Nonparametric Clustering of High Dimensional Data

Peter Meer

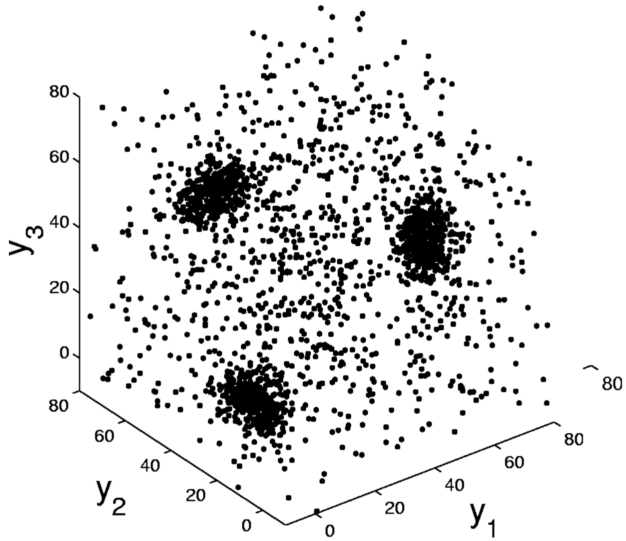
Electrical and Computer Engineering Department

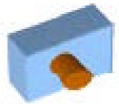
Rutgers University

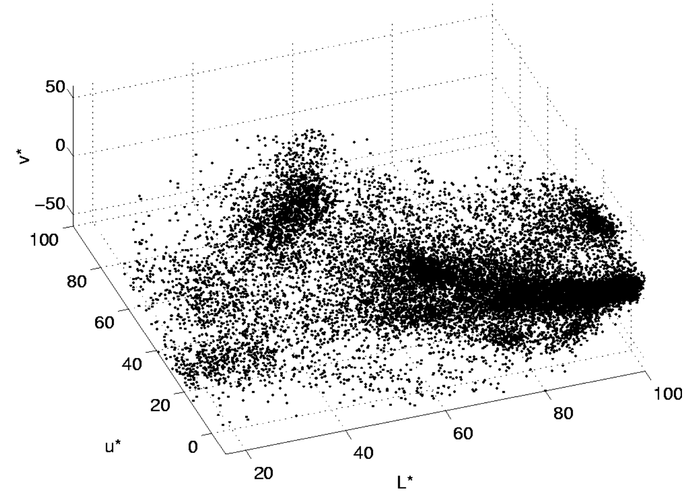
Joint work with

Bogdan Georgescu and Ilan Shimshoni

Robust Parameter Estimation: Location Problems







elliptical	cluster shape	arbitrary
known	number of clusters	estimated
not needed	adaptive techniques	needed

Kernel Density Estimation

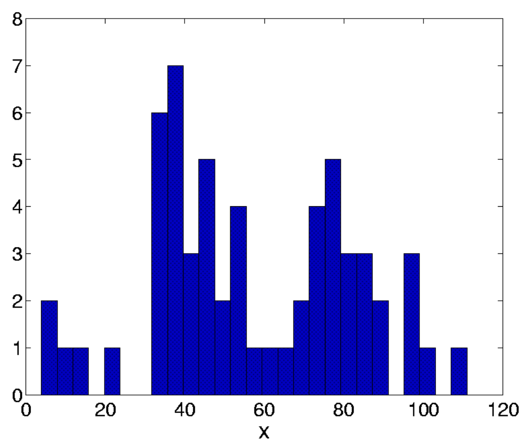
kernel density estimation (Parzen window method)

univariate data $x_i \qquad i = 1, \ldots, n$

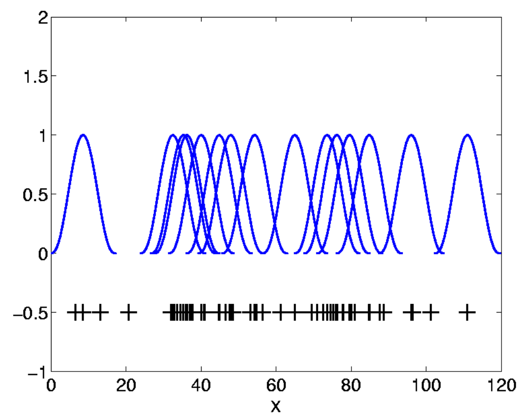
p.d.f. estimate $\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$

$K(u)$ kernel function

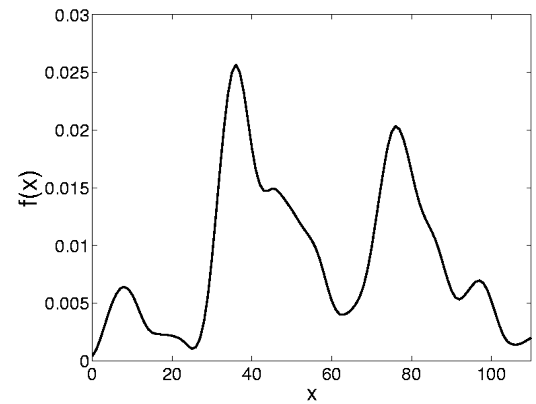
bandwidth h controls resolution $\hat{h} = n^{-1/5} \operatorname{med}_i \mid x_i - \operatorname{med}_j x_j \mid$



data histogram



a few kernels…



kernel density estimate

Mean Shift Property

given $\boldsymbol{x}_i \in \mathcal{R}^p \qquad i = 1, \cdots, n$

choose radially symmetric kernel $K(\mathbf{u}) = c_{k,p}\, k(\mathbf{u}^\top \mathbf{u})$

$k(\mathbf{u})$ profile

define $g(\mathbf{u}) = -k'(\mathbf{u})$ and the kernel $G(\mathbf{u}) = c_{g,p}\, g(\mathbf{u}^\top \mathbf{u})$

kernel density estimate with kernel $K(G)$

$$\hat{f}_{h,K}(\boldsymbol{x}) = \frac{c_{k,p}}{nh^p} \sum_{i=1}^{n} k\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right)$$

the density gradient estimate

$$\hat{\nabla} f_{h,K}(\boldsymbol{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right)} - \boldsymbol{x} \right]$$

Mean Shift Property (2)

the mean shift vector

$$\mathbf{m}_{h,G}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)} - \boldsymbol{x} = \frac{1}{2}h^2 c \frac{\hat{\nabla} f_{h,K}(\boldsymbol{x})}{\hat{f}_{h,G}(\boldsymbol{x})}$$

