Computer Vision – Lecture 9

Pattern recognition applied to vision

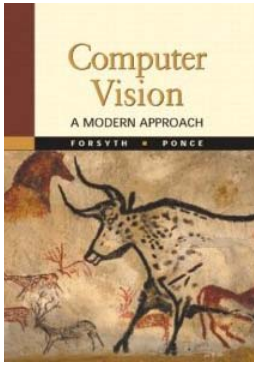
UCT2 – Information Technologies
MAP-I Doctoral Programme

Pedro Quelhas

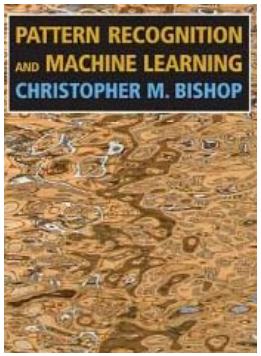
7 December 2010

References

- **Computer vision**
  - David A. Forsyth and Jean Ponce, “Computer Vision: A Modern Approach“ (chapters 22,23,24)



- **Machine learning/ Patter recognition**
  - Christopher M. Bishop , “Pattern Recognition and Machine Learning”



- **Lectures from previous years:**
  - http://www.dcc.fc.up.pt/~mcoimbra/lectures/mapi_0809.html

Outline

- **Issues with pattern recognition tools**
  - Cost
  - Over-fitting
- **Unsupervised Learning and Clustering**
  - Clustering (K-means EM algorithm)
  - Spectral clustering
  - Latent semantic analysis
- **Applied examples of methods from previous lectures**
  - Face and pedestrian detection
- **Feature and classifier fusion**
- **Boosting**
  - ADABoost
  - Viola Jones face detector
- **Local interest point detectors**
- **Object models based on local descriptors**

Cost

- While performing a decision task we are sometimes faced with different cost for our decisions:
- For example:
  - Customers who buy salmon will object vigorously if they see sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.
- How does this knowledge affect our decision?
  - Move from an EER decision to one where certain low cost errors are more likely.
  - In this case we minimize cost and not total error count.

EER – equal error rate (the same point of operation as minimum error only in the case of equal priors).

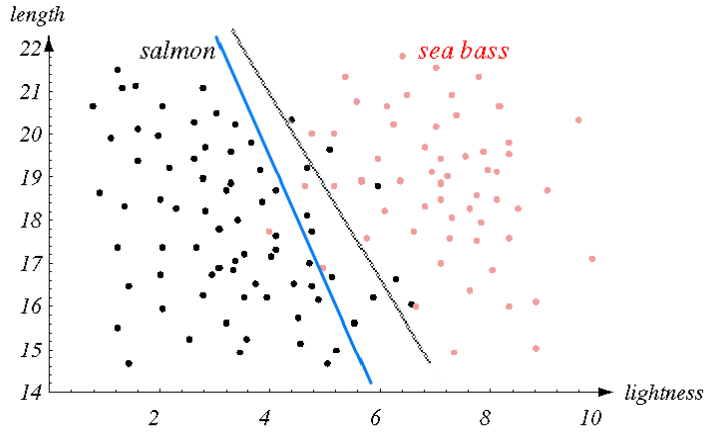


Figure: Scatter plot of lightness and length features for training samples with distinct costs.

Over-fitting

- How do we define the best pattern recognition machine to do the classification job?
  - Is the lowest training error a good estimation of classification performance?
  - NO! The lowest training error is normally a result of an overly complex solution which will most likely not generalize!

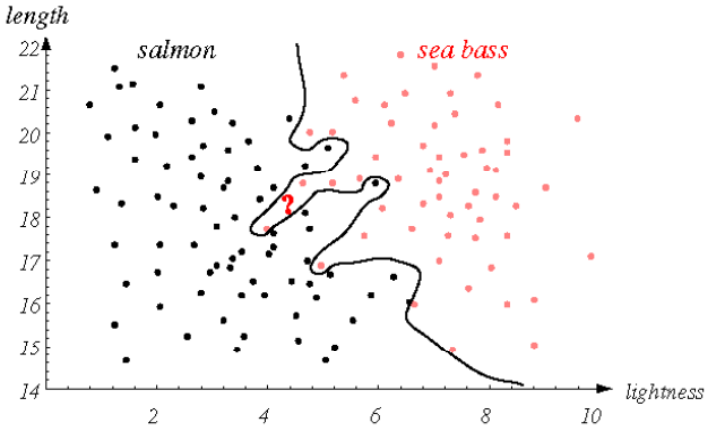


Figure: We may distinguish training samples perfectly but how can we predict how well we can generalize to unknown samples?

Generalization power is the capacity of a classification method to perform equality well in test data as in training

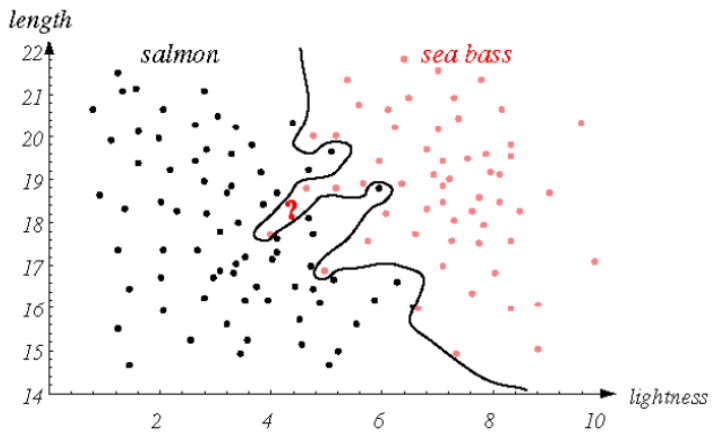
- How do we know if we have a good general solution?
  - Perform cross-validation of the training of our patter recognition tool. While training is done in part of the training data, error evaluation is performed on other data. No over-fitting can occur as the error performance is not calculated in the fitting data.

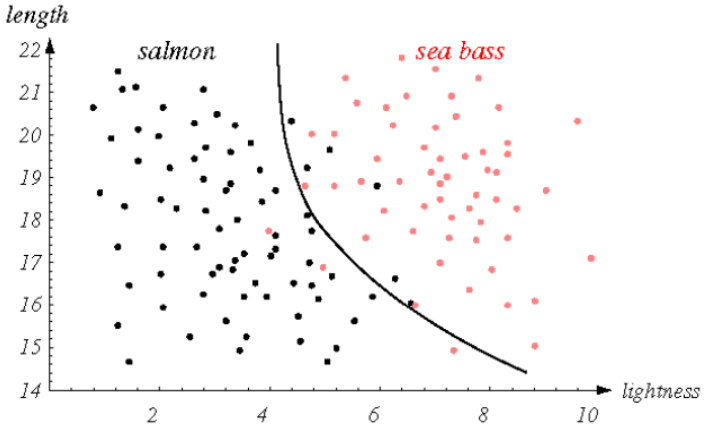
Over-fitting

- Evaluate the final tool's complexity.
  - For similar training errors, tools with lower complexity will most likely lead to better results when testing.
  - Complexity can be estimated by computing the VC dimension (Vapnik–Chervonenkis) -> h
  - Given the VC dimension we can obtain an upper limit on the test error:

$$\text{Training error} + \sqrt{\frac{h(\log(2N/h)+1) - \log(\eta/4)}{N}}$$

- Simpler classifiers (in terms of decision rule) lead to more general, stable, solutions:





Unsupervised learning and clustering

- Unsupervised learning is concerned with inferring properties of the probability density P(x) without the help of a supervisor or teacher providing correct answers or a degree-of-error for each observation. The goal is to characterize x-values, or collections of such values, where P(x) is relatively large. Exploratory data analysis can provide insight into the nature or structure of the data.
  - principal components, multidimensional scaling, self-organizing maps, principal curves, etc, attempt to identify low-dimensional manifolds within the x-space that represent high data density.
  - **cluster analysis** attempts to find multiple (convex) regions of the x-space that contain modes of P(x).
  - association rules attempt to construct simple descriptions (conjunctive rules) that describe regions of high density in the special case of very high dimensional binary-valued data.

Cluster analysis or clustering

- Cluster analysis:
  - grouping or segmenting a collection of objects into subsets of clusters, such that those within each cluster are more closely related to one another that objects assigned to different clusters
  - central to all of the goals of cluster analysis is the notion of the degree of similarity (dissimilarity) between the individual objects being clustered.
  - A cluster is comprised of a number of similar objects collected or grouped together.
  - Patterns within a cluster are more similar to each other than are patterns in different clusters.
  - Clusters may be described as connected regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points.

- Objective: Organizing data into groups (clusters) such that there is
  - high intra-class similarity
  - low inter-class similarity

- Task: Finding the groups attributions and the number of groups directly from the data (in contrast to classification).
- One fundamental problem is that the task of clustering must be based on a definition of distance. This is not easy, but we will assume it is done!

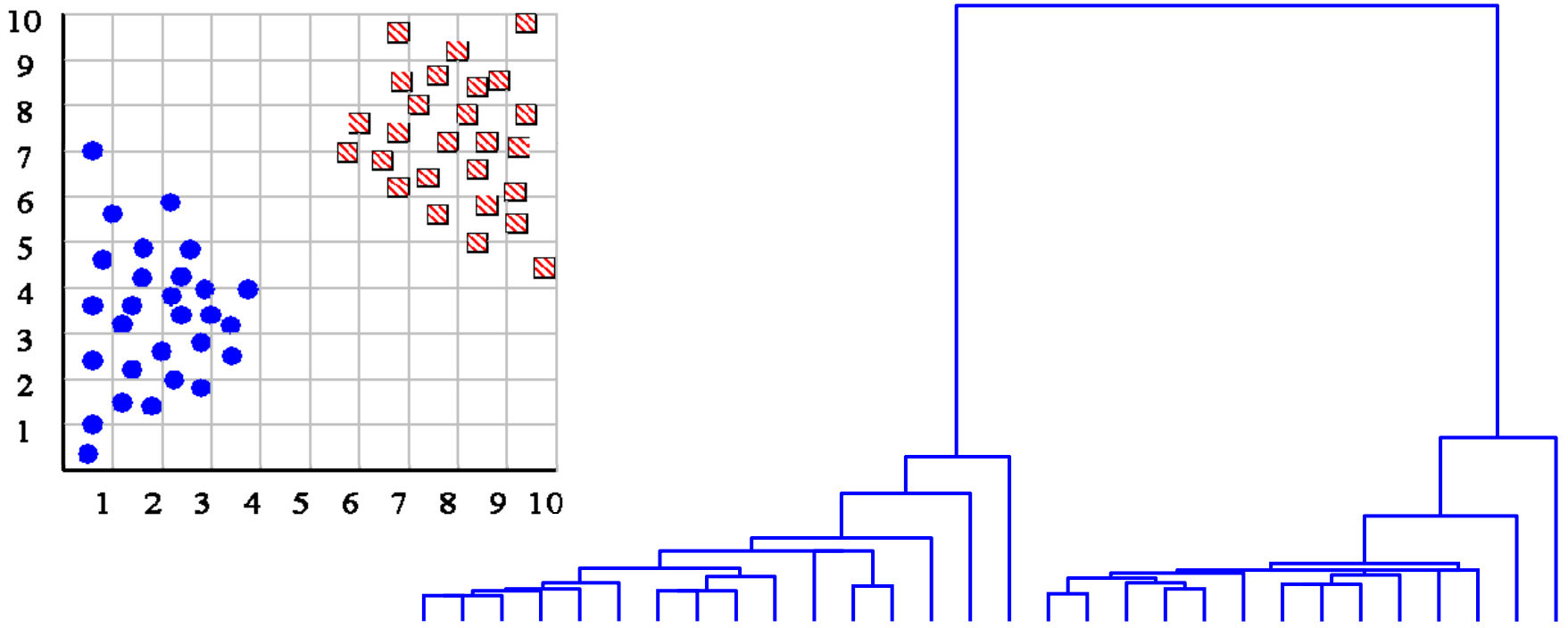
Clustering

- Types of clustering algorithms:
  - Partitional : Construct various partitions and then evaluate them by some criterion
  - Hierarchical : Create a hierarchical decomposition of the set of objects using some criterion

