantization

RGB features

superpixels

RGBXY features

Voronoi cells

XY features only

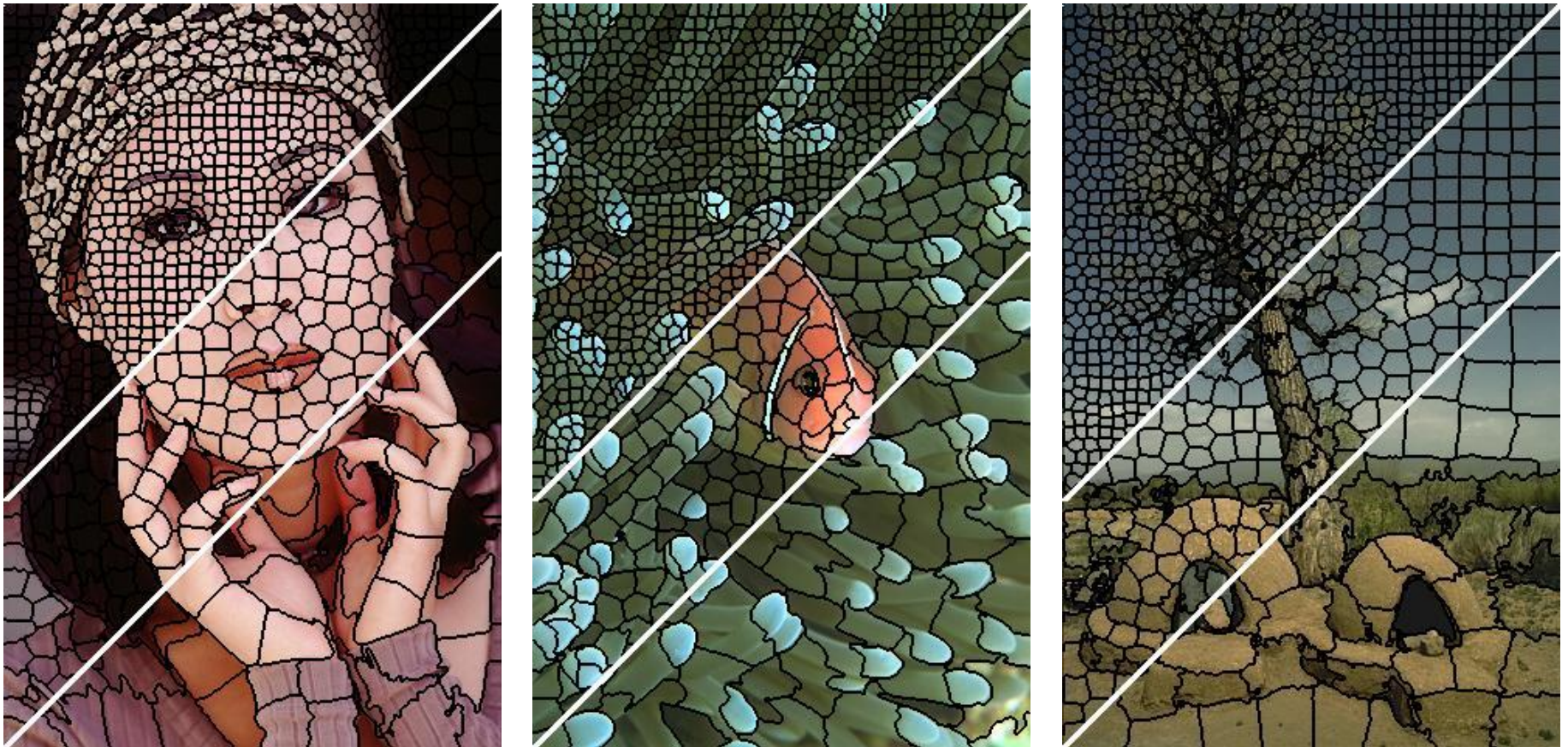
related to HW 1

K-means clustering examples: Superpixels

The University of Western Ontario

- Apply K-means to RGBXY features

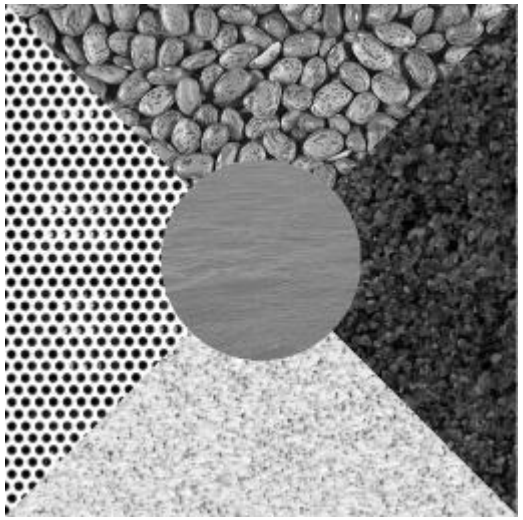
[SLIC superpixels, Achanta et al., PAMI 2011]



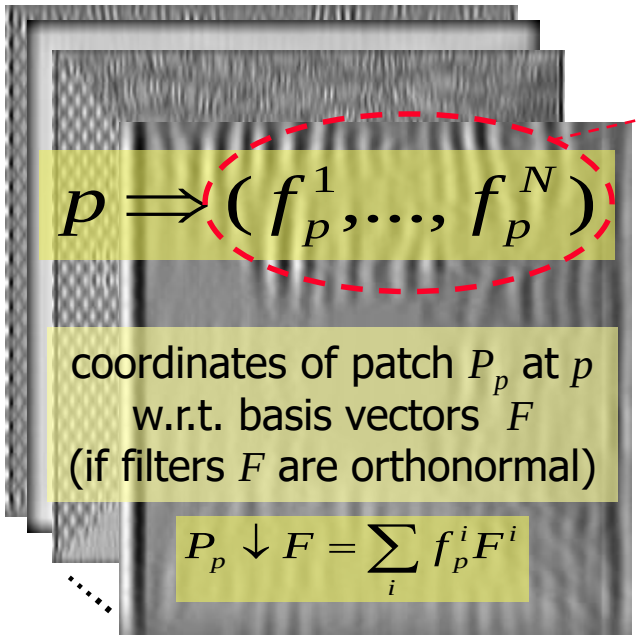
K-means clustering examples: Texture Analysis

The University of Western Ontario

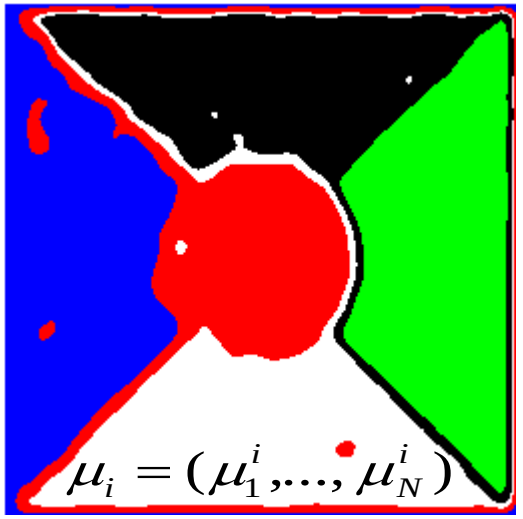
slide from Naotoshi Seo



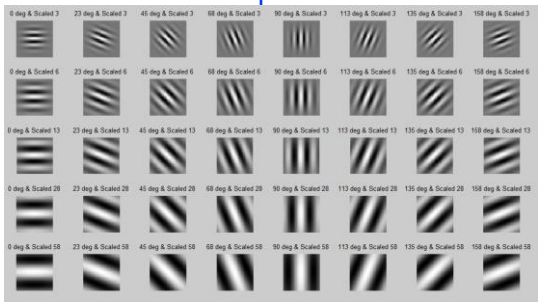
responses to N filters (convolution)



K-means
for points in R^N
instead of R^3

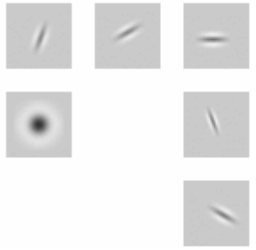


Q: what is μ_i ?



F - bank of N filters for texture, e.g. Gabor filters

texton
weighted
combination
of filters

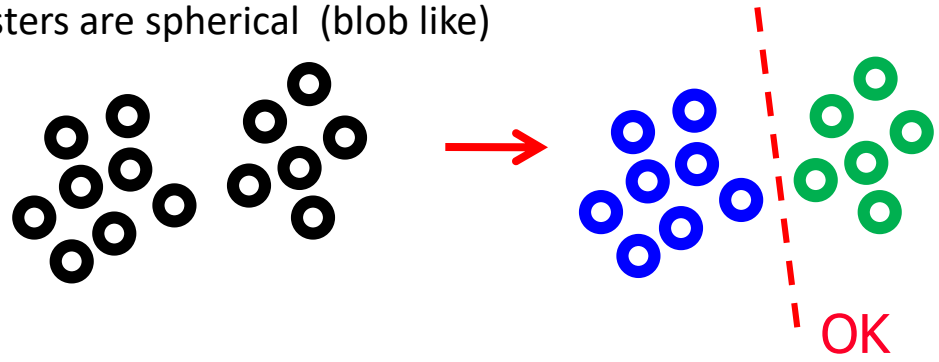


[Zhu et al. ECCV'02]