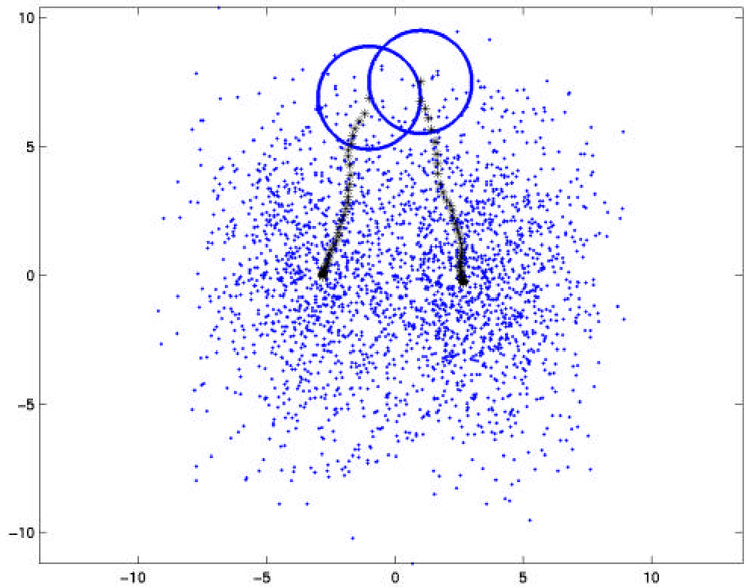
iterative computation of

$$\boldsymbol{x} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right)}$$

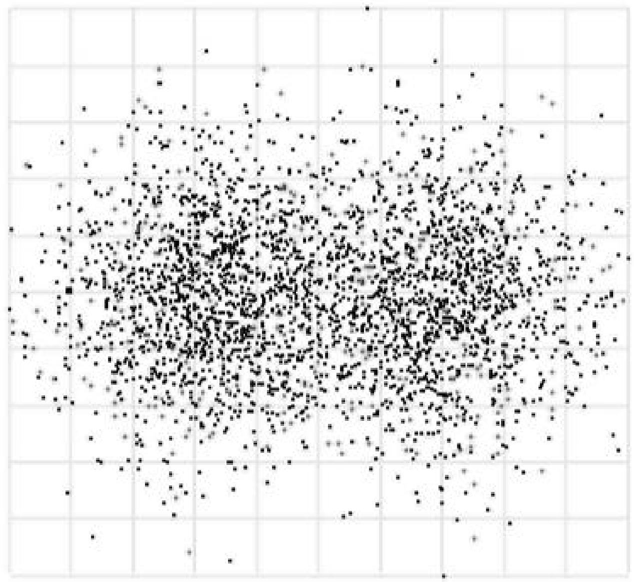
solves $\hat{\nabla} f_{h,K}(\mathbf{x}) = 0$

detects modes (stationary points) of the distribution

[Fukunaga and Hostetler, 1975]

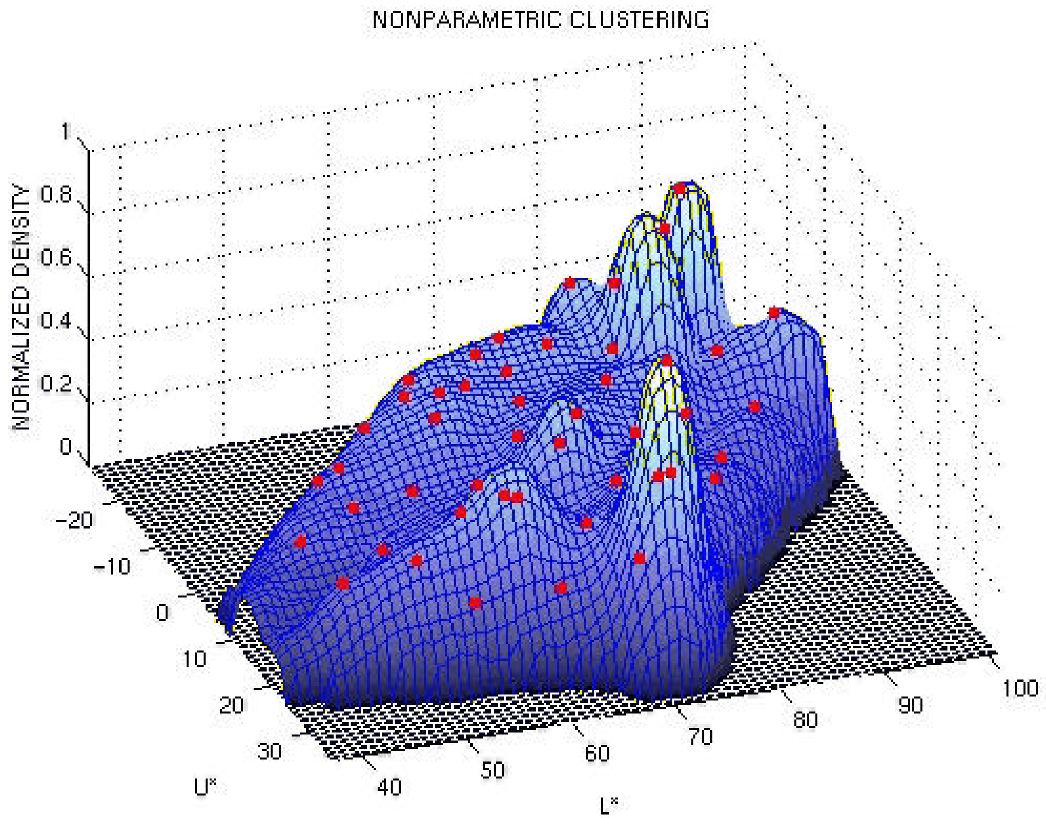


Mean Shift Based Data Analysis



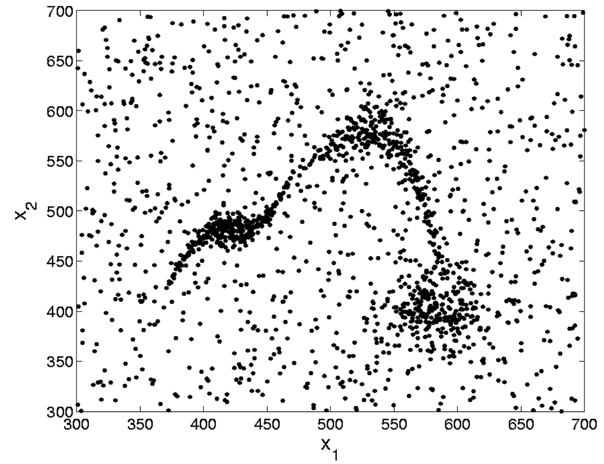
data

Mean Shift Based Data Analysis (2)