- Desirable properties:
  - Scalability (in terms of both time and space)
  - Ability to deal with different data types
  - Minimal requirements for domain knowledge to determine input parameters
  - Able to deal with noise and outliers
  - Insensitive to order of input records
  - Incorporation of user-specified constraints
  - Interpretability and usability

Hierarchical Clustering

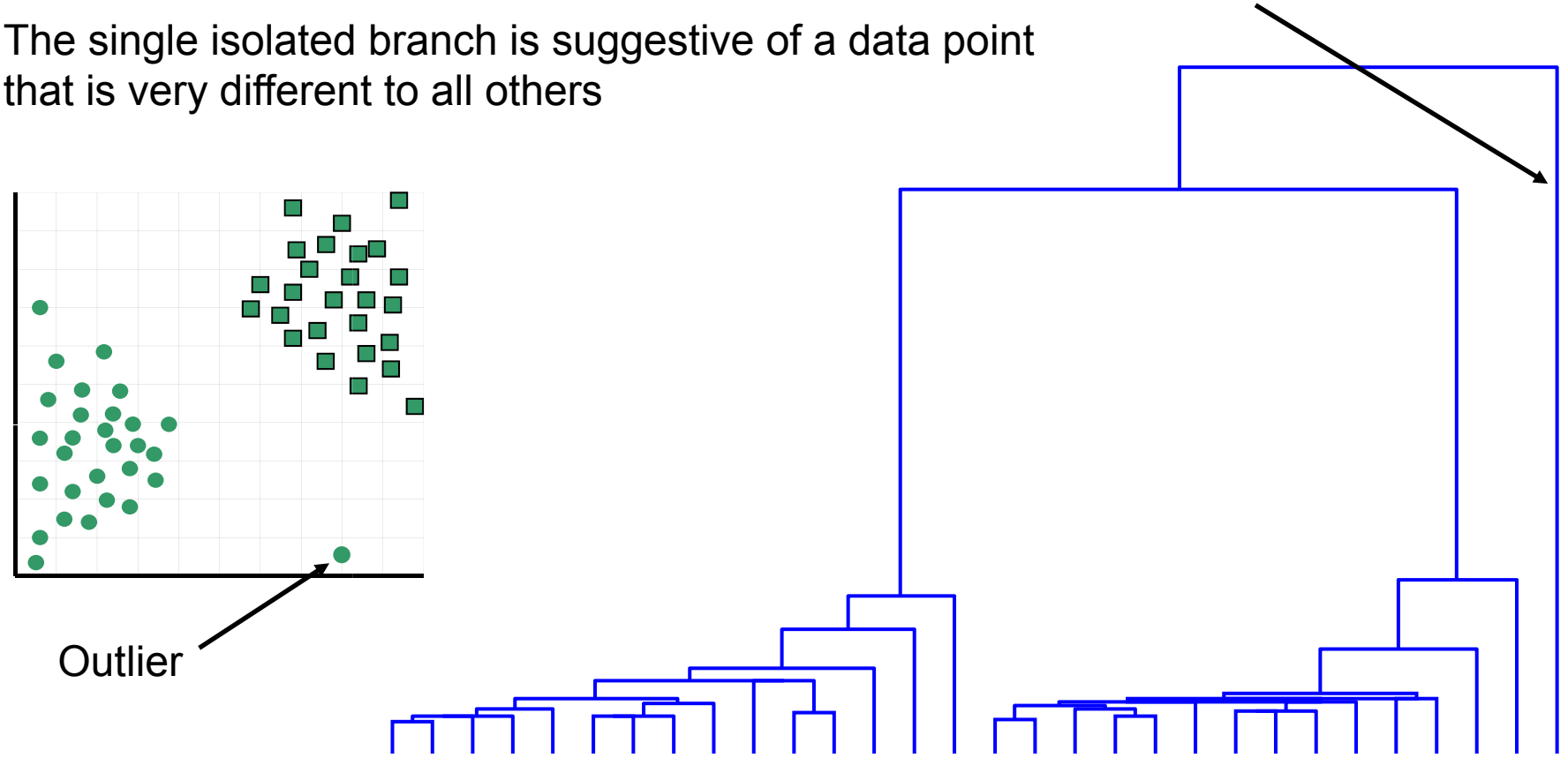
We can look at the dendrogram to determine the “correct” number of clusters. In this case, the two highly separated subtrees are highly suggestive of two clusters.



Hierarchical Clustering

One potential use of a dendrogram is to detect outliers

The single isolated branch is suggestive of a data point that is very different to all others



Hierarchical Clustering

Since we cannot test all possible trees we will have to heuristic search of all possible trees. We could do this..

Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

Top-Down (divisive): Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides.

Cluster distances

We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.

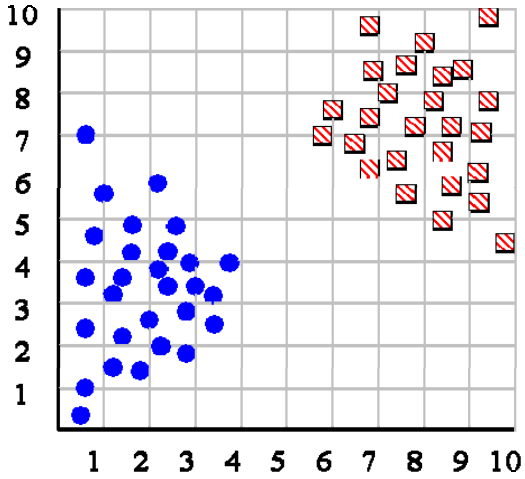
• **Single linkage (nearest neighbor):** In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.

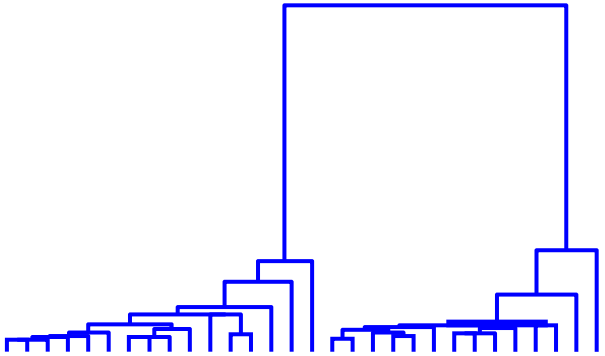
• **Complete linkage (furthest neighbor):** In this method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors").

• **Group average linkage:** In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters.

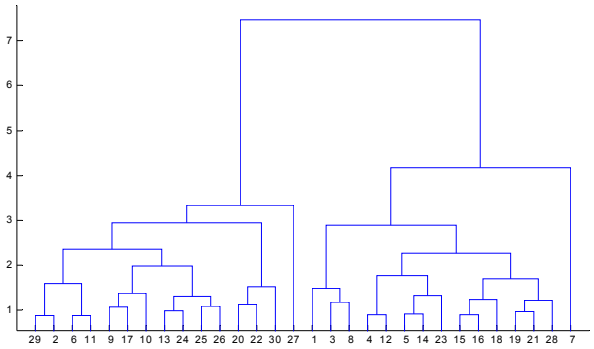
• **Wards Linkage**: In this method, we try to minimize the variance of the merged clusters

Examples

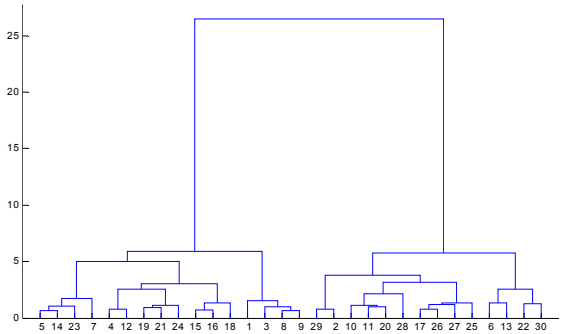




Single linkage



Average linkage



Wards linkage

Summary of Hierarchal Clustering Methods

•No need to specify the number of clusters in advance.

• Hierarchal nature maps nicely onto human intuition for some domains

• They do not scale well: time complexity of at least O(n^2), where n is the number of total objects.

• Like any heuristic search algorithms, local optima are a problem.

• Interpretation of results is (very) subjective.

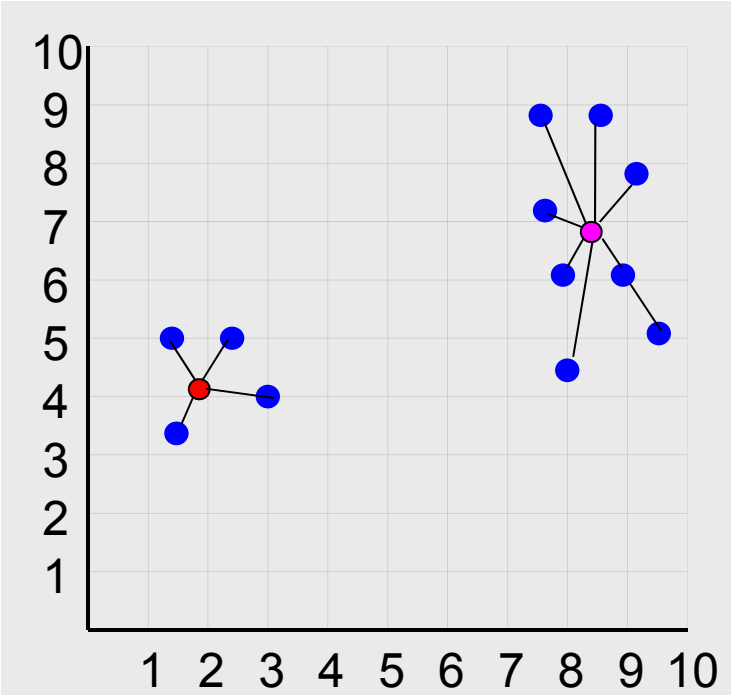
Partitional Clustering

- Nonhierarchical, each instance is placed in exactly one of K non-overlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K.

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

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

Objective Function



Algorithm *k-means*

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (randomly, if necessary).