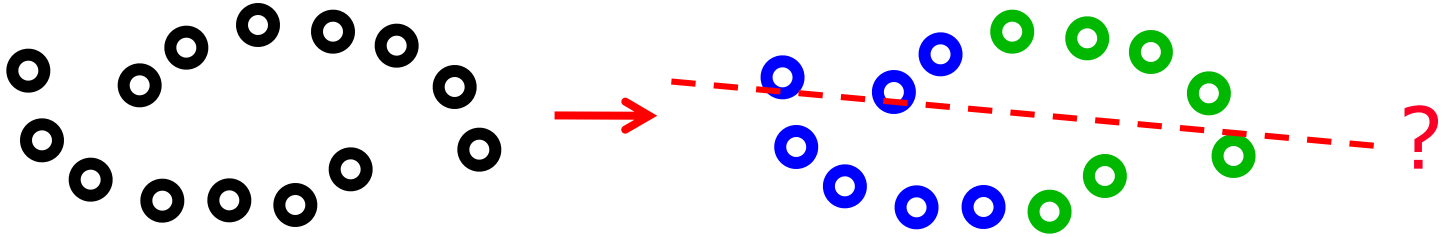
K-means Properties

The University of Western Ontario

- Works best when clusters are spherical (blob like)



- Fails for elongated non-compact clusters

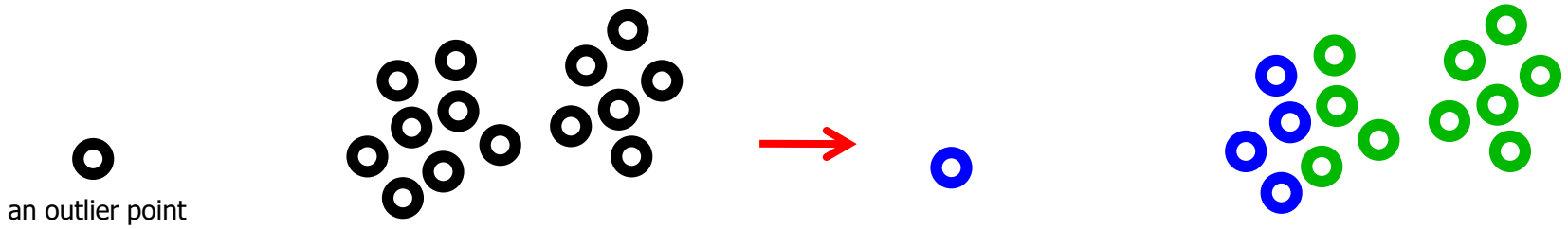


In general, K-means also does not work
when two clusters can not be separated by a line/plane
(data is linearly non-separable)

K-means Properties

The University of Western Ontario

- Sensitive to outliers

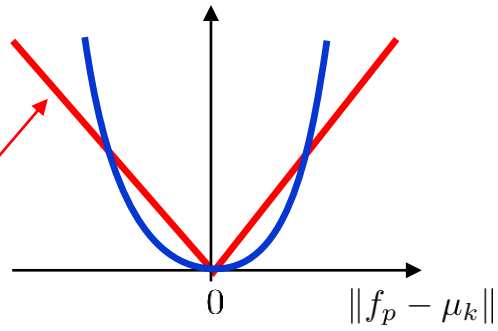


Explanation: squared distance error grows too fast making any outlier extremely costly. This also explains non-robustness of a "sample mean" statistic.

$$SSE = \sum_{k=1}^{K} \sum_{p \in S^k} \left\| f_p - \mu_k \right\|^2$$

Possible solution: replace squared distances by absolute distances that grow at a slower pace.

$$SAE = \sum_{k=1}^{K} \sum_{p \in S^k} \left\| f_p - \mu_k \right\|$$



Interestingly, in this case the optimal value of μ_k is the "median" of set S^k instead of its "mean"

$$SSE = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|^2$$ ➡ K-means

$$SAE = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|$$ ➡ K-medians

NOTE:

besides changing the error measure, there are many other generalizations of K-means that require certain **interpretations of SSE objective**

K-means as probabilistic clustering

(probabilistic model parameter fitting)

maximum likelihood (ML) fitting
of parameters μ_k (means) of Gaussian distributions

$$E_K = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|^2$$

equivalent (easy to check)

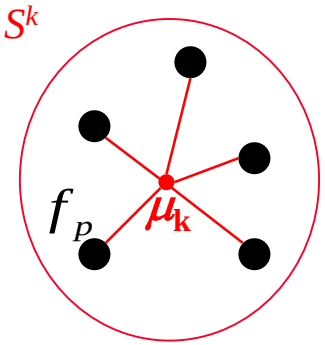
$$E_K \sim -\sum_{k=1}^{K} \sum_{p \in S^k} \log P(f_p \mid \mu_k) + const$$

for Gaussian distribution $P(x \mid \mu_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\| x - \mu_k \|^2}{2\sigma^2} \right)$

K-means as non-parametric clustering

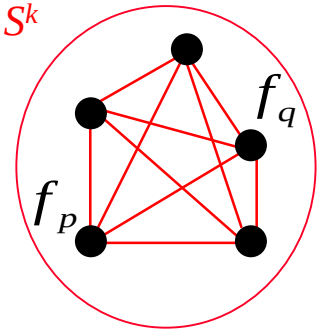
just plug-in expression

$$\mu_k = \frac{1}{|S^k|} \sum_{q \in S^k} f_q$$



$$E_K = \sum_{k=1}^{K} \sum_{p \in S^k} \| f_p - \mu_k \|^2$$

equivalent (easy to check)

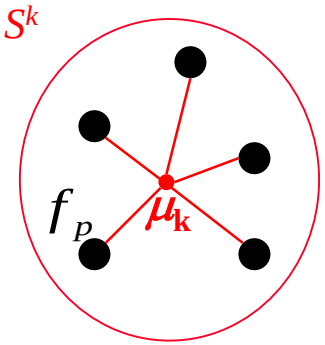


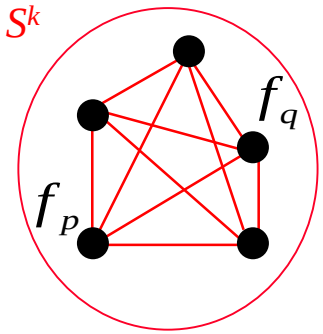
$$E_K = \sum_{k=1}^{K} \sum_{pq \in S^k} \frac{\| f_p - f_q \|^2}{2 \cdot |S^k|}$$

no parameters μ_k

sample variance: $\mathrm{var}(S^k) = \frac{1}{|S^k|} \sum_{p \in S^k} \| f_p - \mu_k \|^2 = \frac{1}{2|S^k|^2} \sum_{pq \in S^k} \| f_p - f_q \|^2$

K-means as variance clustering criteria





both formulas can be written as

$$E_K = \sum_{k=1}^{K} |S^k| \cdot \mathrm{var}(S^k)$$

sample variance: $\mathrm{var}(S^k) = \frac{1}{|S^k|} \sum_{p \in S^k} \| f_p - \mu_k \|^2 = \frac{1}{2|S^k|^2} \sum_{pq \in S^k} \| f_p - f_q \|^2$

K-means Summary

- Advantages
  - Principled (objective function) approach to clustering
  - Simple to implement (the approximate iterative optimization)
  - Fast

- Disadvantages
  - Only a local minimum is found (sensitive to initialization)
  - May fail for non-blob like clusters ← K-means fits Gaussian models
  - Sensitive to outliers ← Quadratic errors are such
  - Sensitive to choice of ***K*** ← Can add sparsity term and make K an additional variable

$$E = \sum_{k=1}^{K} \sum_{p \in S^k} \left\| f_p - \mu_k \right\|^2 + \gamma \cdot |K|$$

Akaike Information Criterion (AIC) or
Bayesian Information Criterion (BIC)

K-means – common extensions

The University of Western Ontario

- Parametric methods with arbitrary *distortion* measure $\|\cdot\|_d$ (*distortion clustering*)

$$\sum_{k=1}^{K}\sum_{p\in S^k}\left\| f_p-\mu_k\right\|_d$$

Examples of $\|\cdot\|_d$: *quadratic* (*K-means*) *absolute* (*K-medians*) *truncated* (*K-modes*)

- Parametric methods with arbitrary likelihoods $P(\cdot|\theta)$ (*probabilistic K-means*)

$$-\sum_{k=1}^{K}\sum_{p\in S^k}\log P(f_p\,|\,\theta_k)$$

Examples of $P(\cdot|\theta)$: *Gaussian, gamma, exponential, Gibbs, etc.*

- Non-parametric methods: **pair-wise clustering** with arbitrary *affinity* measure or kernel $A(x,y)$ (*kernel K-means*, *normalized cuts*, *average association*, *average distortion*)

$$\sum_{k=1}^{K}\sum_{pq\in S^k}\frac{\left\|f_p-f_q\right\|^2}{2\,|S^k|}=const-\sum_{k=1}^{K}\sum_{pq\in S^k}\frac{\langle f_p,f_q\rangle}{|S^k|}\quad\Rightarrow\quad-\sum_{k=1}^{K}\sum_{pq\in S^k}\frac{A(f_p,f_q)}{|S^k|}$$