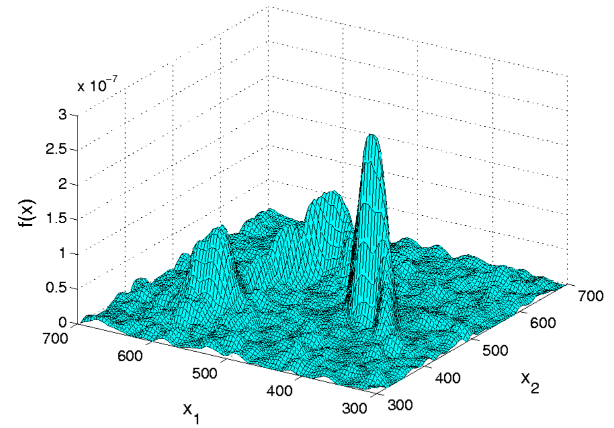


Nonparametric Clustering

data



density



clusters

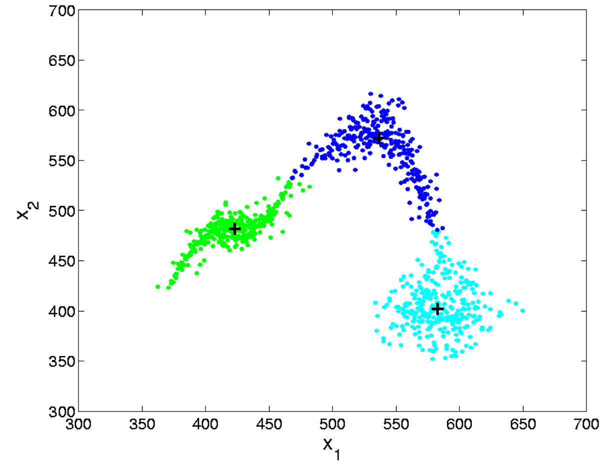


Image Segmentation Algorithm

in 5D (3 color + 2 lattice) or 3D (1 gray +2 lattice) feature space

(1) ***Filtering***: pixel ← value of the nearest mode

(2) ***Fusion***: transitive closure under color information on the region adjacency graph generated by filtering

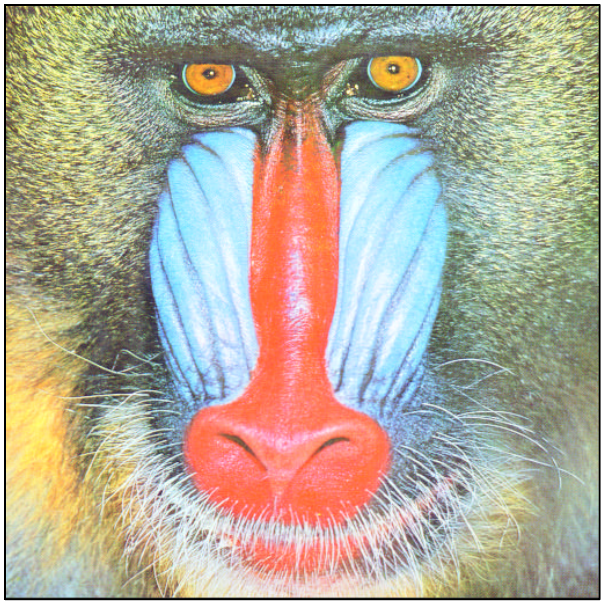
resolution controlled by the window radii: h_s (spatial), h_r (color)

Reference: D. Comaniciu, P. Meer: "Mean Shift: A robust approach toward feature space analysis." *IEEE Trans. Pattern Anal. Machine Intell*, **24**, 603-619, May 2002.