3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

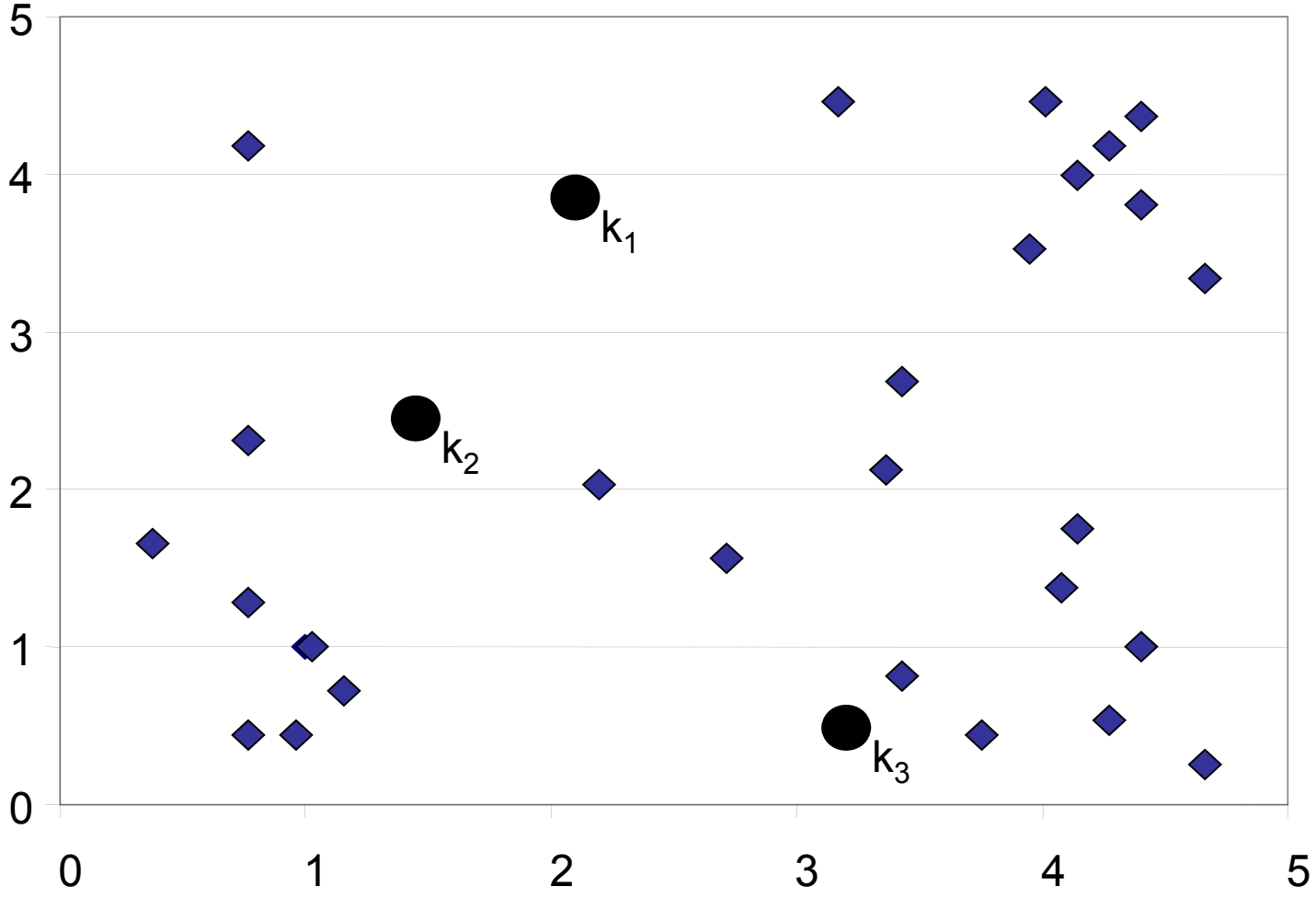
4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise goto 3.

end

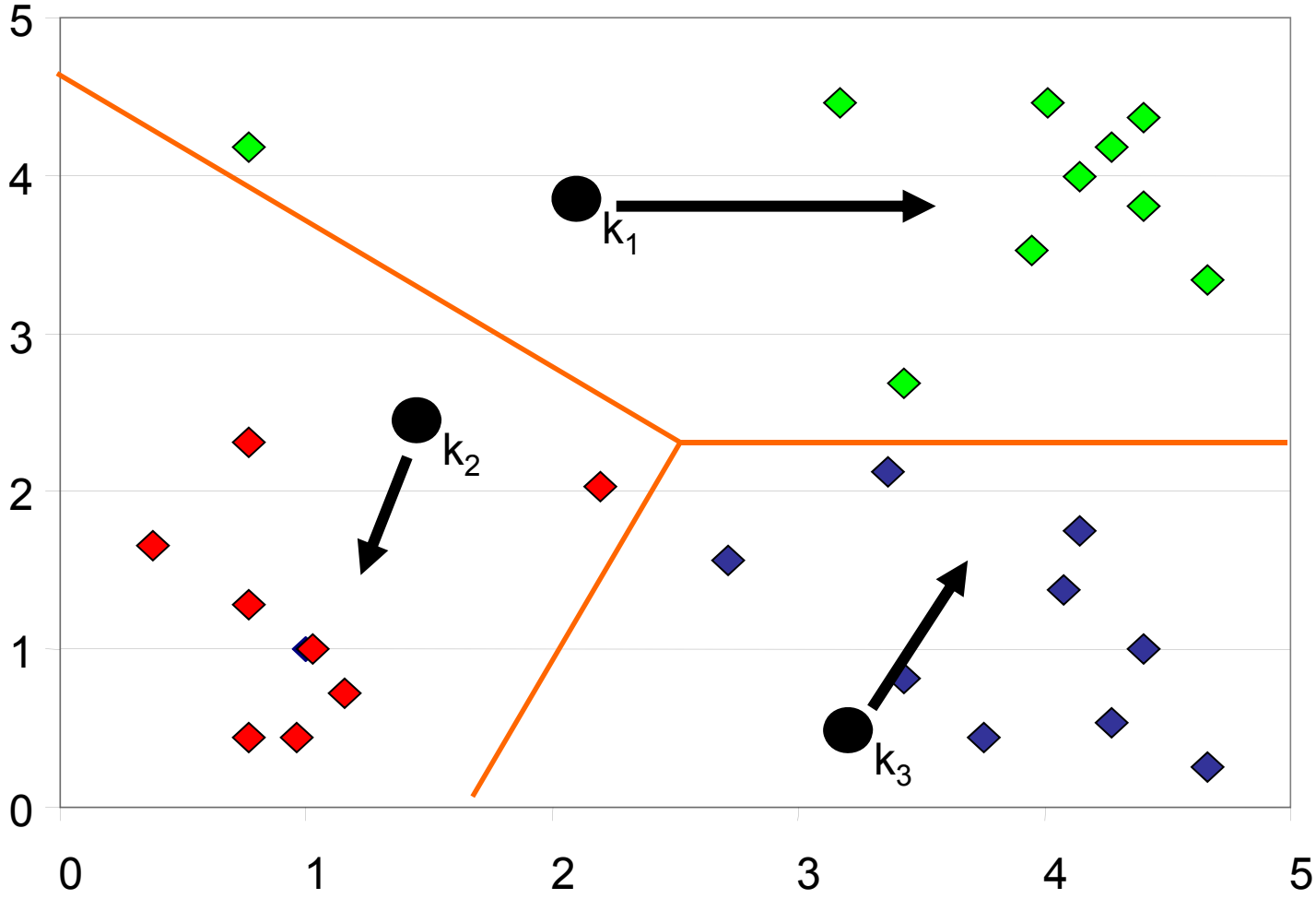
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



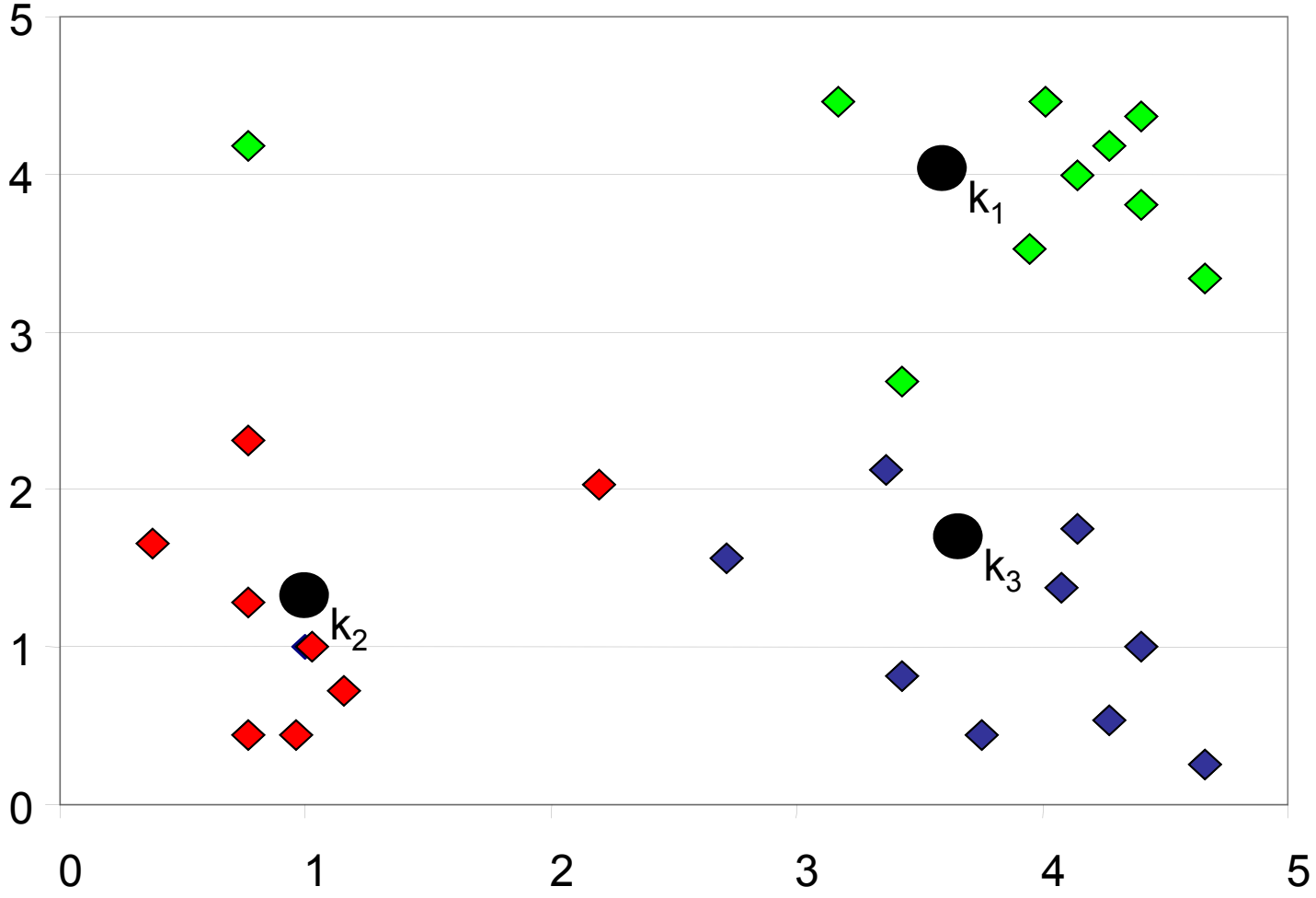
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