replace dot-products with arbitrary affinity/similarity measure (or kernel) A

K-means – common extensions

The University of Western Ontario

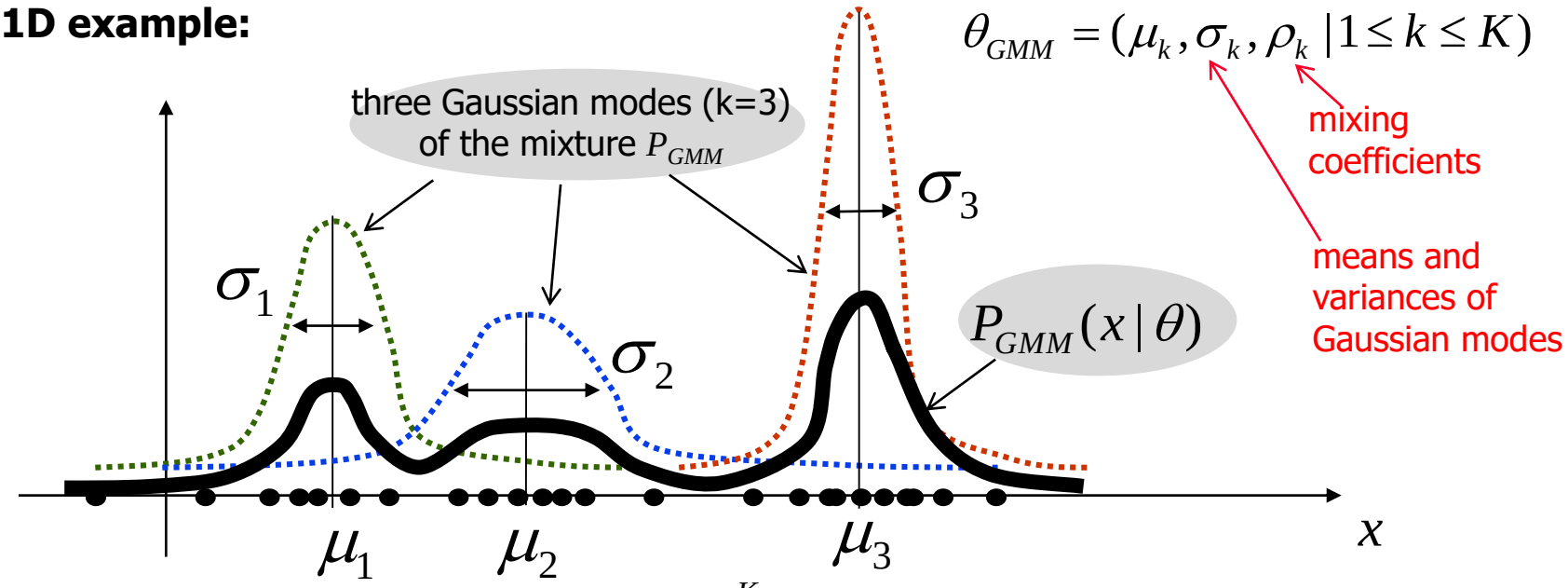
optional material

- Soft clustering using Gaussian Mixture Model (GMM)
  - no "hard" assignments of points to distinct (Gaussian) clusters S^k
  - all points are used to estimate parameters of one complex multi-modal distribution (GMM)

GMMs estimate "true" data distributions

(continuous density analog of histograms)

simple 1D example:



GMM distribution: $P_{GMM}(x \mid \theta) = \sum_{k=1}^{K} \rho_k \cdot N(x \mid \mu_k, \sigma_k)$

K-means – common extensions

The University of Western Ontario

optional material

- Soft clustering using Gaussian Mixture Model (GMM)
  - no "hard" assignments of points to distinct (Gaussian) clusters S^k
  - all points are used to estimate parameters of one complex multi-modal distribution (GMM)

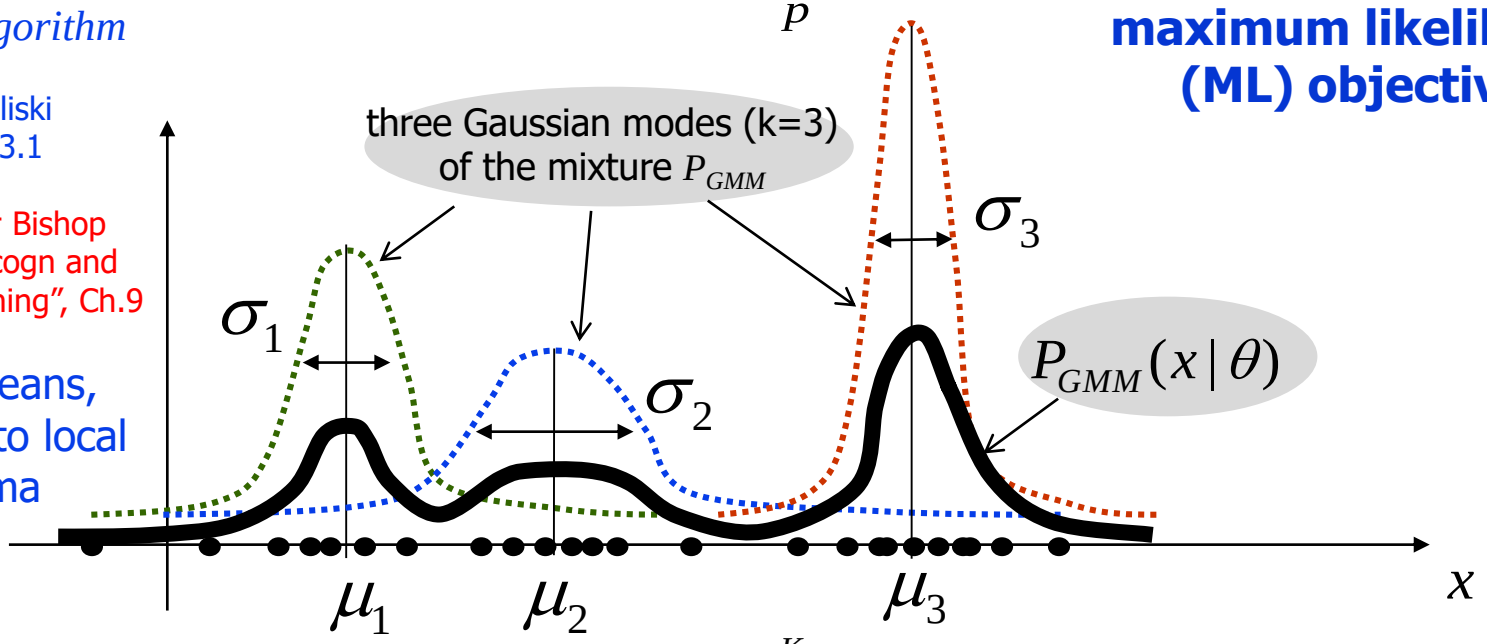
approximate optimization via *EM algorithm*

$$E_{GMM}(\theta) = -\sum_p \log P_{GMM}(f_p \mid \theta)$$

maximum likelihood (ML) objective

see Szeliski Sec. 5.3.1 or Christopher Bishop "Pattern Recogn and Machine Learning", Ch.9

like K-means, sensitive to local minima



GMM distribution: $P_{GMM}(x \mid \theta) = \sum_{k=1}^{K} \rho_k \cdot N(x \mid \mu_k, \sigma_k)$

K-means – common extensions

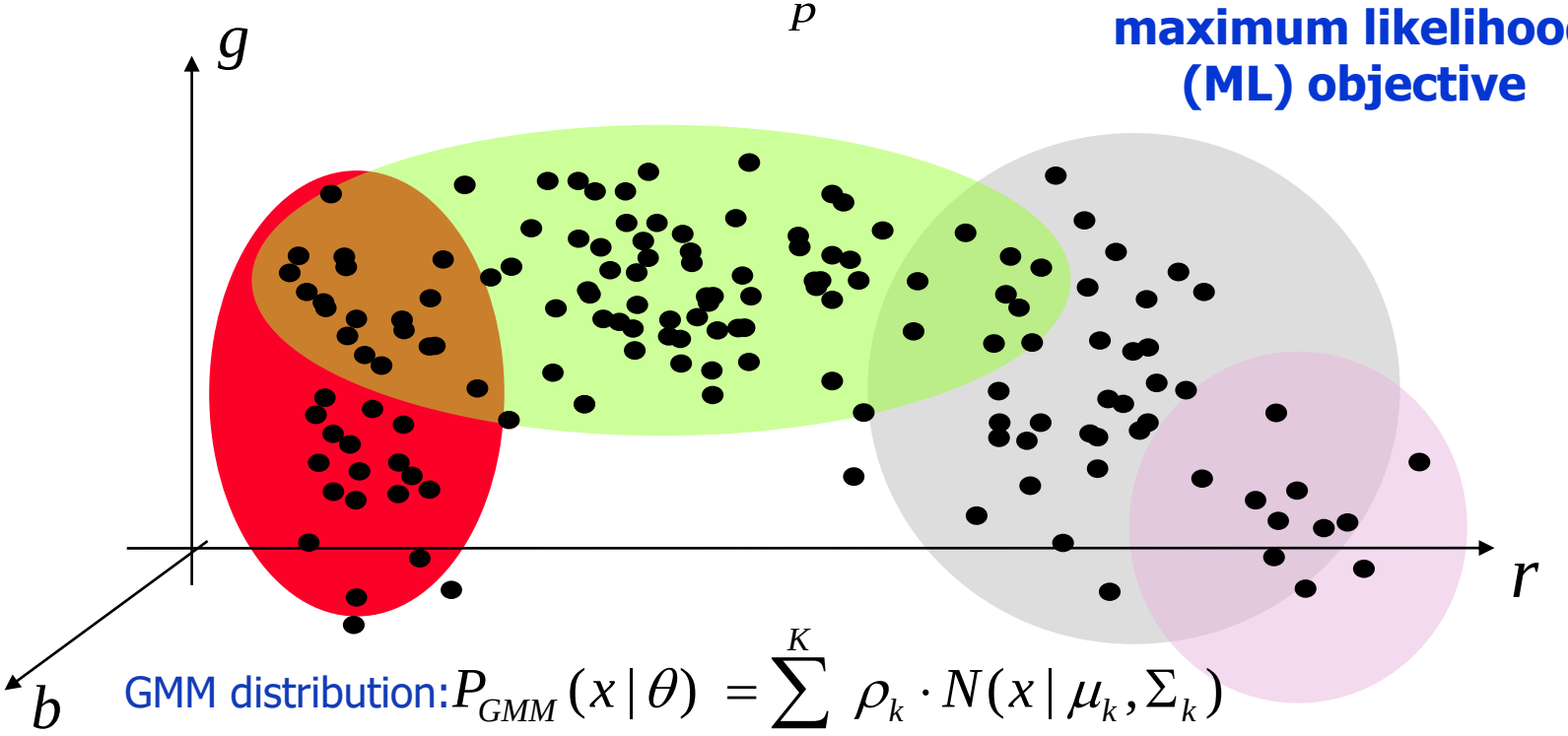
The University of Western Ontario

optional material

- Soft clustering using Gaussian Mixture Model (GMM)
  - no "hard" assignments of points to distinct (Gaussian) clusters S^k
  - all points are used to estimate parameters of one complex multi-modal distribution (GMM)