Filtering Examples

original *squirrel*

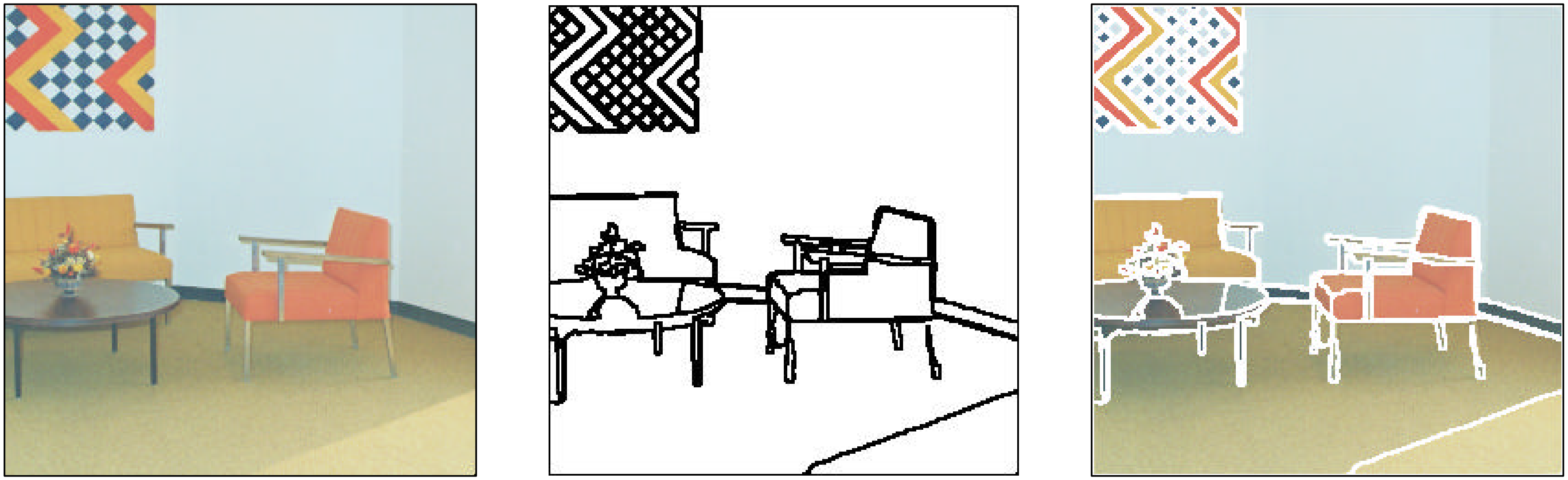
filtered $(h_s, h_r) = (8, 16)$



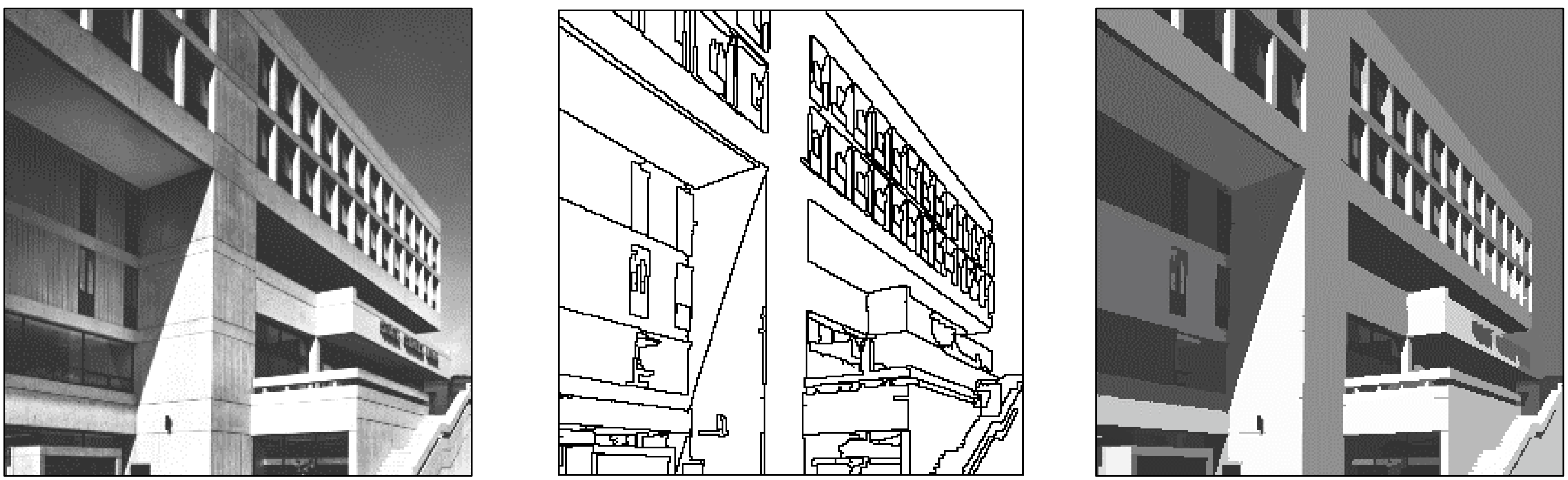
original *baboon*

filtered $(h_s, h_r) = (16, 16)$

Segmentation Examples

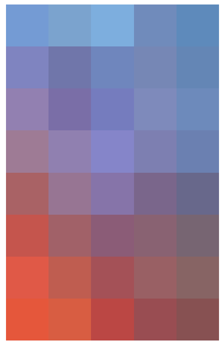


$$(h_s, h_r, M) = (8, 5, 20)$$



$$(h_s, h_r, M) = (8, 7, 20)$$

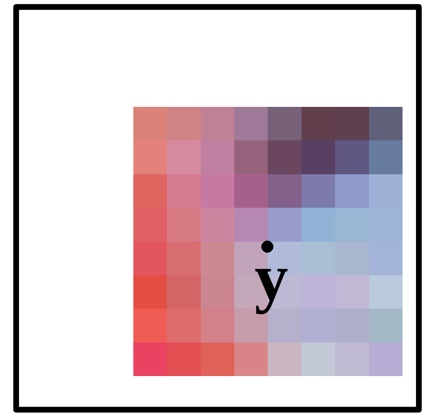
Tracking of Non-Rigid Objects



model

$\{\hat{q}_u\}_{u=1\ldots m}$

target



candidate

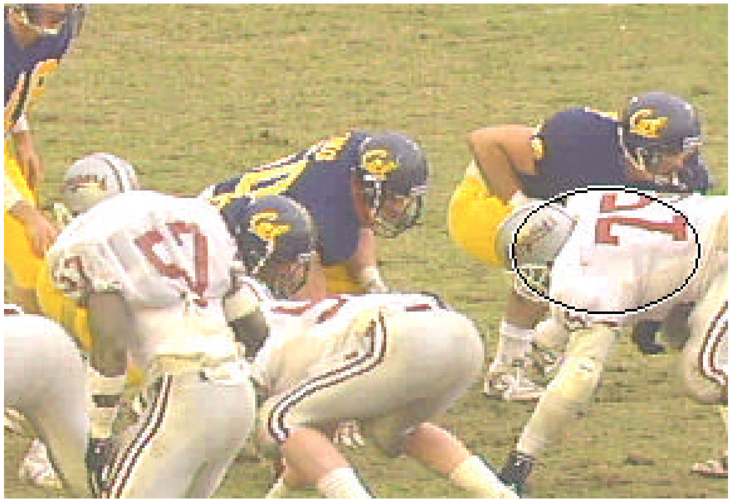
$\{\hat{p}_u(\boldsymbol{y})\}_{u=1\ldots m}$

sampled kernel density estimates in color space

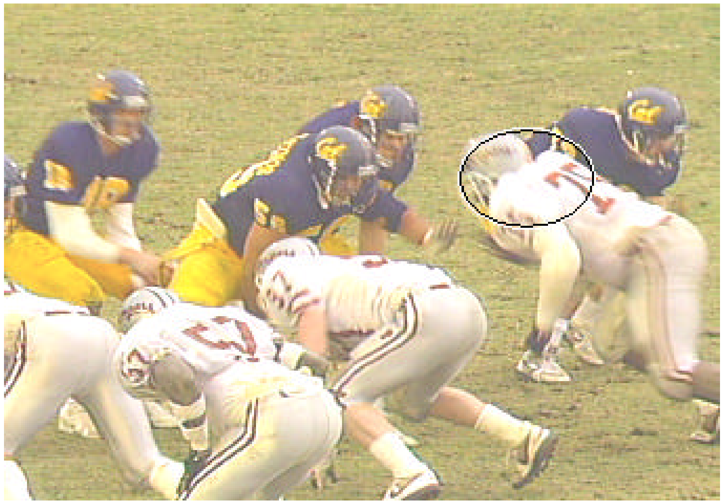
maximizing the Bhattacharyya coefficient $\hat{\rho}(\boldsymbol{y}) = \sum_u \sqrt{\hat{p}_u(\boldsymbol{y}) \cdot \hat{q}_u}$
is equivalent to finding the nearest mode in the scalar field of template matching scores