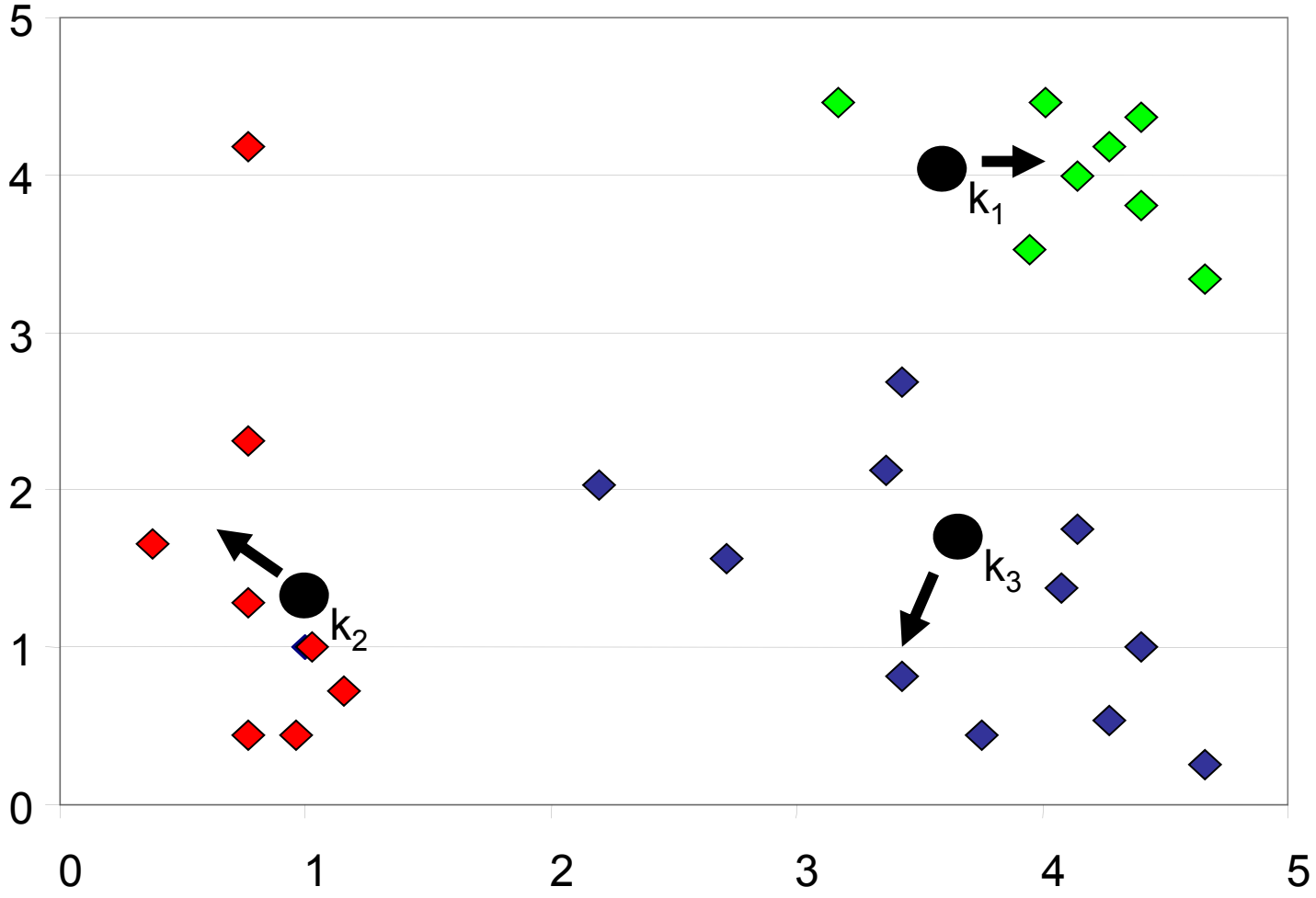
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



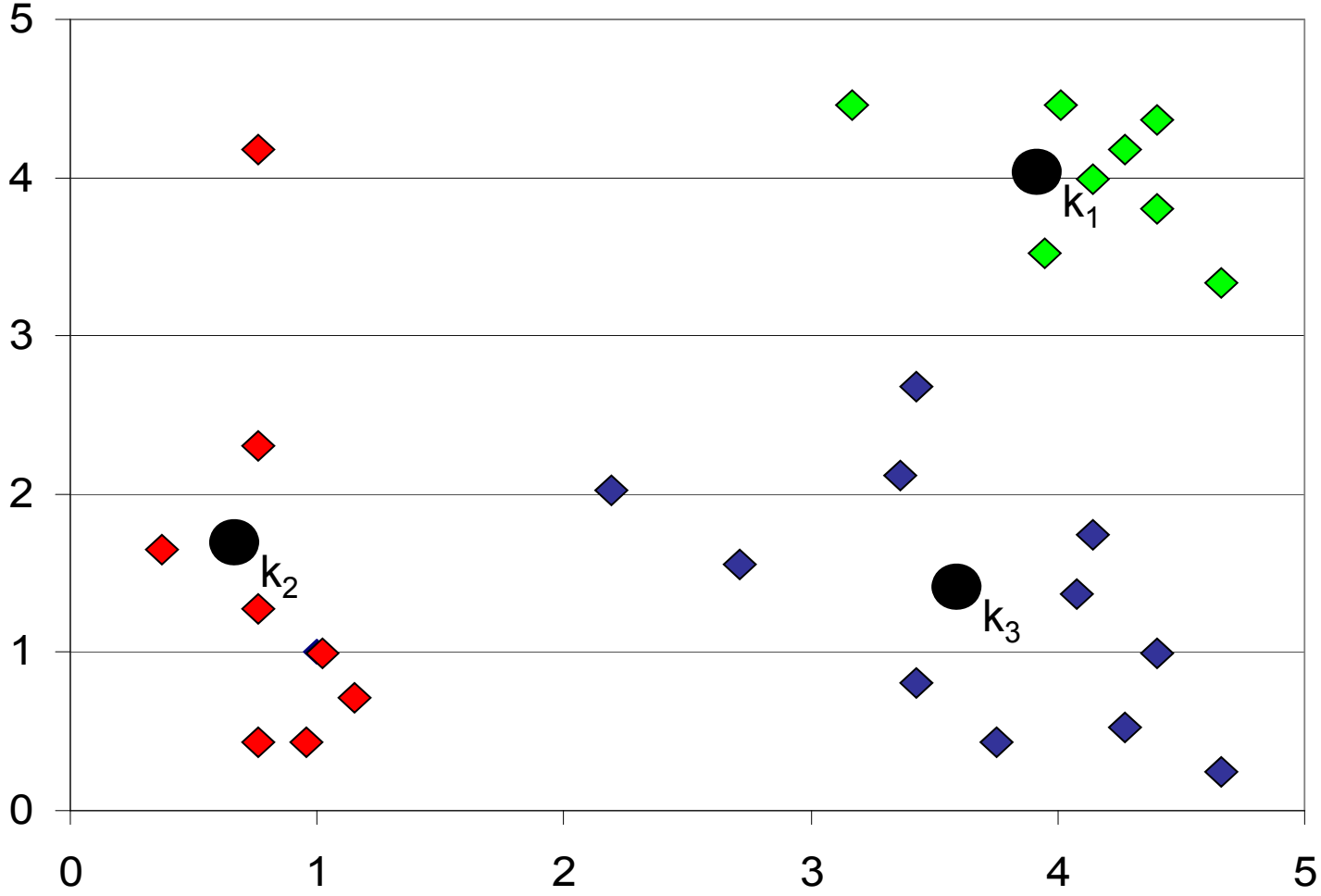
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



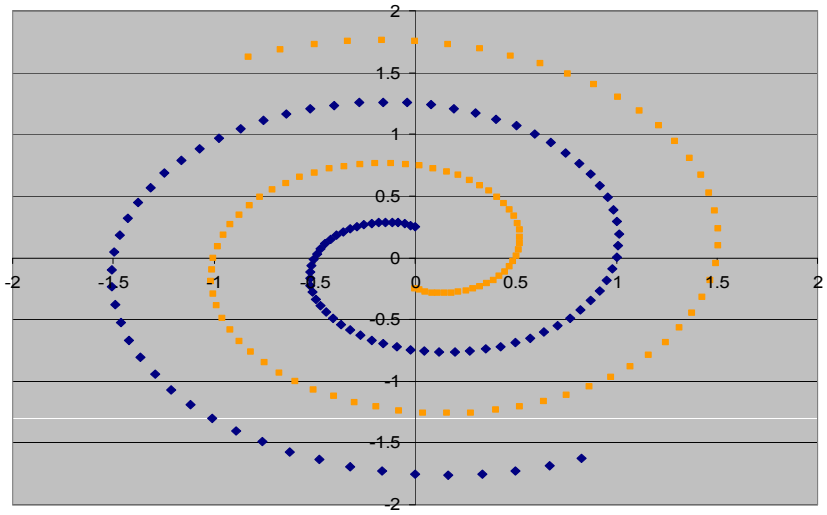
Comments on the *K-Means* Method

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

- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Non-separable clusters!

- **Some groupings are apparent but cannot be easily extracted:**



Dataset exhibits complex cluster shapes

$\Rightarrow$ K-means performs very poorly in this space due bias toward dense spherical clusters.

Spectral Clustering Algorithm

- Starting point:
  - Given a set of points

$$S = \{s_1, \ldots, s_n\} \in R^l$$

  - We would like to cluster them into k subsets

- Form the affinity matrix $W \in R^{nxn}$

- Define $W_{ij} = e^{-\|s_i - s_j\|^2 / 2\sigma^2}$ for all $i \neq j$

$$W_{ii} = 0$$

- Define D a diagonal matrix whose (i,i) element is the sum of W's row i

On **Spectral Clustering**: Analysis and an algorithm: **Ng** A.Y., **Jordan**, M.I., and **Weiss** Y NIPS 2001

Spectral Clustering Algorithm

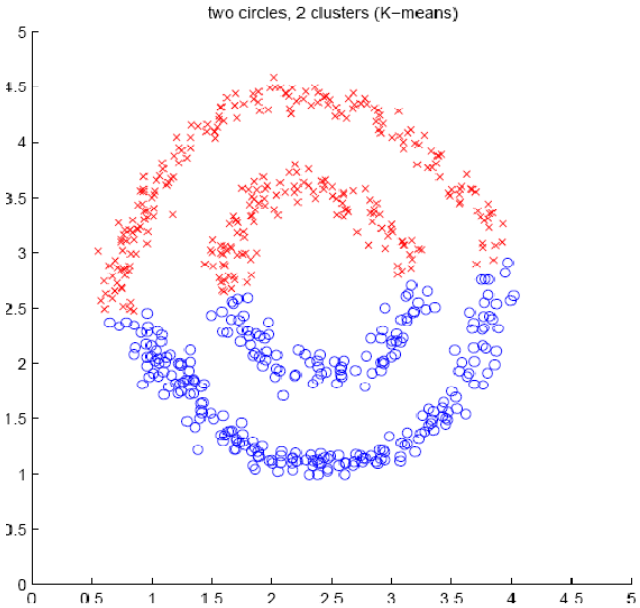
- Form the matrix $L = D^{-1/2} W D^{-1/2}$
- Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L
- These form the columns of the new matrix X
  - Note: have reduced dimension from nxn to nxk