$$E_{GMM}(\theta) = -\sum_p \log P_{GMM}(f_p \mid \theta)$$

maximum likelihood (ML) objective



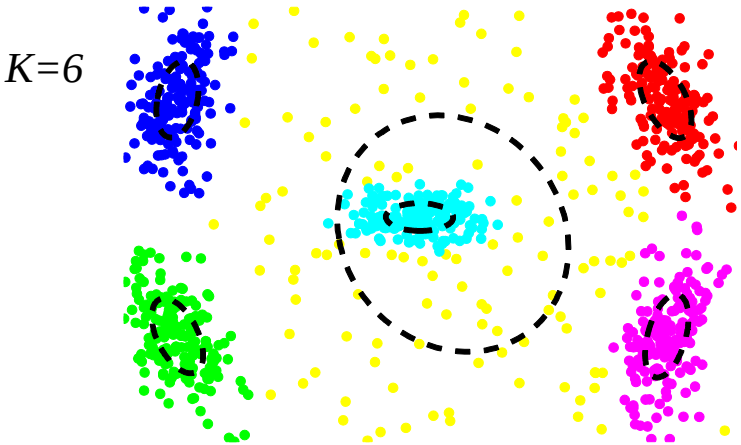
GMM distribution: $P_{GMM}(x \mid \theta) = \sum_{k=1}^{K} \rho_k \cdot N(x \mid \mu_k, \Sigma_k)$

The University of Western Ontario

Gaussian clusters/modes in: K-means vs. GMM

optional material

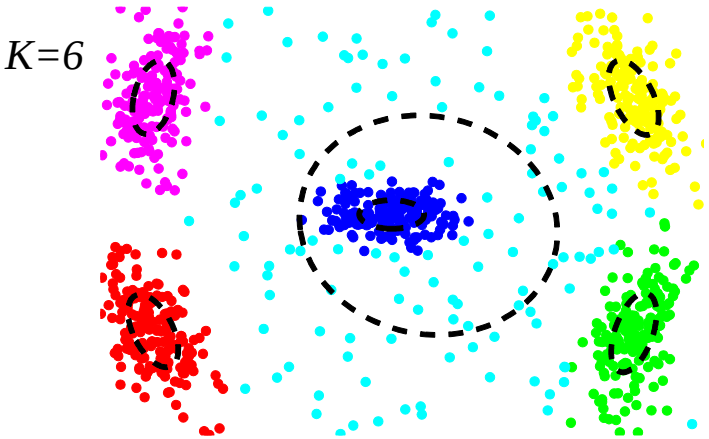
- *hard* assignment to clusters
  - separates data points into multiple Gaussian blobs
- only estimates means μ_k
  - Σ_k can also be added as a cluster parameter (*elliptic K-means*)



(elliptic) K-means

color indicates assigned cluster

- *soft* mode searching
  - estimates data distribution with multiple Gaussian modes
- estimates both mean μ_k and (co)variance Σ_k for each mode



GMM

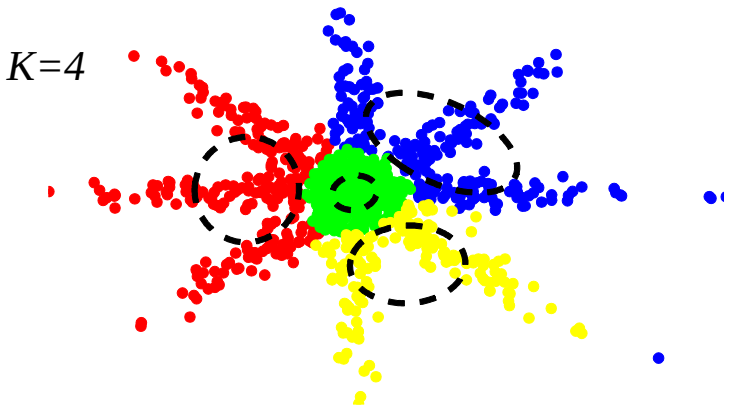
color indicates locally strongest mode

The University of Western Ontario

Gaussian clusters/modes in:

K-means vs. GMM

- *hard* assignment to clusters
  - separates data points into multiple Gaussian blobs
- only estimates means μ_k
  - Σ_k can also be added as a cluster parameter (*elliptic K-means*)

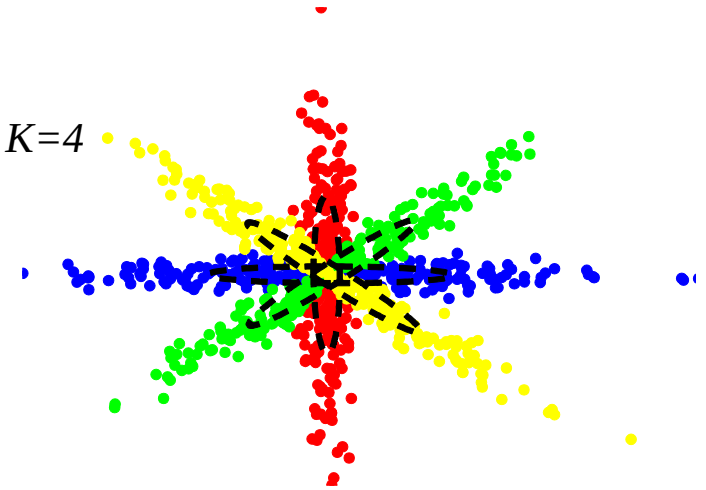


hard clustering may not work well when clusters overlap

(generally not a problem in segmentation, since objects do not "overlap" in RGBXY)

optional material

- *soft* mode searching
  - estimates data distribution with multiple Gaussian modes
- estimates both mean μ_k and (co)variance Σ_k for each mode



While this is an optimal GMM, it is hard to find for standard EM algorithm due to local minima.

Gaussian clusters/modes in: K-means vs. GMM

- *hard* assignment to clusters
  - separates data points into multiple Gaussian blobs
- only estimates means μ_k
  - Σ_k can also be added as a cluster parameter (*elliptic K-means*)
- computationally cheap

  (block-coordinate descent)
- sensitive to local minima
- **scales to higher dimensions** (*kernel K-means*)

optional material

- *soft* mode searching
  - estimates data distribution with multiple Gaussian modes
- estimates both mean μ_k and (co)variance Σ_k for each mode
- more expensive

  (EM algorithm, Szeliski Sec.5.3.1)
- sensitive to local minima
- **does not scale to higher dimensions**

The University of Western Ontario

Normalized Cut

and non-parametric (kernel) clustering

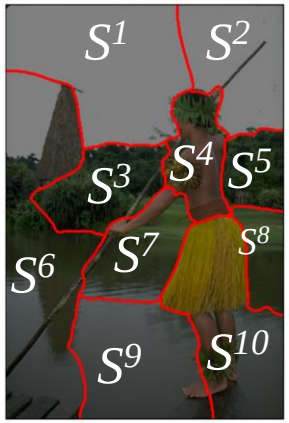
[Shi, Malik, PAMI 2000]

$$NC(S) = -\sum_{k=1}^{K} \frac{assoc(S^k, S^k)}{assoc(\Omega, S^k)}$$

Ω - set of all points (graph nodes)

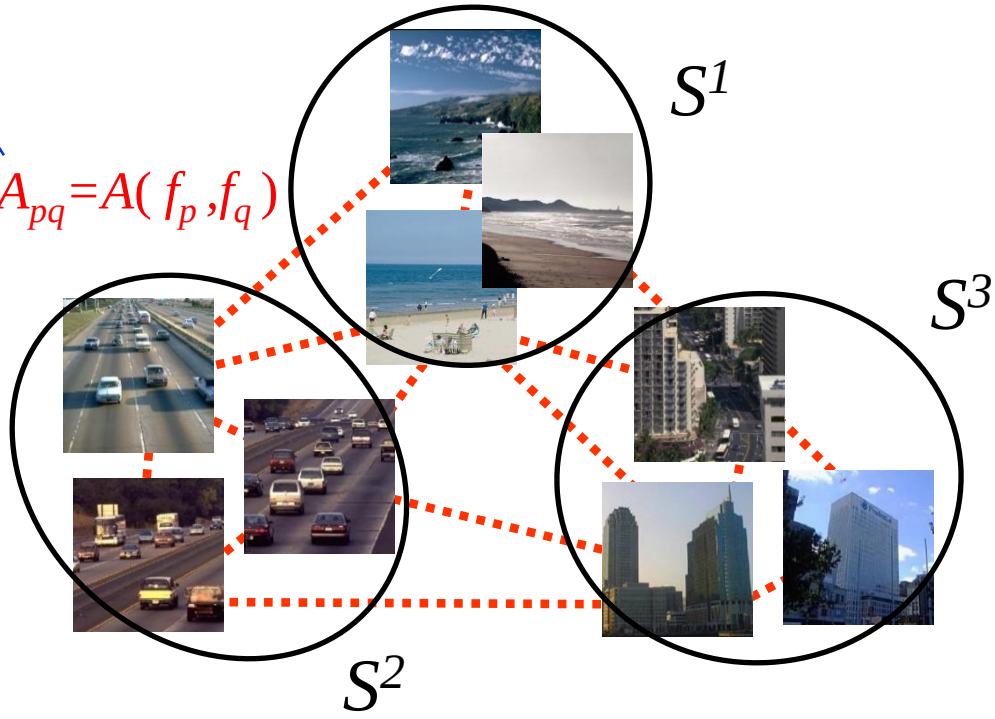
where $assoc(S^i, S^j) = \sum_{p \in S^i, q \in S^j} A_{pq}$