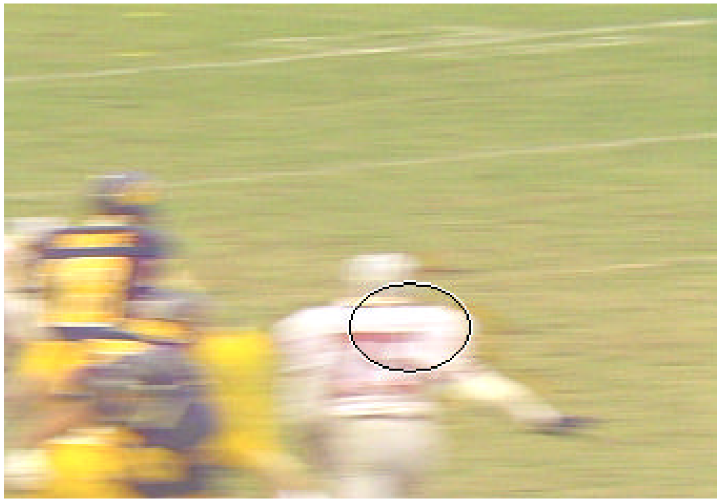
Reference: D. Comaniciu, V. Ramesh, P. Meer: "Kernel-based object tracking'' *IEEE Trans. Pattern Anal. Machine Intell*, **25**, 564-577, May 2003.

Football Sequence (150 frames)



RGB space
histogram of 32 x 32 x 32 bins
Java implementation
30 fps on 600 MHz PC
IIR filter for scale





General Kernel Density Estimation

given $\boldsymbol{x}_i \in \mathcal{R}^p \quad i = 1, \cdots, n$ and the bandwidth matrix $\mathbf{H}$

$$\hat{f}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} K_{\mathbf{H}}(\boldsymbol{x} - \boldsymbol{x}_i)$$

$$K_{\mathbf{H}}(\boldsymbol{x}) = [\det[\mathbf{H}]]^{-1/2} K(\mathbf{H}^{-1/2}\boldsymbol{x}) = c_{k,p}[\det[\mathbf{H}]]^{-1/2} k(\boldsymbol{x}^{\top}\mathbf{H}^{-1/2}\boldsymbol{x})$$

given $\boldsymbol{x}_i \in \mathcal{R}^p$ *and* $\mathbf{H}_i \quad i = 1, \cdots, n$

sample point kernel density estimator

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_{\mathbf{H}_i}(\mathbf{x} - \mathbf{x}_i)$$

is the adequate variable bandwidth technique

important particular case: $\mathbf{H}_i = h_i^2 \mathbf{I}_p$

Adaptive Mean Shift

$$\widetilde{\mathbf{H}}(\boldsymbol{x}) = \left(\sum_{i=1}^{n} w_i(\boldsymbol{x})\mathbf{H}_i^{-1}\right)^{-1}$$

where

$$w_i(\boldsymbol{x}) = \frac{[\det[\mathbf{H}_i]]^{-1/2}\, g\left(\mathsf{D}\left[\boldsymbol{x}, \boldsymbol{x}_i, \mathbf{H}_i\right]^2\right)}{\sum_{i=1}^{n} [\det[\mathbf{H}_i]]^{-1/2}\, g\left(\mathsf{D}\left[\boldsymbol{x}, \boldsymbol{x}_i, \mathbf{H}_i\right]^2\right)}$$

and

$$\mathsf{D}\left[\boldsymbol{x}, \boldsymbol{x}_i, \mathbf{H}\right]^2 \equiv (\boldsymbol{x} - \boldsymbol{x}_i)^\top \mathbf{H}^{-1} (\boldsymbol{x} - \boldsymbol{x}_i)$$

the adaptive mean shift

$$\mathbf{m}_G(\boldsymbol{x}) = \widetilde{\mathbf{H}}(\boldsymbol{x}) \sum_{i=1}^{n} w_i(\boldsymbol{x})\mathbf{H}_i^{-1}\boldsymbol{x}_i - \boldsymbol{x}$$

has the property

$$\mathbf{m}_G(\boldsymbol{x}) = \frac{c}{2}\, \widetilde{\mathbf{H}}(\boldsymbol{x}) \frac{\hat{\nabla} f_K(\boldsymbol{x})}{\hat{f}_G(\boldsymbol{x})}$$

Mean Shift in High Dimensions

statistical curse of dimensionality:

sparseness of the data

↓

mode detection requires variable bandwidth

computational curse of dimensionality:

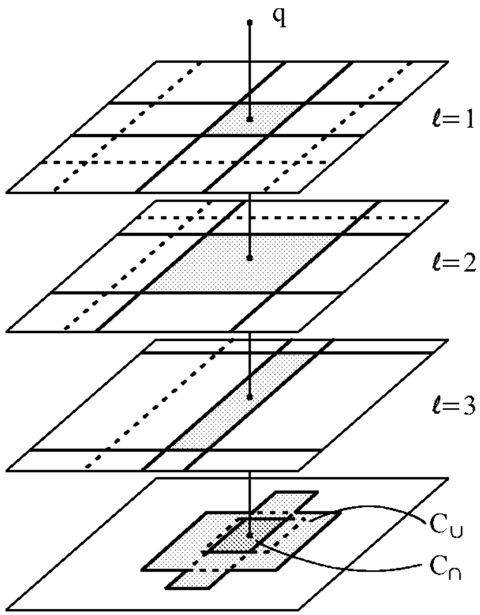
range queries very expensive

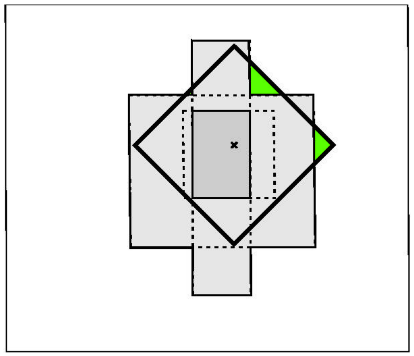
↓

approximate nearest neighbor

Mean Shift in High Dimensions: LSH

idea: locality sensitive hashing [Gionis, Indyk, Motwani, 99]