Spectral Clustering Algorithm

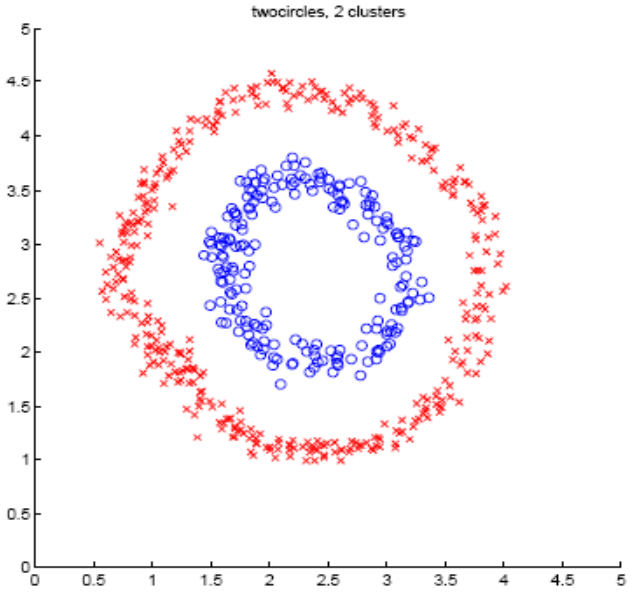
- Form the matrix Y
  - Renormalize each of X's rows to have unit length
  - $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^2$
  - Y $\in R^{nxk}$
- Treat each row of Y as a point in R^k
- Cluster into k clusters via K-means
- Final Cluster Assignment
  - Assign point to cluster j if row i of Y was assigned to cluster j

Results

- K-means clustering



Spectral clustering



Semantics

- Syntax - structure of words, phrases and sentences
- Semantics - meaning of and relationships among words in a sentence
- Extracting an important *meaning* from a given document
- Contextual meaning
- Compositional semantics
  - uses parse tree to derive a hierarchical structure
  - informational and intentional meaning
  - rule based
- Classification
  - Bayesian approach
- Statistics-algebraic approach (LSA)

Latent Semantic Analysis

- LSA is a fully automatic statistics-algebraic technique for extracting and inferring relations of expected contextual usage of words in documents
- It uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars
- Takes as input row text

Building latent semantic space

- Training corpus in the domain of interest
- document
  - a sentence, paragraph, chapter
- vocabulary size
  - remove stopwords

• Given - N documents, vocabulary size M

•Generate a word-documents co-occurrence matrix **W**

W =

	d_1	d_2	…..	d_N
w_1				
w_2				
:				
:				
w_M				

$c_{i,j}$ number of times w_i occurs in d_j;

n_j total number of words present in d_j;

Discriminate words

- Normalized entropy

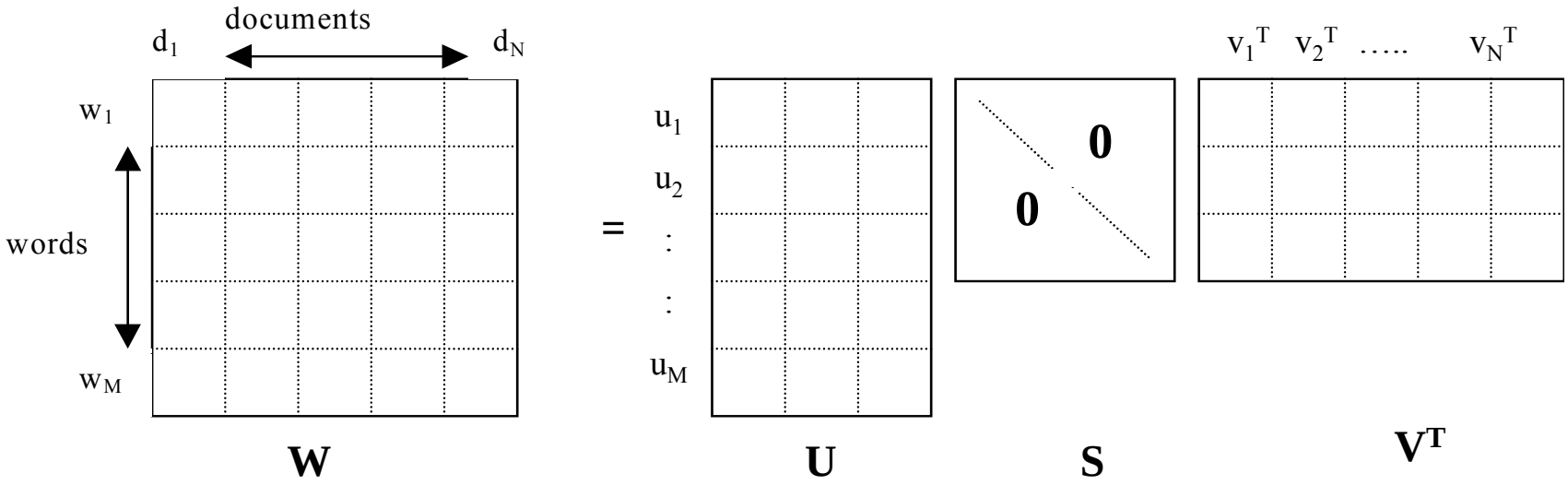
$$\varepsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{t_i} \qquad t_i = \sum_j c_{i,j}$$

  - close to 0 : very important
  - close to 1 : less important

- Scaling and normalization

$$w_{i,j} = (1-\varepsilon_i)\frac{c_{i,j}}{n_j}$$

Singular Value Decomposition



SVD approximation

- Dimensionality reduction
  - Best rank-R approximation
  - Optimal energy preservation
  - Captures major structural associations between words and documents
  - Removes ‘noisy’ observations

- Columns of U : orthonormal documents
- Columns of V : orthonormal words
- Word vector : $\mathbf{u}_i\mathbf{S}$
- Document vector : $\mathbf{v}_j\mathbf{S}$
- words close in LS space appear in similar documents
- documents close in LS space convey similar meaning

LSA as knowledge representation

- Projecting a new document in LS space
- Calculate the frequency count [d_i] of words in the document.

$$\mathbf{d} = \mathbf{U}\,\mathbf{S}\,\mathbf{v}^T$$

$$\Rightarrow \mathbf{U}^T\mathbf{d} = \mathbf{S}\mathbf{v}^T$$

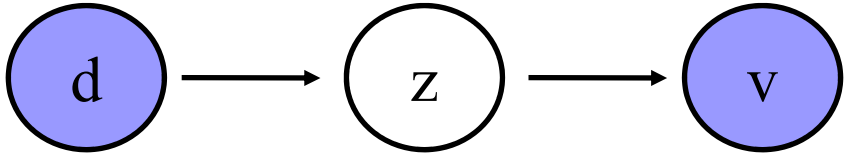
- Thus,

$$\hat{\mathbf{d}}_{LSA} = \mathbf{S}\mathbf{v}^T = \mathbf{U}^T\mathbf{d} = \sum_i (1-\varepsilon_i) d_i \mathbf{u}_i$$

Semantic Similarity Measure

- To find similarity between two documents, project them in LS space
- Then calculate the cosine measure between their projection
- With this measure, various problems can be addressed e.g., natural language understanding, cognitive modeling etc

PLSA [hofmann 01]



$$P(v_j, z_k, d_i) = P(d_i)\, P(z_k \mid d_i)\, P(v_j \mid z_k)$$

$$P(v_j, d_i) = P(d_i) \sum_k P(z_k \mid d_i)\, P(v_j \mid z_k)$$

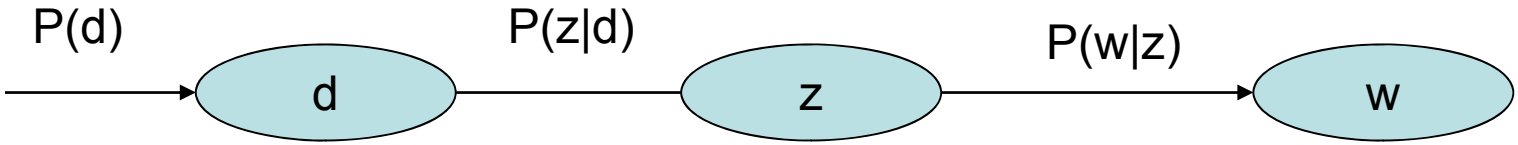
- $P(v_j \mid z_k)$: probability of visterm j given aspect k
- $P(z_k \mid d_i)$: probability of aspect k given image i

The Setting

- Set of N documents
  - D={d_1, … ,d_N}
- Set of M words
  - W={w_1, … ,w_M}
- Set of K Latent classes
  - Z={z_1, … ,z_K}
- A Matrix of size N * M to represent the frequency counts

PLSA – Aspect Model

- Aspect Model
  - Document is a mixture of underlying (latent) K aspects
  - Each aspect is represented by a distribution of words p(w|z)
  - Latent Variable model for general co-occurrence data
  - Associate each observation (w,d) with a class variable z Є Z{z_1,…,z_K}
- Generative Model
  - Select a doc with probability P(d)
  - Pick a latent class z with probability P(z|d)
  - Generate a word w with probability p(w|z)



Aspect Model

- To get the joint probability model

$$P(d,w) = P(d)P(w|d), \text{ where}$$
$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d) \ .$$

- (d,w) – assumed to be independent

- Using Bayes' rule

$$P(d,w) = \sum_{z \in \mathcal{Z}} P(z)P(w|z)P(d|z).$$

$$\begin{aligned} P(d,w) &= P(d)P(w|d), \text{ where} \\ P(w|d) &= \sum_{z \in \mathcal{Z}} P(w|z)P(z|d) \ . \end{aligned}$$

Advantages of this model over Documents Clustering

- Documents are not related to a single cluster (i.e. aspect)
  - For each z, P(z|d) defines a specific mixture of factors
  - This offers more flexibility, and produces effective modeling

Now, we have to compute P(z), P(z|d), P(w|z). We are given just documents(d) and words(w).

Model fitting with Tempered EM

- We have the equation for log-likelihood function from the aspect model, and we need to maximize it.