given pairwise affinity (kernel) matrix $A_{pq} = A(f_p, f_q)$



Typical unsupervised image segmentation example: pairwise affinities (kernel) between RGBXY image features

with Gaussian kernel: $A_{pq} = \exp -\frac{\|f_p - f_q\|^2}{2\sigma^2}$



Normalized Cut

and non-parametric (kernel) clustering

$$NC(S) = -\sum_{k=1}^{K} \frac{assoc(S^k, S^k)}{assoc(\Omega, S^k)} \equiv -\sum_{k=1}^{K} \frac{\sum_{pq \in S^k} A_{pq}}{\sum_{p \in S^k} d_p}$$

where $assoc(S^k, S^k) = \sum_{pq \in S^k} A_{pq}$

$$assoc(\Omega, S^k) = \sum_{p \in S^k} \sum_{q \in \Omega} A_{pq} = \sum_{p \in S^k} d_p$$

assuming a vector of "node degrees"

$$d_p = \sum_q A_{pq}$$

NOTE:

remember pairwise formulation of K-means

$$-\sum_{k=1}^{K} \sum_{pq \in S^k} \frac{\langle f_p, f_q \rangle}{|S^k|}$$

- special case of NC energy above for $A_{pq} = \langle f_p, f_q \rangle$ and d = **1**

$$\frac{\sum_{pq \in S^k} \langle f_p, f_q \rangle}{\sum_{p \in S^k} 1}$$

The University of Western Ontario

Normalized Cut

and non-parametric (kernel) clustering

$$NC(S) = -\sum_{k=1}^{K} \frac{assoc(S^k, S^k)}{assoc(\Omega, S^k)} \equiv -\sum_{k=1}^{K} \frac{S^{k\prime} A S^k}{d' S^k}$$

where $assoc(S^k, S^k) = \sum_{pq \in S^k} A_{pq} \equiv S^{k\prime} A S^k$

using **matrix notation**
where S^k are indicator vectors,
` stands for transpose, and
$A=[A_{pq}]$ is affinity matrix.

$$assoc(\Omega, S^k) = \sum_{p \in S^k} d_p \equiv d' S^k$$

assuming a vector of "node degrees"

$$d_p = \sum_q A_{pq}$$

NOTE:

remember pairwise formulation of K-means

$$-\sum_{k=1}^{K} \sum_{pq \in S^k} \frac{\langle f_p, f_q \rangle}{|S^k|}$$

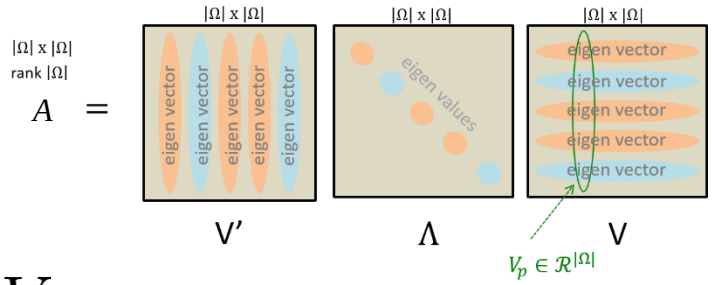
- special case of NC energy above for $\mathcal{E}_{pq} = \langle f_p, f_q \rangle$ and d = **1**

$$\frac{S^{k\prime} \mathcal{E} S^k}{1' S^k}$$

Is the general NC objective above related to K-means?

Normalized Cut and K-means

$$NC(S) = -\sum_{k=1}^{K} \frac{S^{k\prime} A S^{k}}{d' S^{k}}$$



Now, consider *eigen decomposition* $A = V'\Lambda V$

and some new "feature points" defined as
(assuming A is pos.def., that is, positive eigen values Λ)

$$\phi_p := \sqrt{\Lambda} V_p \quad \in \mathcal{R}^{|\Omega|}$$

embedded in very high-dimensional space

Obviously, we get $A_{pq} = \phi_p{}' \phi_q \equiv \langle \phi_p, \phi_q \rangle$

Normalized Cut and K-means

$$NC(S) = -\sum_{k=1}^{K} \frac{S^{k\prime} A S^{k}}{d' S^{k}} \overset{c}{=} \sum_{k=1}^{K} \sum_{p \in S^k} d_p \|\phi_p - \mu^k\|^2$$

Easy to check equality (up to constant) for

$$\mu^k := \overset{\text{weighted}}{mean}\{\phi_p \,|\, p \in S^k\} \equiv \frac{\sum_{p \in S^k} d_p \phi_p}{d' S^k}$$

$$A_{pq} = \phi_p{}' \phi_q \equiv \langle \phi_p, \phi_q \rangle$$

Normalized Cut and K-means

$$NC(S) = -\sum_{k=1}^{K} \frac{S^{k\prime} A S^{k}}{d' S^{k}} \quad \stackrel{c}{=} \quad \sum_{k=1}^{K} \sum_{p \in S^k} d_p \|\phi_p - \mu^k\|^2$$

(weighted) K-means energy

[Bach & Jordan 2003, Dhillon et al., 2004]

$$\mu^k := \stackrel{weighted}{mean}\{\phi_p \,|\, p \in S^k\} \equiv \frac{\sum_{p \in S^k} d_p \phi_p}{d' S^k}$$

for $\phi_p := \sqrt{\Lambda} V_p \in \mathcal{R}^{|\Omega|}$

points embedded in very high-dimensional space
(could be expensive and more sensitive to local minima)

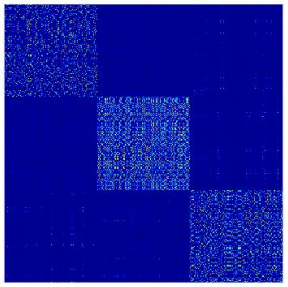
It is possible to approximate via K-means in low-dimensional space (next)

Technical notes from linear algebra: low-rank matrix approximation

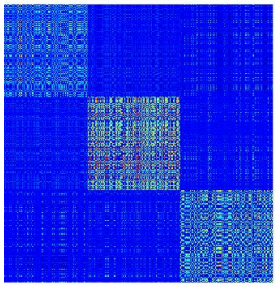
The University of Western Ontario

Consider rank m approximation $\widetilde{A} \approx A$
minimizing Frobenious norm (errors):

$$\|A - \widetilde{A}\|_F := \sum_{pq} (A_{pq} - \widetilde{A}_{pq})^2$$



A



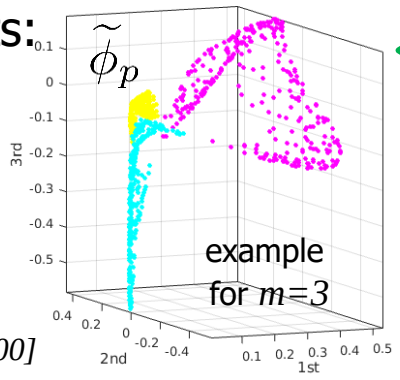
$\widetilde{A}$

Standard solution:
uses top m eigenvectors and eigenvalues of A :

$$\widetilde{A} = V^{m\prime} \Lambda^m V^m$$

Consider **low-dimensional** points:

$$\widetilde{\phi}_p := \sqrt{\Lambda^m} V_p^m \in \mathcal{R}^m$$



isometric low-dimensional Euclidean Embedding

$$A_{pq} \approx \widetilde{A}_{pq} = \langle \widetilde{\phi}_p, \widetilde{\phi}_q \rangle$$

multi-dimensional scaling (MDS) *[Cox & Cox., 2000]*
kernel PCA *[Schölkopf, Smola, Müller, 1998]*

Normalized Cut and K-means

NC $\widetilde{A} \approx A$

low-dimensional weighted K-means

$$\sum_k -\frac{S^{k'} A S^k}{d' S^k} \approx \sum_k -\frac{S^{k'} \widetilde{A} S^k}{d' S^k} \stackrel{c}{=} \sum_k \sum_{p \in S^k} d_p \|\widetilde{\phi}_p - \mu^k\|^2$$

As earlier, easy to check equality (up to constant) for

$$\mu^k := \overset{weighted}{mean}\{\widetilde{\phi}_p \,|\, p \in S^k\}$$

$$\widetilde{A}_{pq} = < \widetilde{\phi}_p, \widetilde{\phi}_q >$$

for $\phi_p := \sqrt{\Lambda^m} V_p^m \in \mathcal{R}^m$

points embedded in low-dimensional space

NC is commonly approximated by **K-means** for some low-dimensional embedding $\widetilde{\phi}_p \in \mathcal{R}^m$ for $m \ll |\Omega|$ based on top m eigen values/vectors for affinity matrix A.