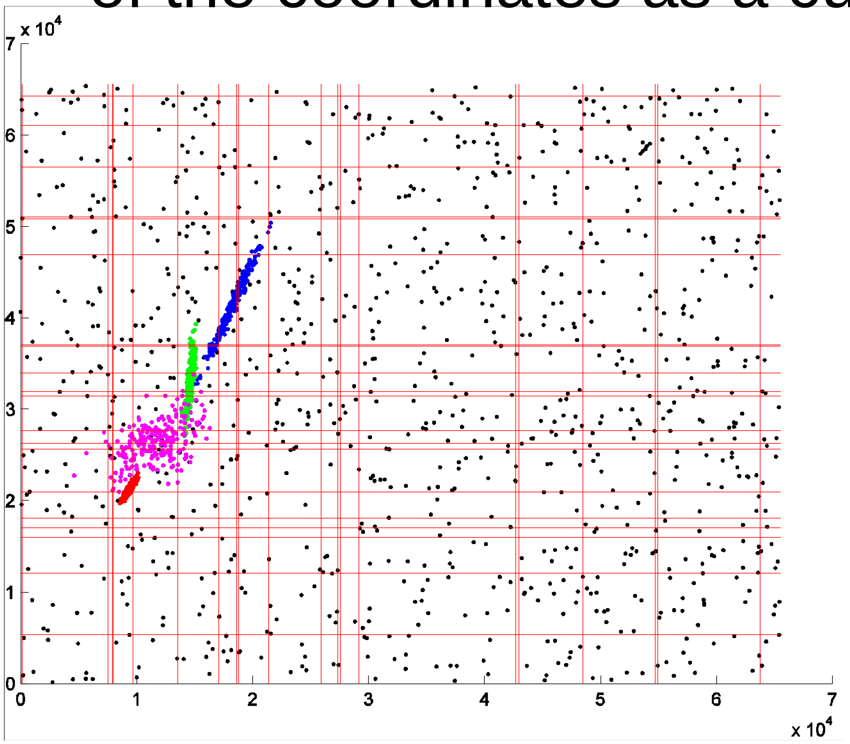
L random tessellations of the space by K random cuts each

query returns the content of the L cells in which the query point q is located in *sublinear time*

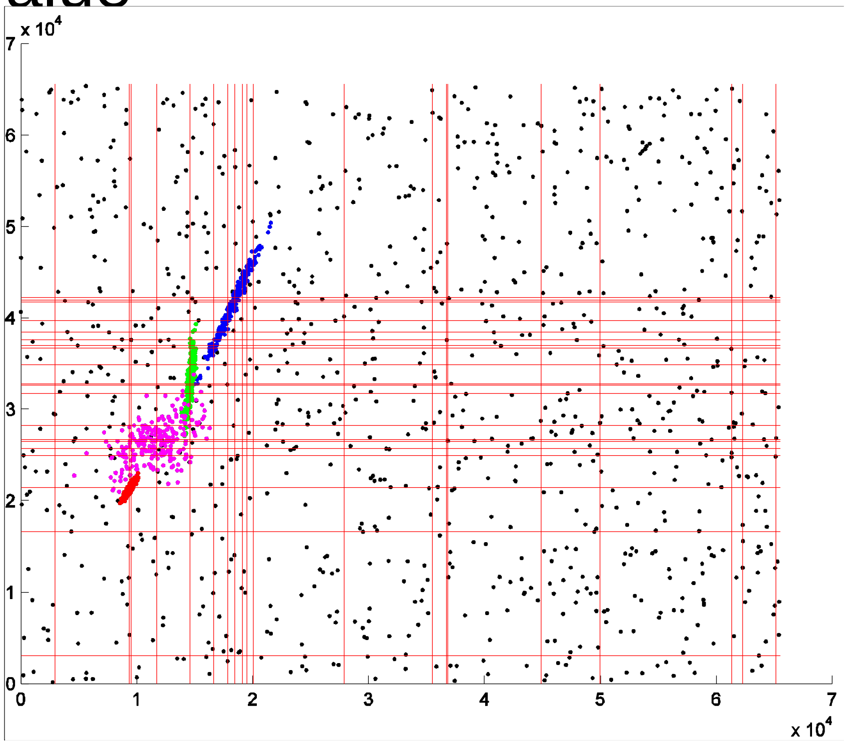
Randomly Selecting the Cuts

in original LSH cut values are randomly selected in the *range* of the data

we randomly select a *point* from the data and use one of the coordinates as a cut value