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d, w) \log P(d, w) ,$$

- Expectation Maximization (EM) is used for this purpose
  - To avoid overfitting, tempered EM is proposed

EM Steps

- E-Step
  - Expectation step where expectation of the likelihood function is calculated with the current parameter values
- M-Step
  - Update the parameters with the calculated posterior probabilities
  - Find the parameters that maximizes the likelihood function

E Step

- It is the probability that a word w occurring in a document d, is explained by aspect z

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')},$$

(based on some calculations)

M Step

- All these equations use p(z|d,w) calculated in E Step

$$P(w|z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{d,w'} n(d,w')P(z|d,w')},$$

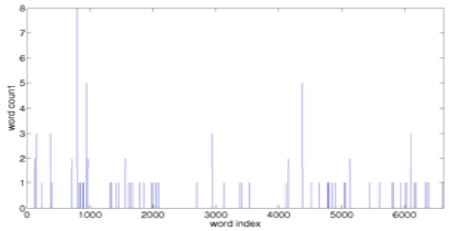
$$P(d|z) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{d',w} n(d',w)P(z|d',w)},$$

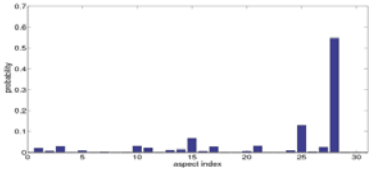
$$P(z) = \frac{1}{R}\sum_{d,w} n(d,w)P(z|d,w), \quad R \equiv \sum_{d,w} n(d,w).$$

- Converges to local maximum of the likelihood function

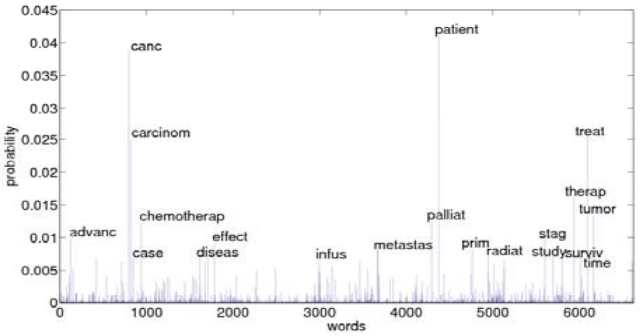
PLSa example: medical diagnosis documents

chemotherapy of breast cancer the majority of cancer patients who are referred for chemotherapy are in far advanced stages of the disease. the difficulty in deliver- ing an effective dose of an oncolytic agent to the involved areas may limit the results of treatment unless the patient's tumor is an unusually responsive one. the variability in response between apparently similar cases of cancer of the same cell type has become increasingly evident, and is as puzzling as the fact that 20 per cent of patients with breast cancer will live 5 years with no treatment. The decisions as to the choice between chemotherapy and hormone.... therapy, as well as to the choice of the chemotherapeutic ....

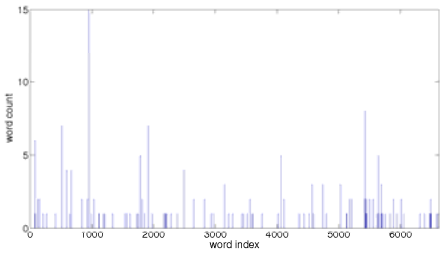


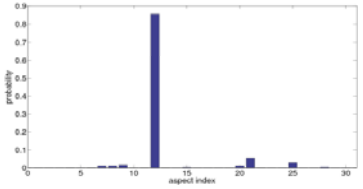


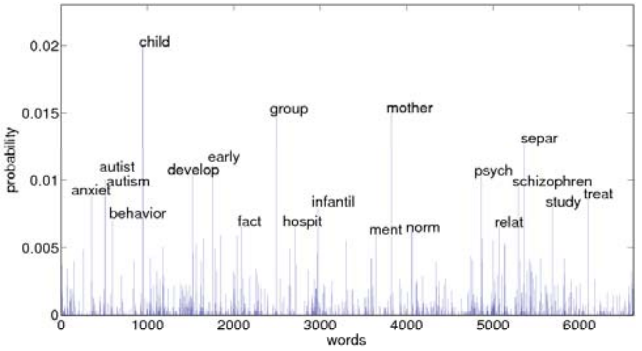
Latent aspects



it was possibleto carry out simultaneous behavioural andeeg observations by means of telemetry.the children were observed for three minuteperiods in four environments of varyingcomplexity: a. an empty room;..... the.... room with toy wooden blocks present;
c. blocks plus passive adult; d. blocks plus active adult. except in a the visual fixations and manipulatory activities were of significantly shorter duration in the autistic children than in the normals, lacked clear 'cut-off' points and showed relatively little variability between environments. the autistic children as a whole showed virtually no constructive play in contrast with the normals, though more time was spent in contact with play materials in the social environments. changes in the time spent in locomotion, gesturing, block-play and manipulation of other objects in the environment showed that the autistic children were acutely responsive to changes in their environment, including the presence of the social stimulus. if the autistic children were subdivided into those







Combining Features

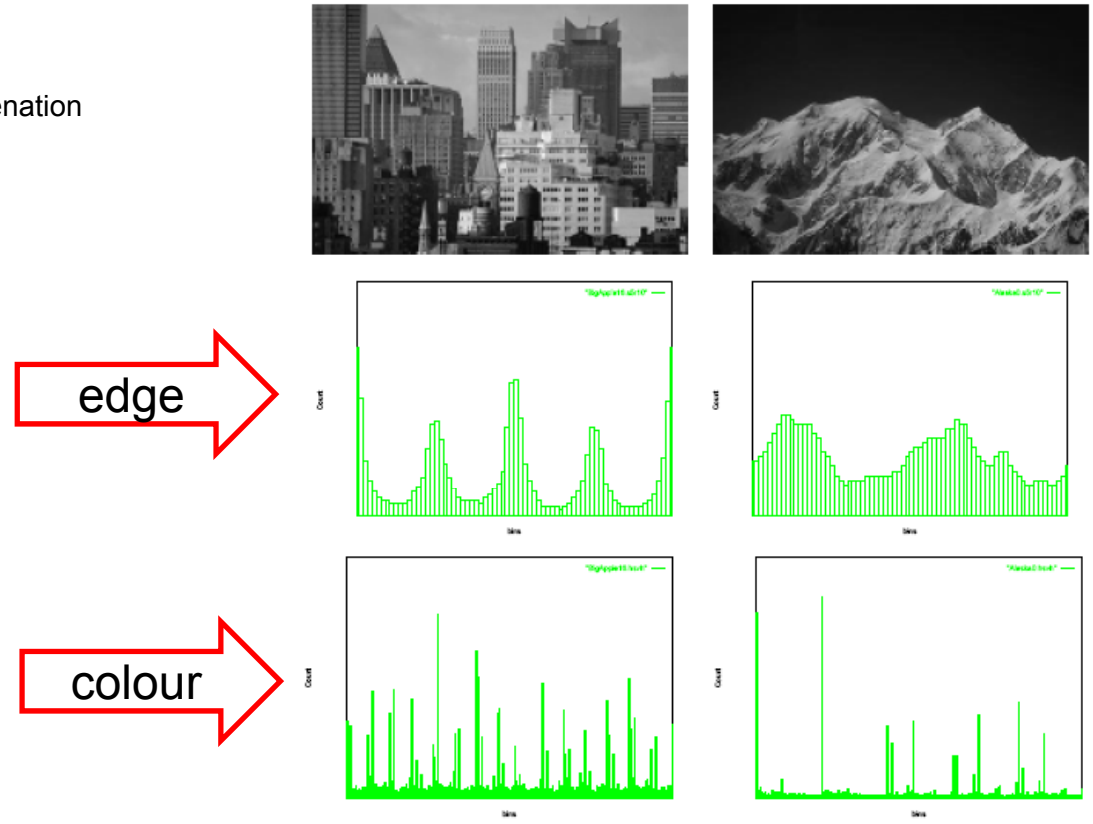
- Until now we have assumed feature concatenation to be simple, but it is not!
- The best feature to use for an image's description depends on what is its content:
  - For detecting different objects, different features may be required.
  - We do not know the content (we are trying to find it).
- Features can be combined by concatenation into a larger feature vector:
  - However, the features may have different "importance" for the image recognition system.

$$F_{fusion} = \alpha \times F_1 + (1 - \alpha) \times F_2$$

  - α is the fusion weighting, an additional hyper-parameter in the system which must be validated experimentally.

Global features Kumar figueiredo

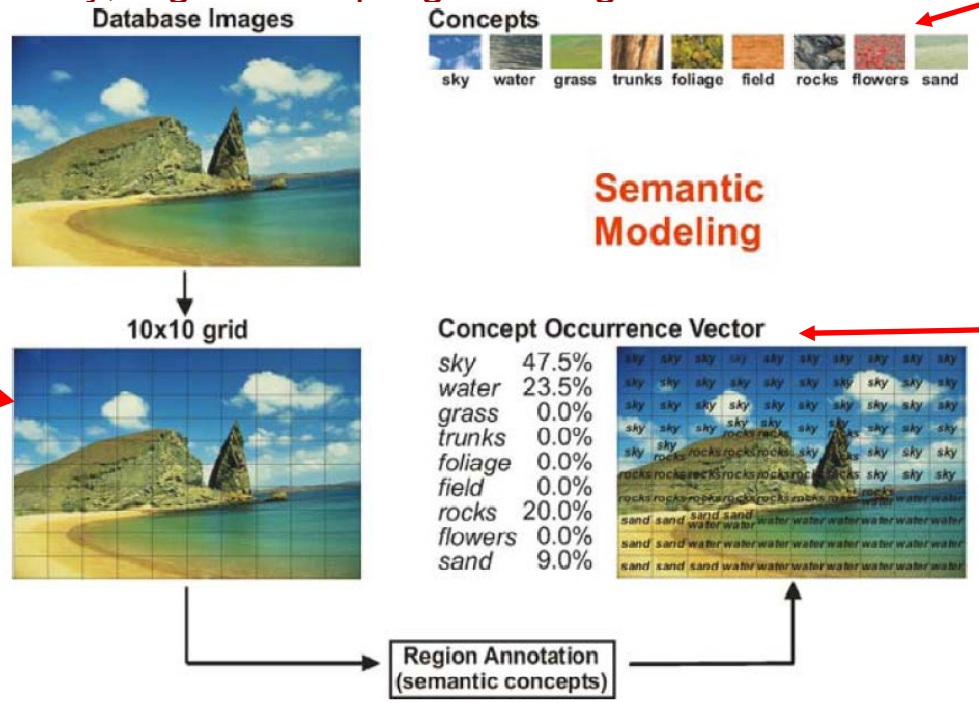
- City Landscape image classification:
  - Based on colour and edge histograms
  - KNN classifier
  - Features fusion using weighted concatenation



Vailaya, A. and Jain, A. and Zhang, H. J., On Image Classification: City vs. Landscape, IEEE Workshop on Content - Based Access of Image and Video Libraries,1998

Block based image classification

- If we assume repeatability, regular sampling is enough:



Mid-level features are extracted from each block

Each image block is decribed by:
-Greylevel co-occurence matrix
-color histograms
-color moments

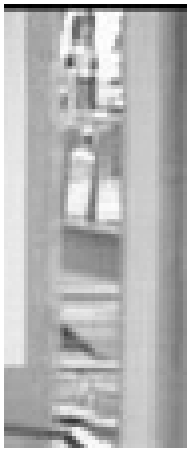
A histogram of occurrence of each mid-level feature is the final feature to be used in an SVM classification

Semantic Scene Modeling and Retrieval for Content-Based Image Retrieval. Julia Vogel and Bernt Schiele. *International Journal of Computer Vision*. Vol. 72, No. 2, pp. 133-157, April 2007.