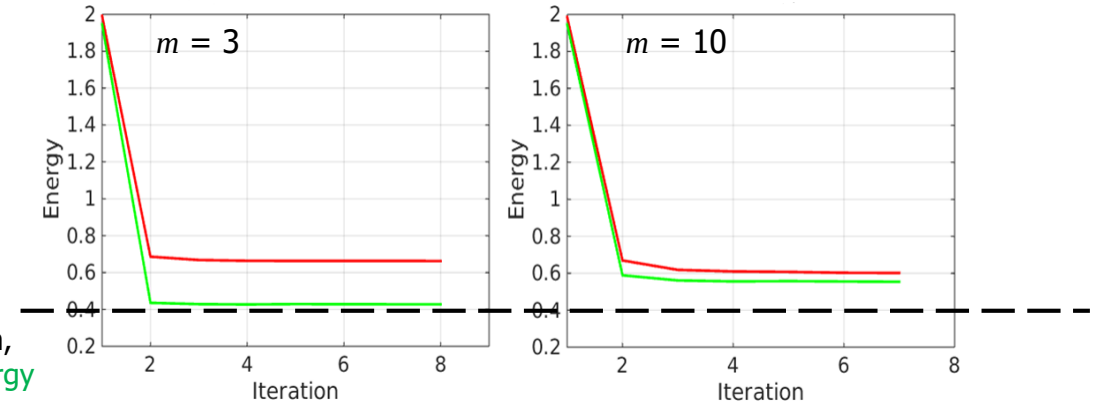
- discretization heuristic for spectral relaxation [Shi&Malik 2000]

- Frobenius-norm approximation [Tang et al., 2016]

The University of Western Ontario

Normalized Cut and K-means

Convergence of iterative K-means procedure for different m



[Tang et al., 2016]

NOTE:
approximation gap is larger for smaller m, but solution's NC energy could be lower

NC energy $\sum_k -\frac{S^{k\prime} A S^k}{d' S^k}$

K-means energy $\sum_k \sum_{p \in S^k} d_p \|\widetilde{\phi}_p - \mu^k\|^2$

for $\phi_p := \sqrt{\Lambda^m} V_p^m \in \mathcal{R}^m$

points embedded in low-dimensional space

Typical iterative K-means algorithm for NC :

- compute top m eigen values and vectors for A
- compute points above and run (weighted) K-means

- discretization heuristic for spectral relaxation [Shi&Malik 2000]
- Frobenius-norm approximation [Tang et al., 2016]

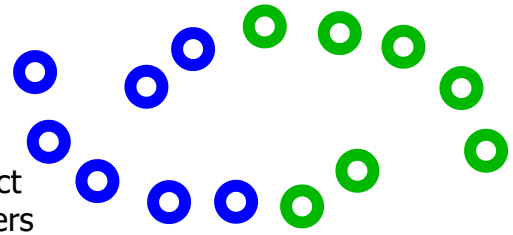
Basic K-means vs Kernel Clustering

Basic K-means

$$-\sum_k \frac{S^{k\prime}\,\mathcal{E}\,S^k}{\mathbf{1}'S^k}$$

for **dot product affinities**

$$\mathcal{E}_{pq} = \langle f_p, f_q \rangle$$



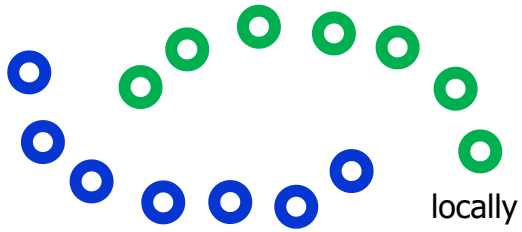
globally compact
(blob-like) clusters

Normalized Cut

$$-\sum_k \frac{S^{k\prime}\,A\,S^k}{d'S^k}$$

e.g. for **Gaussian affinities/kernel**

$$A_{pq} = \exp -\frac{\|f_p - f_q\|^2}{2\sigma^2}$$



locally coherent clusters
(do not have to be blobs)

NOTE: NC is K-means over points $\widetilde{\phi}_p \neq f_p$ s.t.

$$\langle \widetilde{\phi}_p, \widetilde{\phi}_q \rangle \approx A_{pq} \neq \mathcal{E}_{pq} = \langle f_p, f_q \rangle$$

The University of Western Ontario

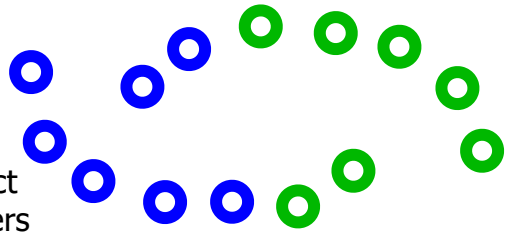
Basic K-means vs Kernel Clustering

Basic K-means

$$-\sum_k \frac{S^{k\prime}\,\mathcal{E}\,S^k}{\mathbf{1}'\,S^k}$$

for **dot product affinities**

$$\mathcal{E}_{pq} = \langle f_p, f_q \rangle$$



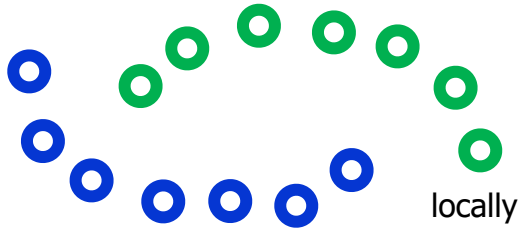
globally compact (blob-like) clusters

Normalized Cut

$$-\sum_k \frac{S^{k\prime}\,A\,S^k}{d'\,S^k}$$

e.g. for **Gaussian affinities/kernel**

$$A_{pq} = \exp -\frac{\|f_p - f_q\|^2}{2\sigma^2}$$



locally coherent clusters (do not have to be blobs)

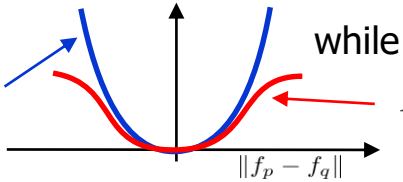
NOTE: an intuitive interpretation of affinity measures:

Any affinities A_{pq} imply "equivalent" distortions (dis-similarities) $D^A_{pq} := A_{pp} + A_{qq} - 2A_{pq}$

dot-product affinity implies **squared errors**

$$\mathcal{E}_{pp} + \mathcal{E}_{qq} - 2\mathcal{E}_{pq} = \langle f_p - f_q, f_p - f_q \rangle \equiv \|f_p - f_q\|^2$$

distant points can not be grouped



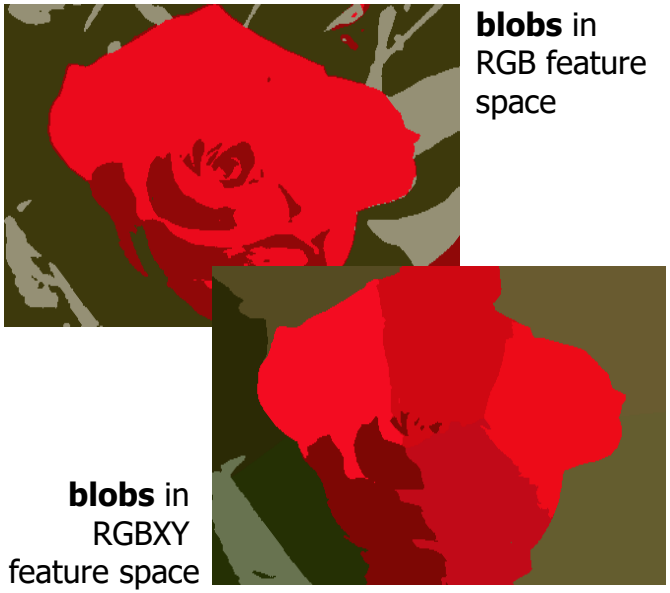
while Gaussian kernel implies **"robust" errors**

$$A_{pp} + A_{qq} - 2A_{pq} = 2 - 2\exp -\frac{\|f_p - f_q\|^2}{2\sigma^2}$$

"indifferent" to very large errors

Basic K-means vs Kernel Clustering

Basic K-means

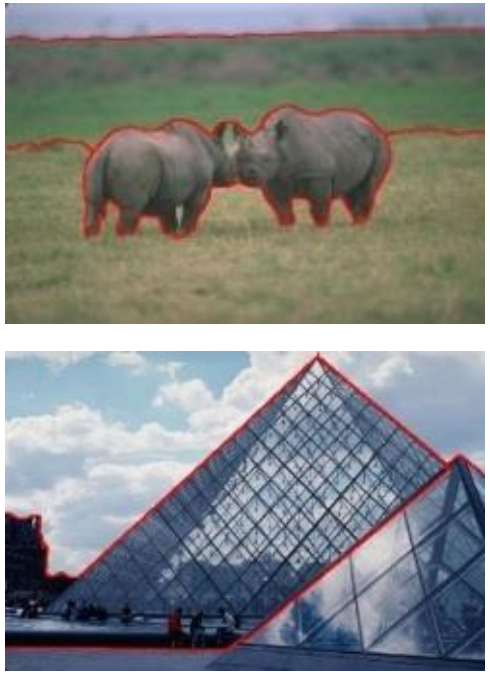


blobs in RGB feature space

blobs in RGBXY feature space

Color quantization, super-pixels

Normalized Cut



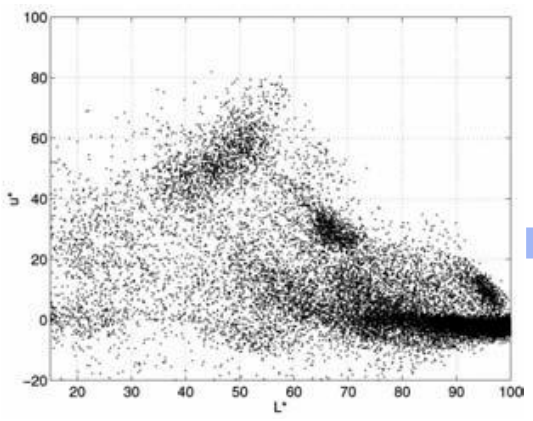
coherent segments in RGBXY feature space **do not have to look like blobs** !