uniform



data driven

Fast Adaptive Mean Shift: Synthetic Data

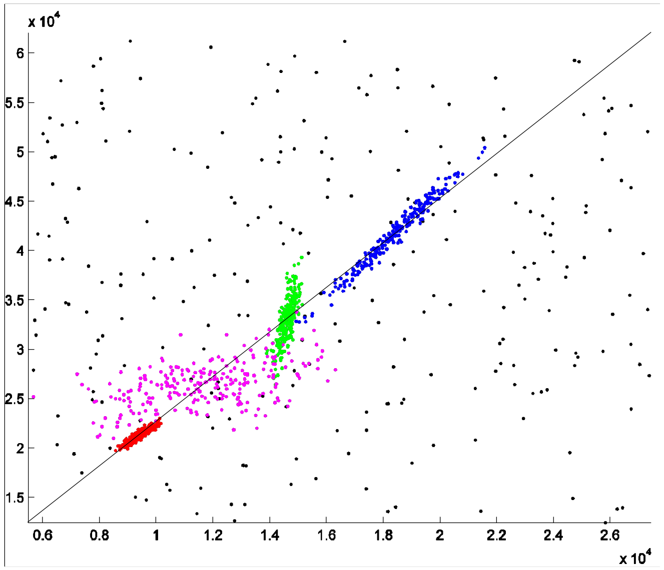
50,000 points in 50D

10x2500: normally distributed

means along a line

random covariances

25,000 points uniformly distributed

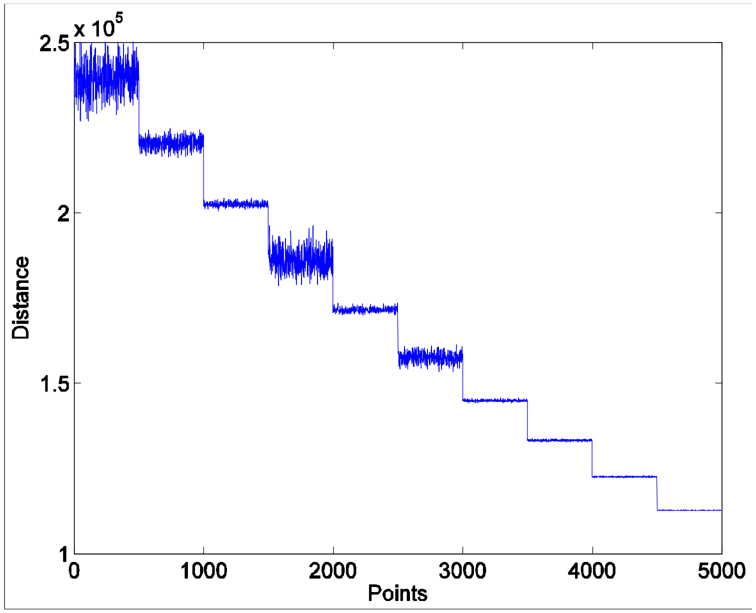


K and L chosen to minimize running time

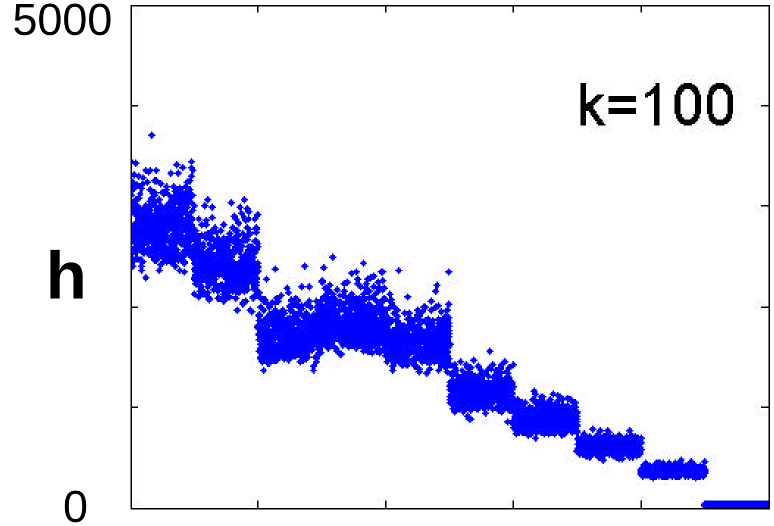
for finding *all* the points in a neighborhood

the bandwidth associated with a point is defined as the region containing *k* neighbors

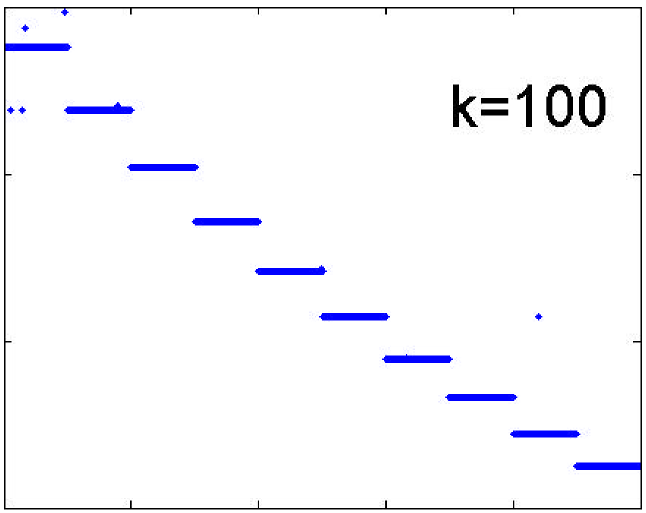
Fast Adaptive Mean Shift: Synthetic Data (2)



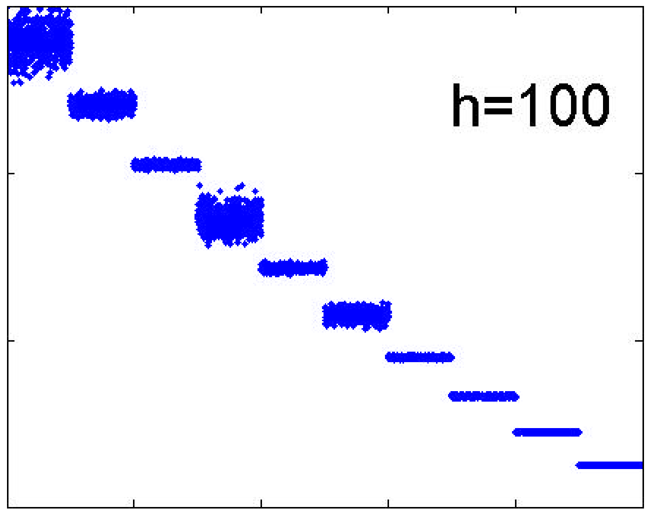
original data



bandwidth



adaptive



non-adaptive

Example: Multi-spectral Images

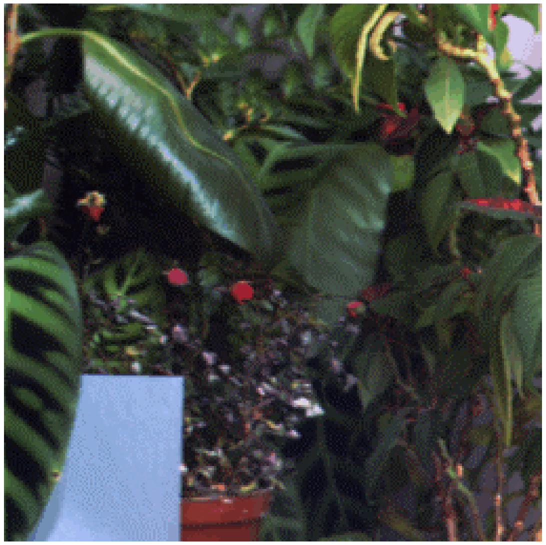
a pixel 31 bands in the visual spectrum

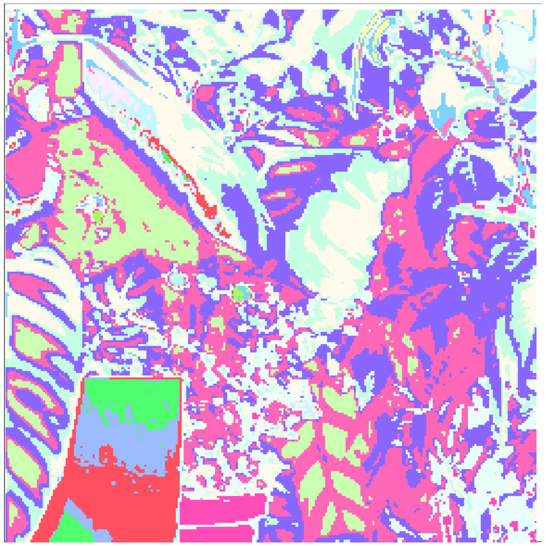
modes detected with fast adaptive mean-shift