Face and Pedestrian Detection

- **Features: Haar wavelet response (similar to the homogenous filter analysis in MPEG-7)**

- **Modelling:**



- **Apply the Haar wavelets to the blocks in several scales and locations**
- **Learn the important coefficients over the training dataset (response above a threshold)**
- **Train a classifier based on the chosen coefficients (SVM)**

"A general framework for object detection," by C. Papageorgiou, M. Oren and T. Poggio, Proc. Int. Conf. Computer Vision, 1998, copyright 1998, IEEE

Face and Pedestrian Detection

- **Learning the important wavelet coefficients:**

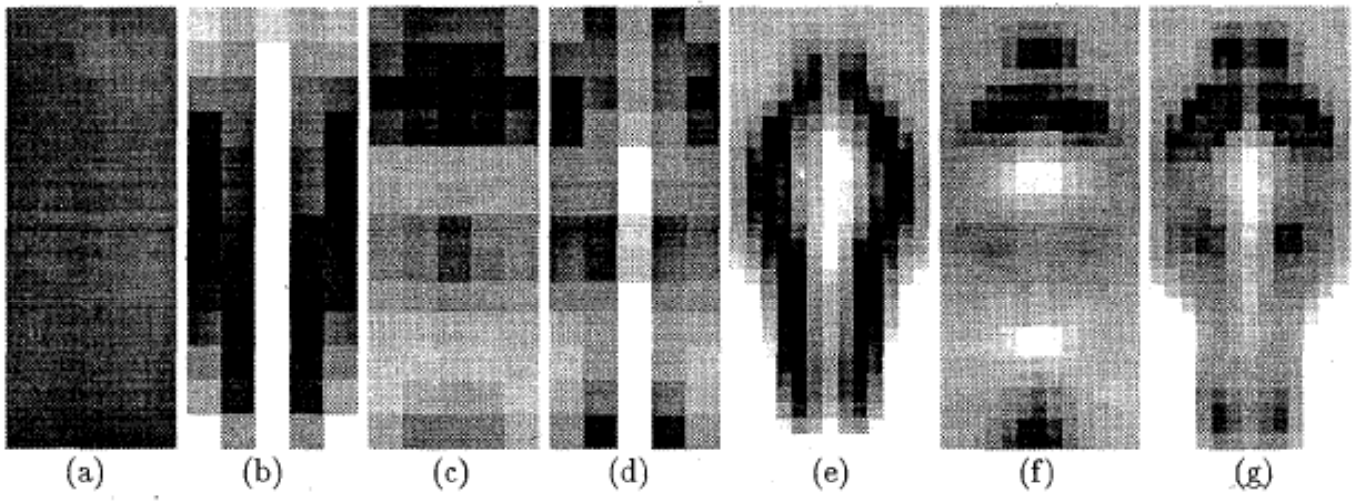


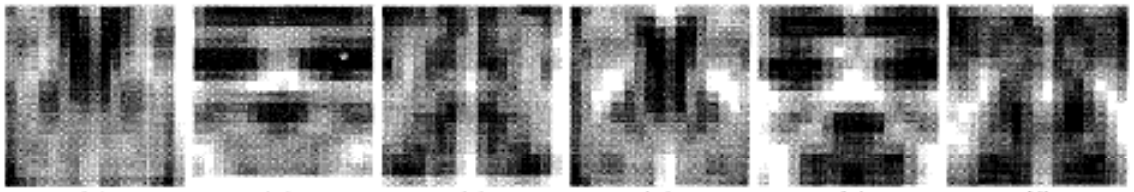
Figure 6: Ensemble average values of the wavelet coefficients coded using gray level. Coefficients whose values are above the template average are darker, those below the average are lighter. (a) vertical coefficients of random scenes. (b)-(d) vertical, horizontal and corner coefficients of scale 32×32 of images of people. (e)-(g) vertical, horizontal and corner coefficients of scale 16×16 of images of people.

- **The darker average responses indicate locations in the window where the specific wavelet is an important feature**
- **Given a window to classify all important responses are inputed into an SVM.**

Face and Pedestrian Detection

- **The same can be done with faces:**

Figure 3: Examples of faces used for training. The images are gray level of size 19 × 19 pixels.



(a) (b) (c) (d) (e) (f)

Figure 4: Ensemble average values of the wavelet coefficients for faces coded using color. Each basis function is displayed as a single square in the images above. Coefficients whose values are close to the average value of 1 are coded gray, the ones which are above the average are coded using red and below the average are coded using blue. We can observed strong features in the eye areas and the nose. Also, the cheek area is an area of almost uniform intensity, ie. below average coefficients. (a)-(c) vertical, horizontal and diagonal coefficients of scale 4 × 4 of images of faces. (d)-(f) vertical, horizontal and diagonal coefficients of scale 2 × 2 of images of faces.

Bootstrapping the training data

- Unbalanced set problem:
  - In the case of pedestrian and face detection, the dataset has a large number of negative examples we want to train with. However it has only a small amount of positive examples.
  - **BootStrapping:** Training is performed and classifier is used on non-pedestrian images. Detections are then used as negative examples for the classifier re-training.

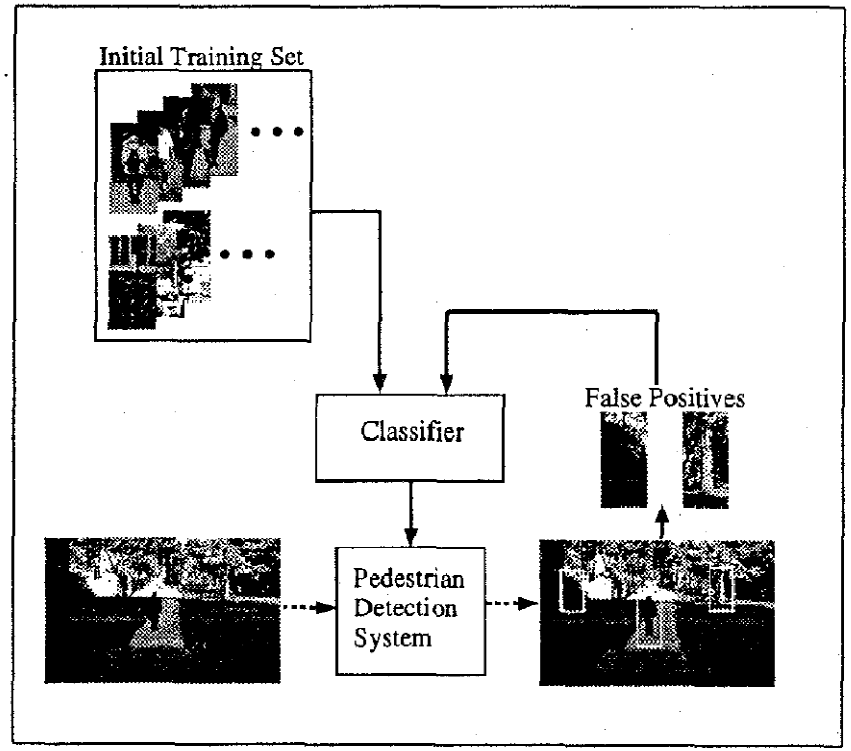
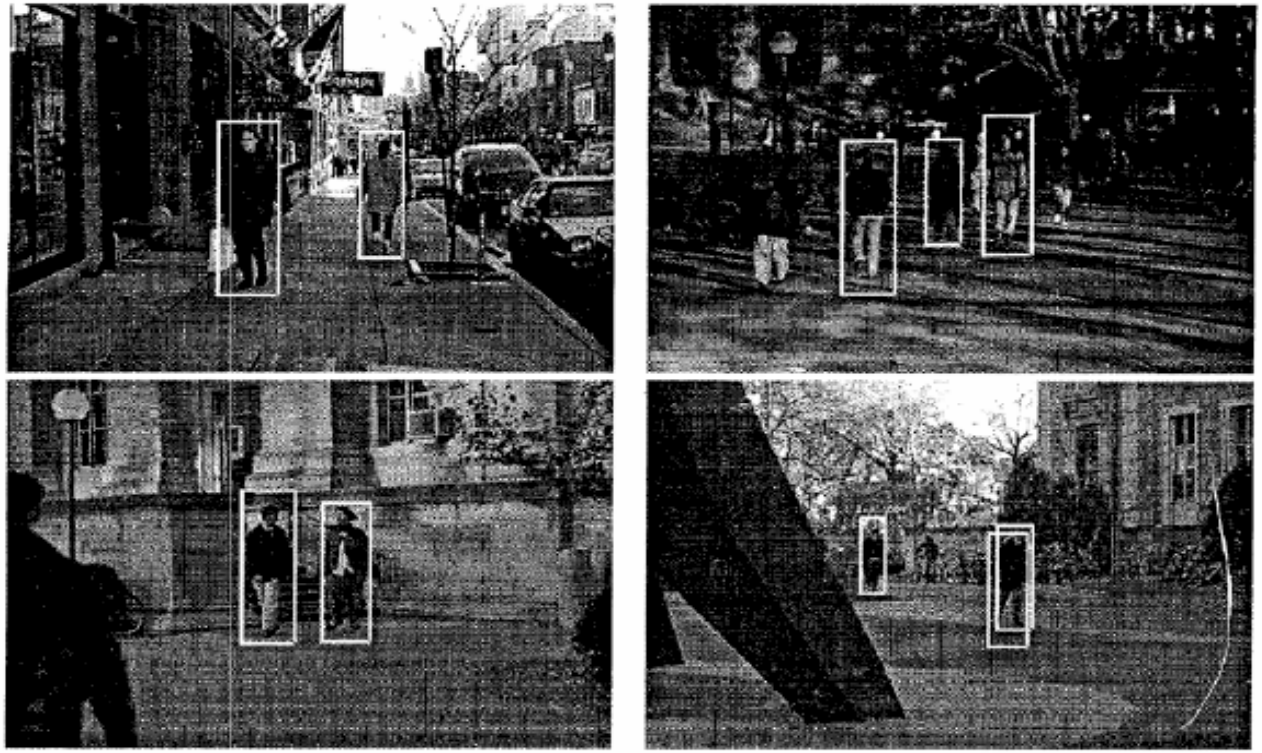


Figure 8: Incremental bootstrapping to improve the system performance.

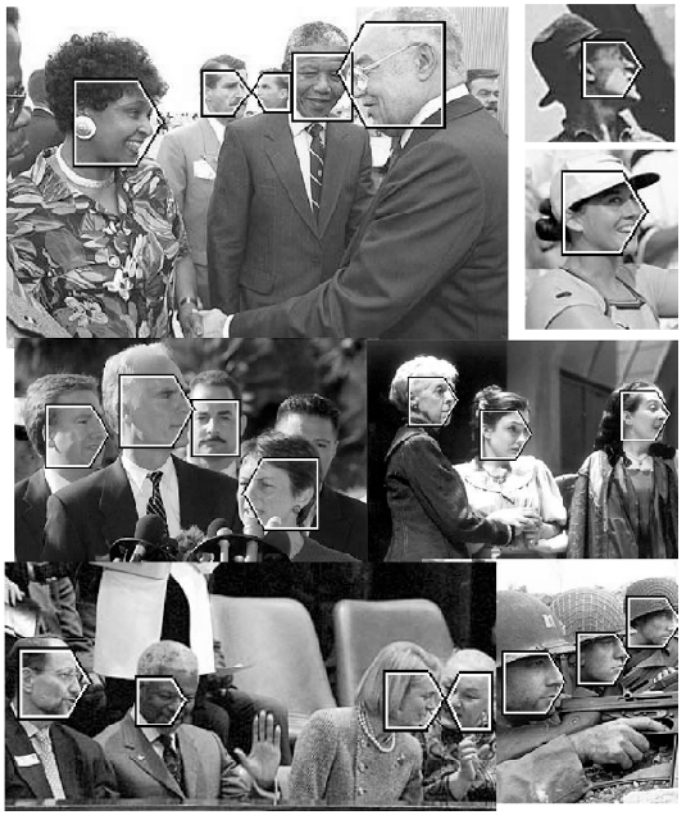
Face and Pedestrian Detection

- **Examples:**



More face detection

- Example for face detection:
  - Schneiderman and Kanade, “A Statistical Method for 3D Object Detection Applied to Faces and Cars”
  - Image patch described using a wavelet decomposition.
  - Wavelet coefficients are inputs to a classifier.
- All detections are independent.
- Features are extracted once and several classifiers indicate if the face is frontal or side (left/right).
- More robust and versatile than using direct pixel information.