Often used as "advanced" (albeit more expensive) **unsupervised image segmentation** in general applications

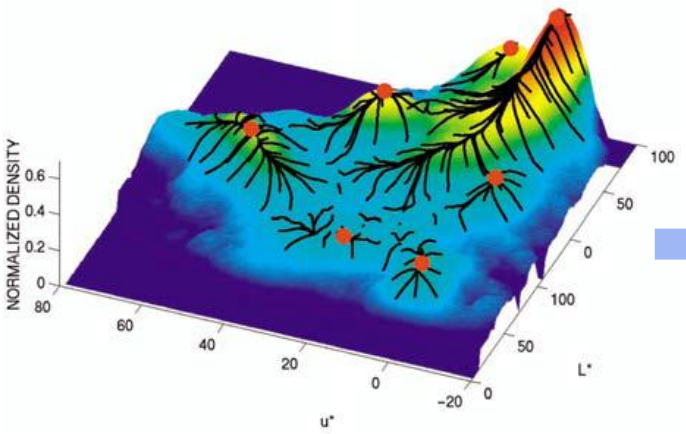
A simple non-parametric alternative: *mean-shift* clustering

The University of Western Ontario

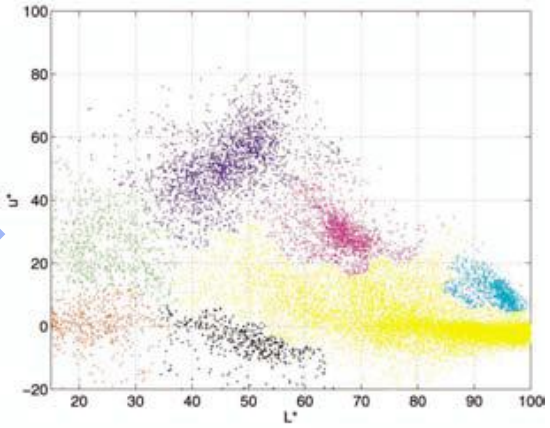
- Formulates clustering as *histogram partitioning*
  - Also looks for **modes** in data histograms
  - Does not assume the number of clusters known



data points

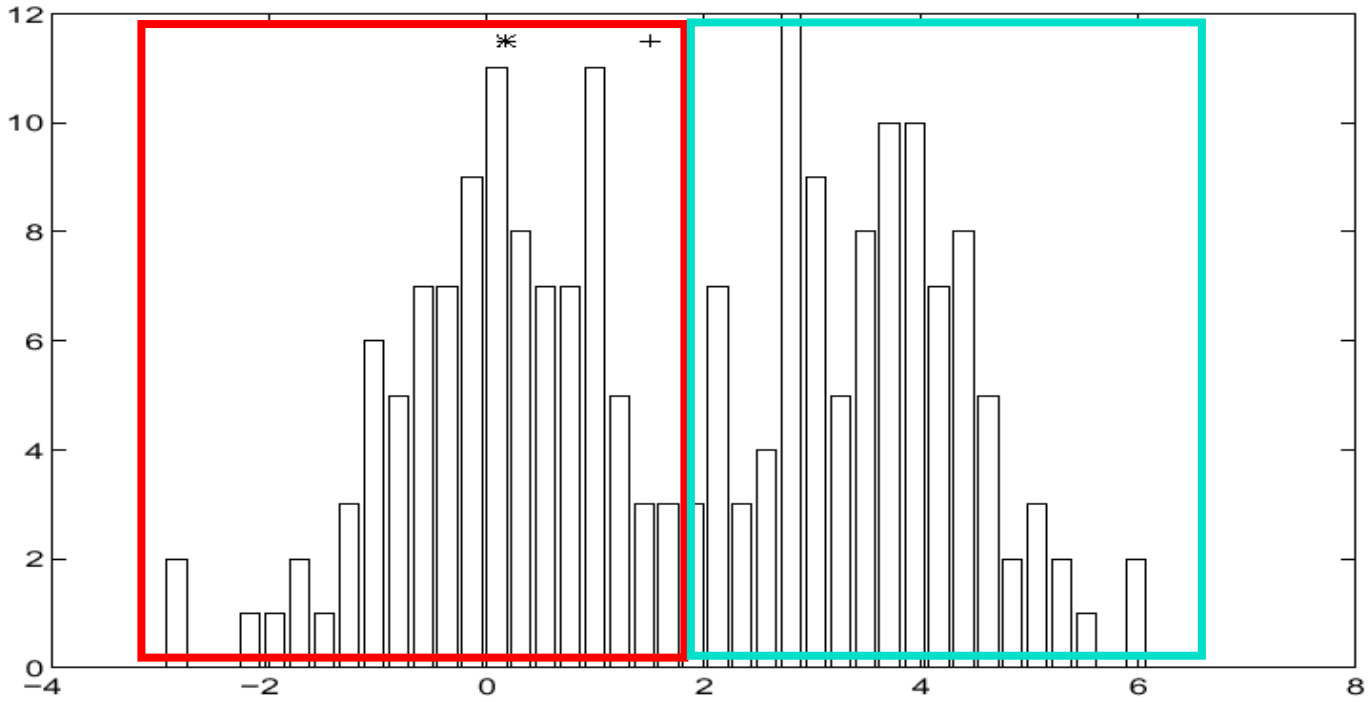


data histogram and its modes



clustering

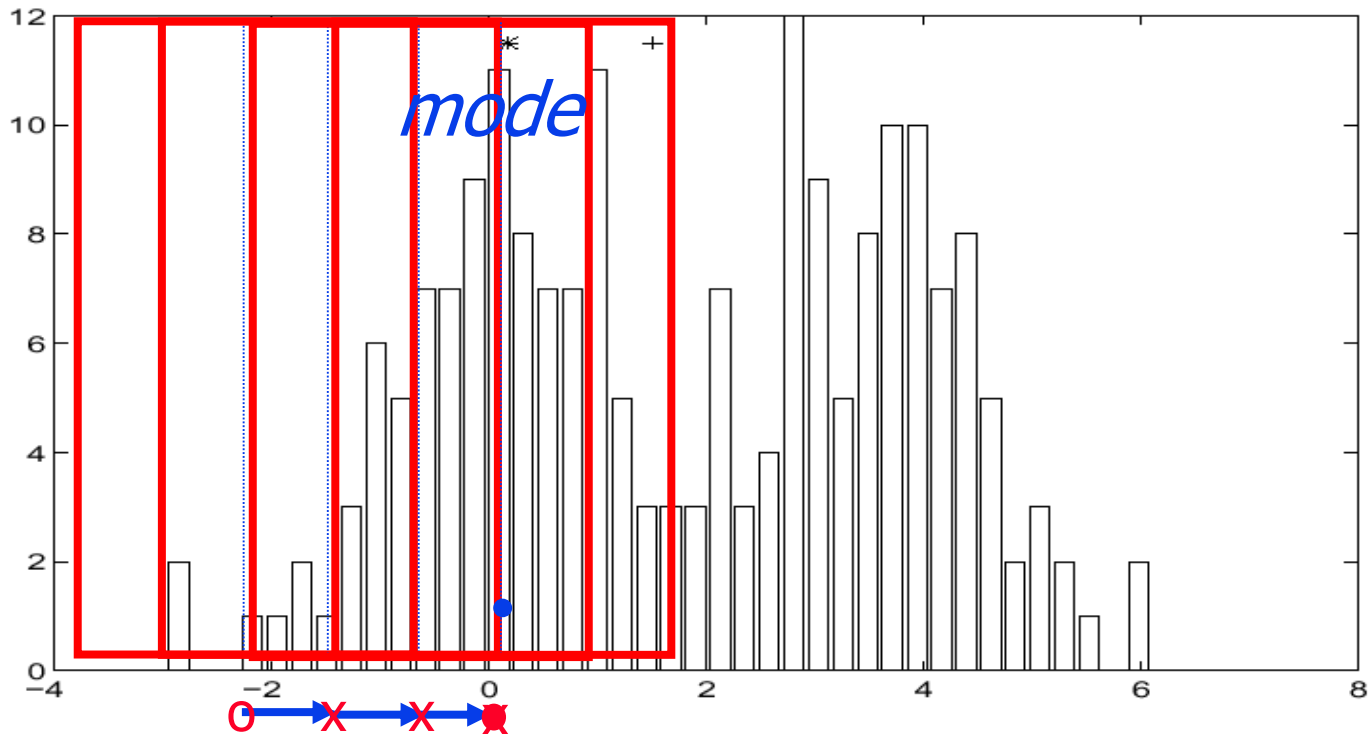
Finding Modes in a Histogram



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

Mean Shift

[Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu & Meer 2002]