pixels allocated

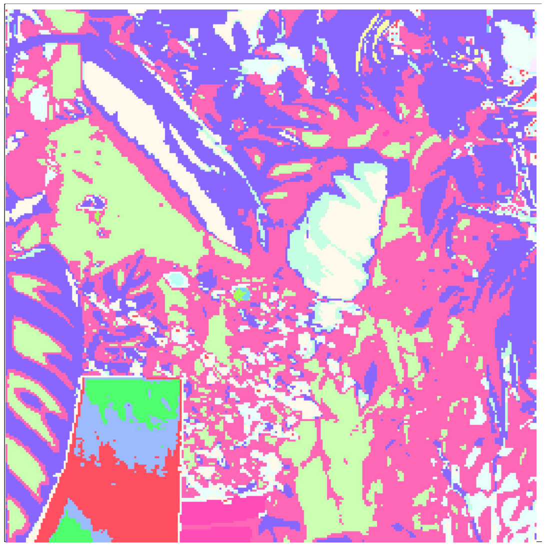
to nearest mode (vector quantization)

by basin of attraction



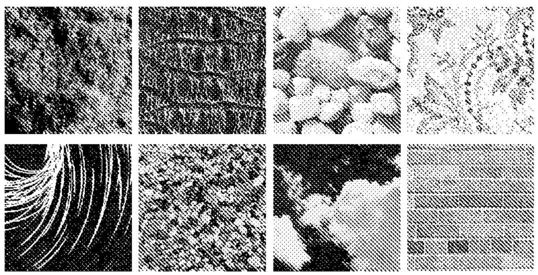


nearest mode

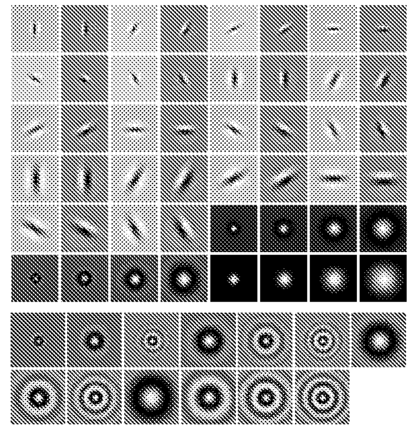


basin of attraction

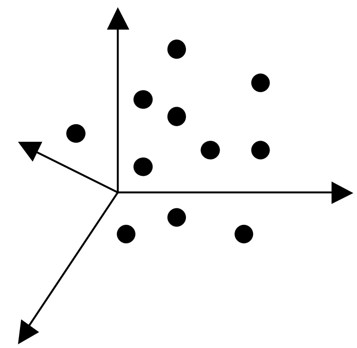
Application: Texture Classification



Brodatz texture examples



filter banks



feature vectors space

cluster centers in feature space = textons [Leung and Malik' 99]

the characteristic repetitive structures of a texture

clustering → *texton* dictionary

each texture represented by the histogram of textons

query: texture class assigned based on the $?^2$ distance between histograms

Application: Texture Classification (2)

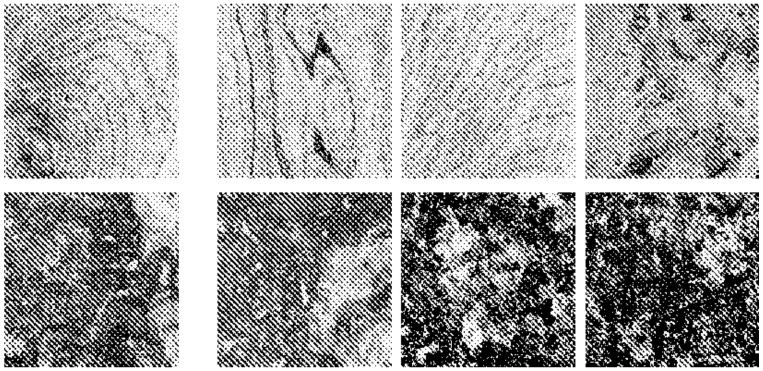
Brodatz database:

112 textures; each texture: 2 train, 2 test;

classification performance

Filter	M4	M8	S	LM
RND	84.82%	88.39%	89.73%	92.41%
k-means	85.71%	94.64%	93.30%	97.32%
FMS	85.71%	94.64%	93.75%	98.21%
AFMS				98.66%

retrieval example:



query retrievals

Application: Texture Classification (3)

different texton locations yield different representations

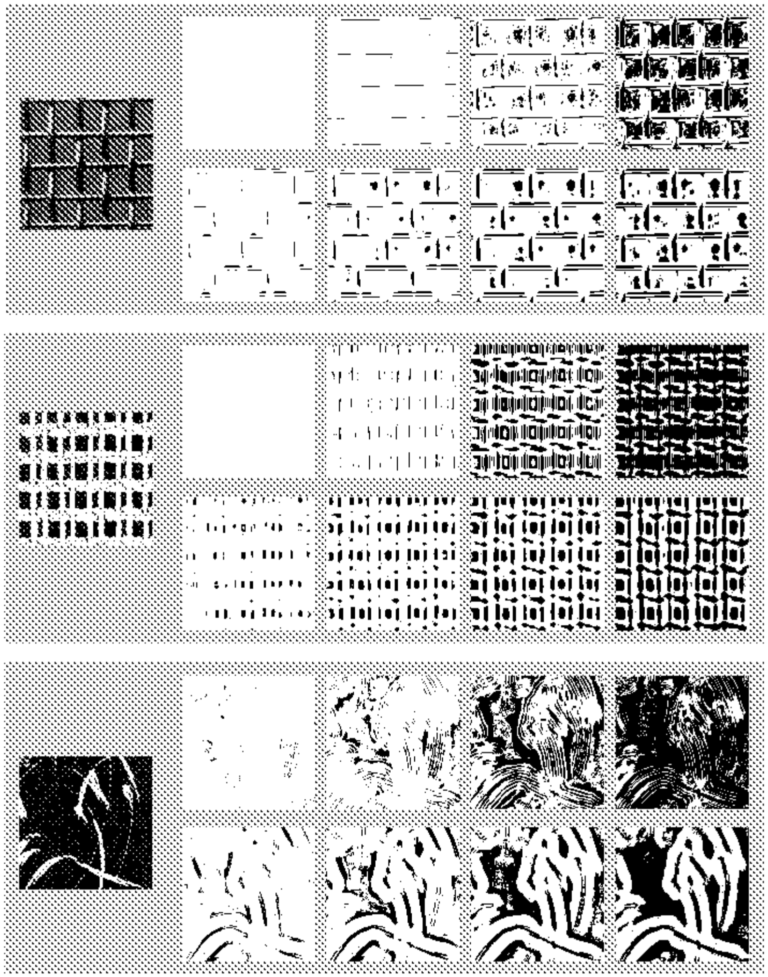


image pixels at increasing distance from textons

k-means textons

AFMS textons

Conclusion

mean shift: proven nonparametric method for mode seeking

applications: image segmentation
object tracking
texture classification
etc…

implementing adaptive mean shift using LSH extends the method to high dimensional spaces