Iterative Mode Search

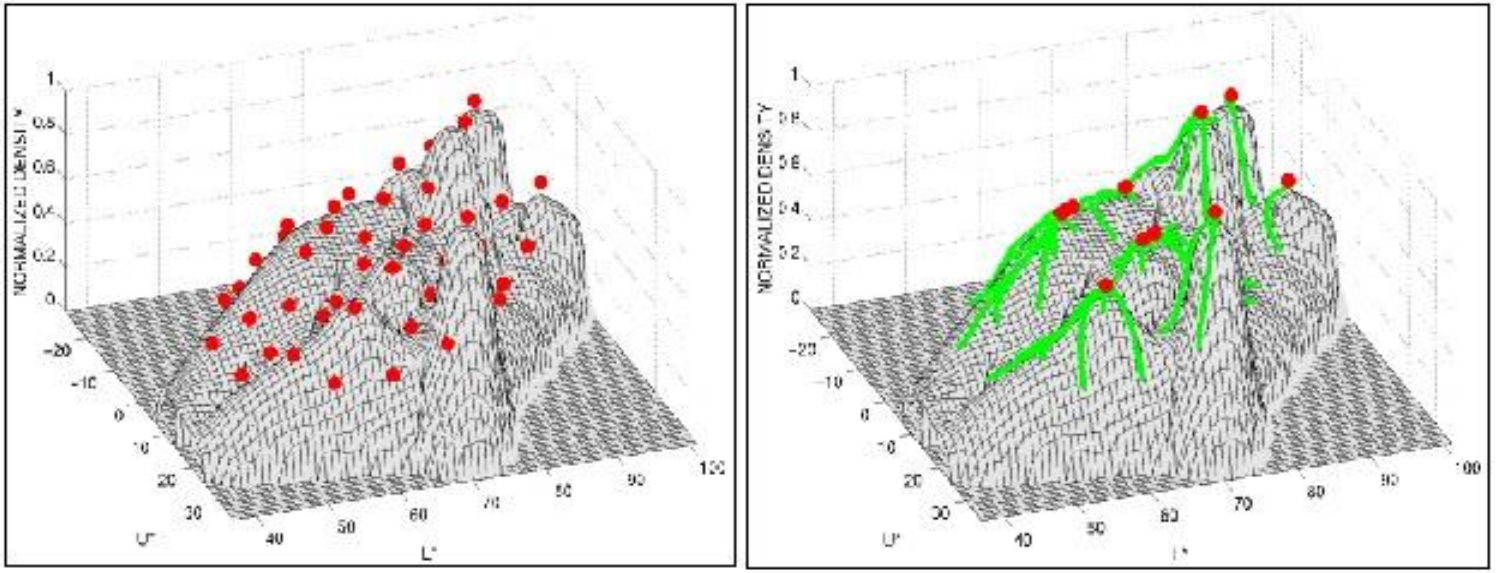
1. Initialize random seed, and fixed window
2. Calculate center of gravity ‘x’ of the window (the“mean”)
3. Translate the search window to the mean
4. Repeat Step 2 until convergence

Mean Shift

[Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu & Meer 2002]

Multimodal Distributions

- Parallel processing of an initial tessellation.
- Pruning of mode candidates.
- Classification based on the basin of attraction.



Mean shift trajectories

Mean-shift results for segmentation

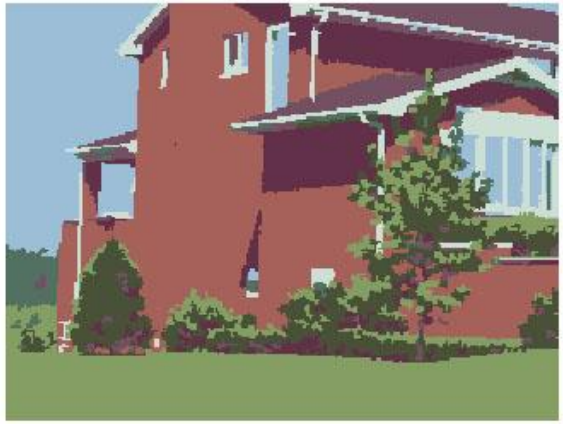
The University of Western Ontario

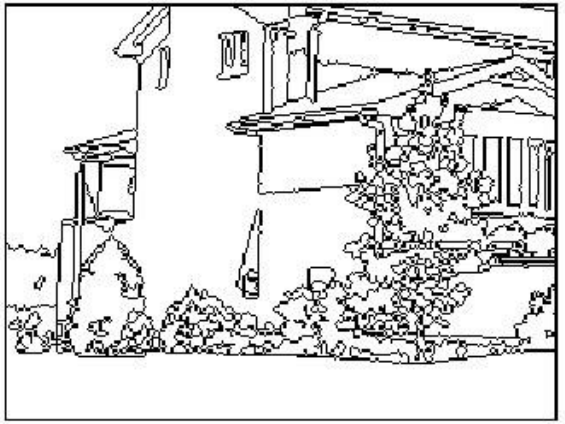
RGB+XY clustering
[Comaniciu & Meer 2002]

5D features

adding XY helps
"coherence"
in the image domain

Figure 2: The *house* image, 255 × 192 pixels, 9603 colors.



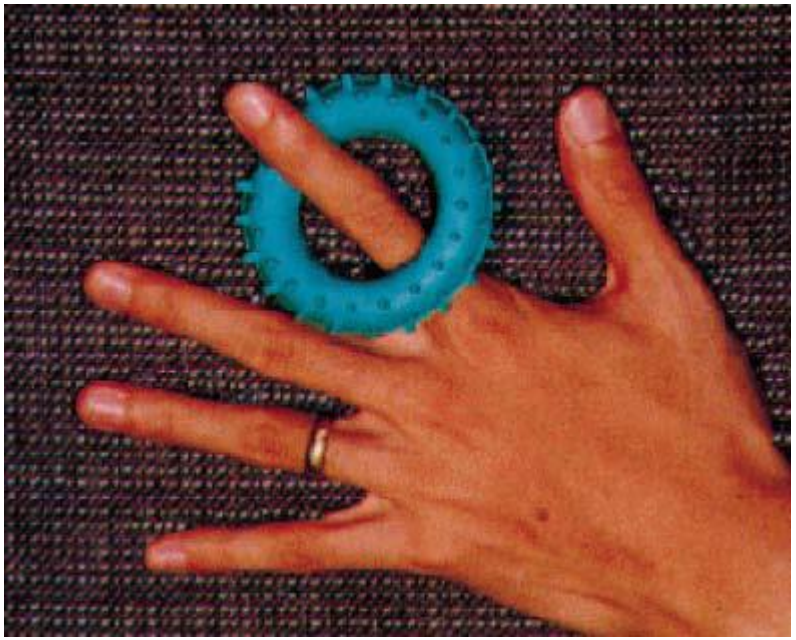


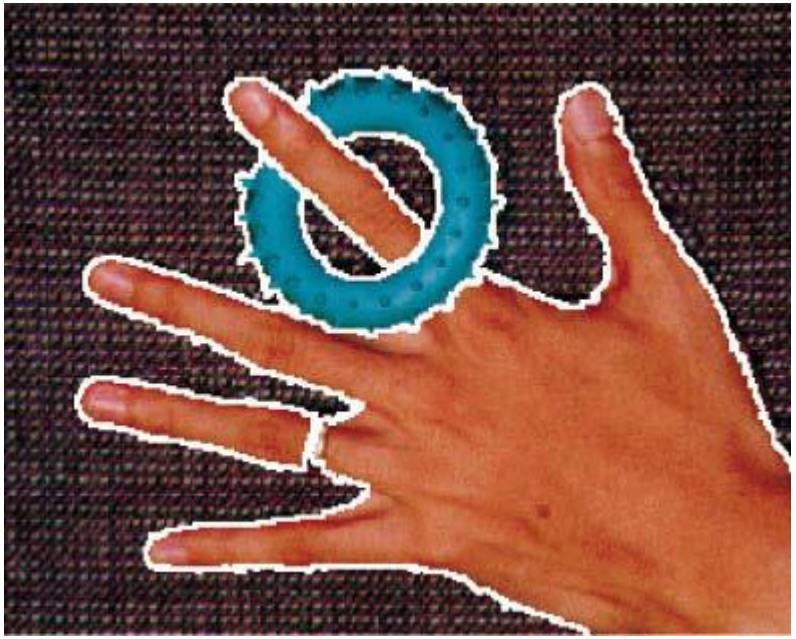
Mean-shift results for segmentation

The University of Western Ontario

RGB+XY clustering

[Comaniciu & Meer 2002]

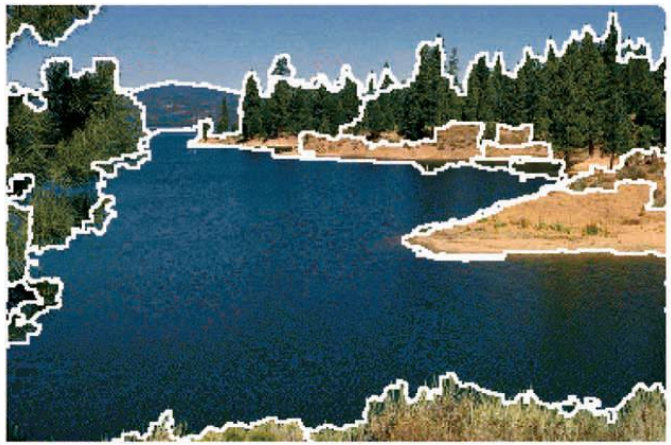




Mean-shift results for segmentation

The University of Western Ontario

RGB+XY clustering
[Comaniciu & Meer 2002]



works well for color quantization

Issues:

- Window size (*kernel bandwidth*) selection is critical
  - can not be too small or too large
  - indirectly controls the number of clusters (*k*)
  - different width in RGB and XY parts of the space
- Color may not be sufficient

  (e.g. color overlap between object and background)
- Integrating detailed boundary cues

  more on this later
  - contrast edges
  - explicit shape priors (smoothness, curvature, convexity, atlas)

Brief summary

Clustering
Partitioning
Segmentation

Q: differences between these terms?

Not really.

But, *segmentation* often implies *image features* (pixels) and importance of geometric component (XY), e.g. shape.