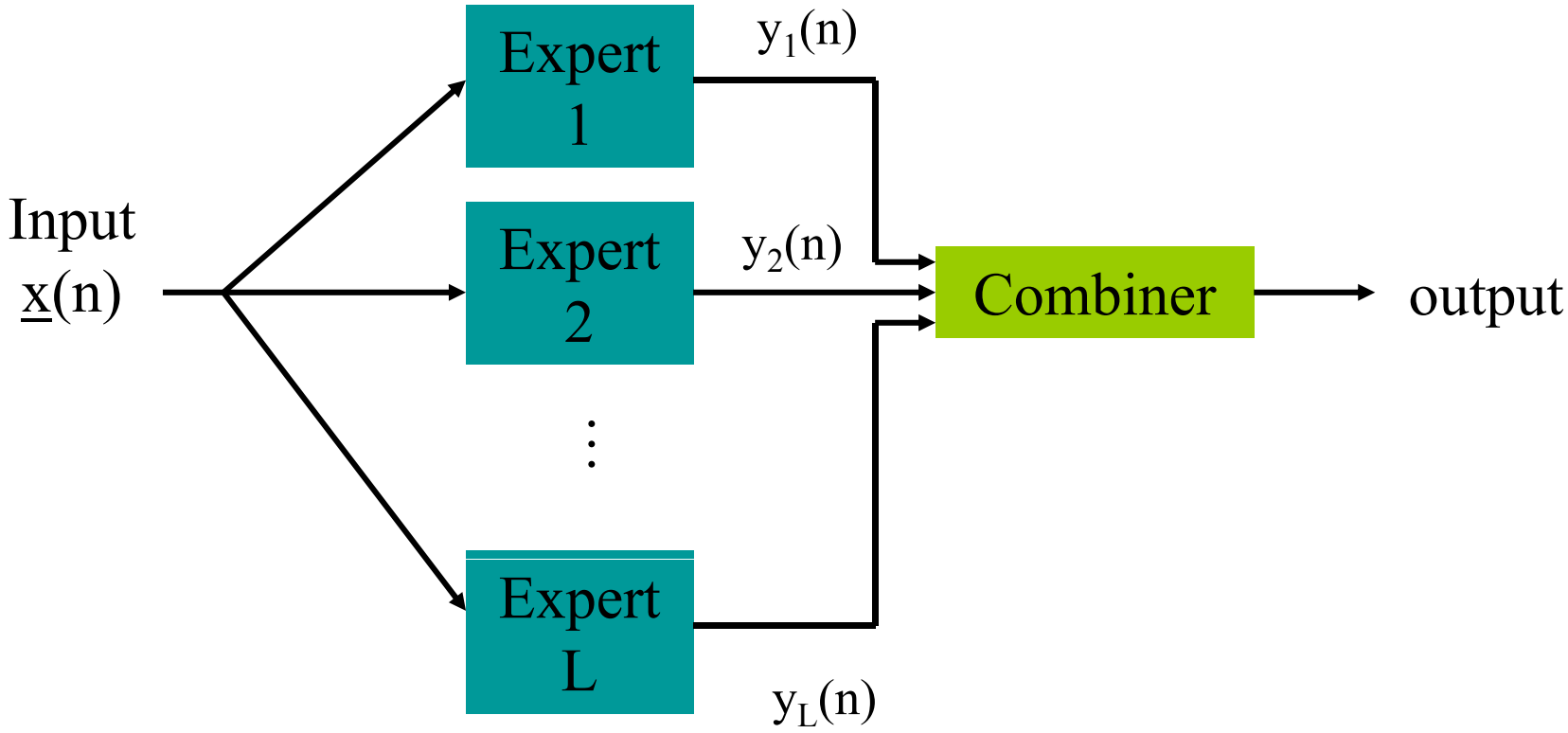


Combining classifiers

- There are many classifiers. There is no reason to believe one is in general superior to others. It is much more likely that some mixture of classifiers can perform better for a particular problem.
- Resulting classification result is obtained as a combination of each classifiers output
- Combination rules
  - average
  - product
  - voting
  - supervised combination (new classifier)
- We can use different classifiers with the same features, the same classifiers with different features (a different way to combine multiple features – late fusion)
- There is also the possibility to use the same classifier with the same features but using diferent instances to train each one (bagging).

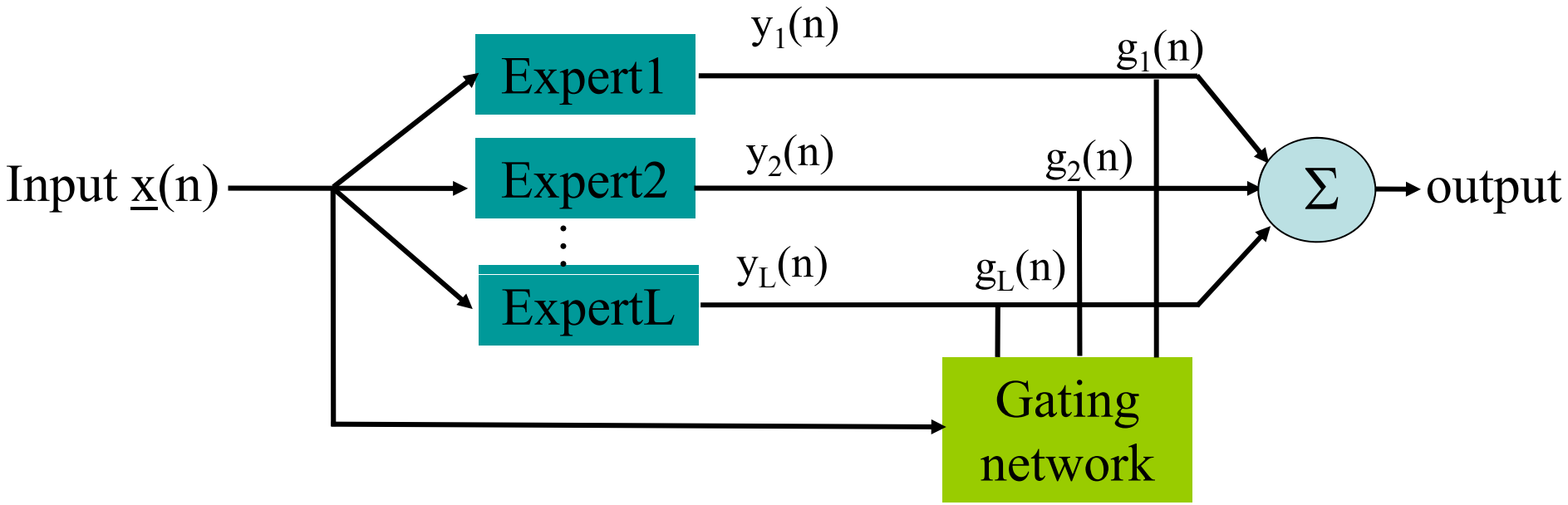
Combining classifiers

- Ensemble methods, mixture of experts or (Dynamic) committee machines.



Combining classifiers

- Dynamic committee machines:
  - Input signal is directly involved in combining outputs
  - Mixtures of experts and hierarchical mixtures of experts



- Gating network decides the weighting of each network

Boosting

- Boosting is a general approach for improving the accuracy of any given learning algorithm.
- The focus of these methods is to produce a series of classifiers. The training set used for each member of the series is chosen based on the performance of the earlier classifier(s) in the series.
- In Boosting, examples that are incorrectly predicted by previous classifiers in the series are chosen more often, or weighted more, than examples that were correctly predicted. Thus Boosting attempts to produce new classifiers that are better able to predict examples for which the current ensemble's performance is poor.
- Boosting is closely related to ensemble methods and classifier bagging.

AdaBoost algorithm

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$.
- Update:

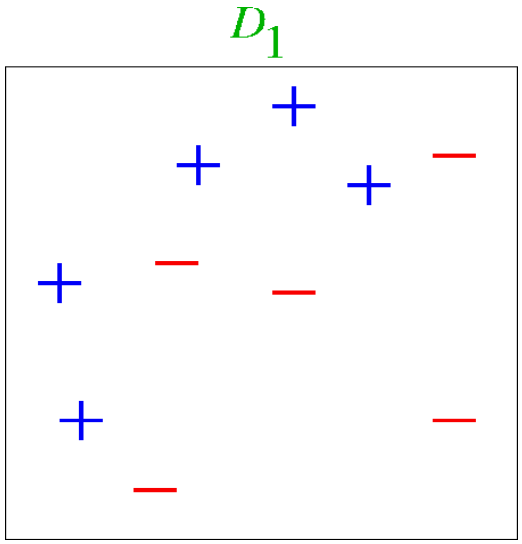
$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

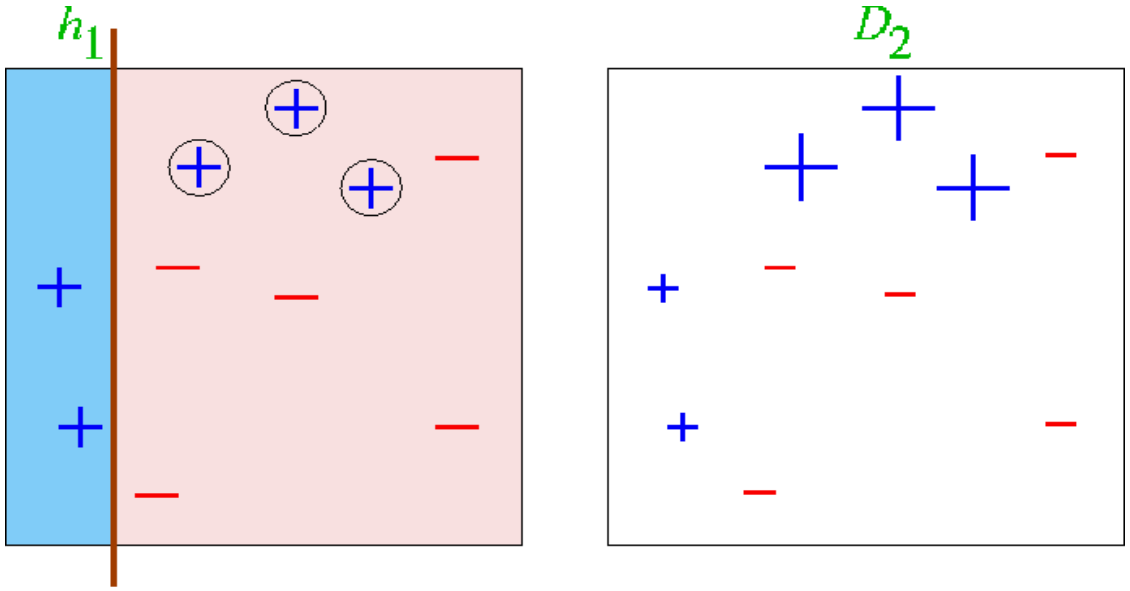
Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

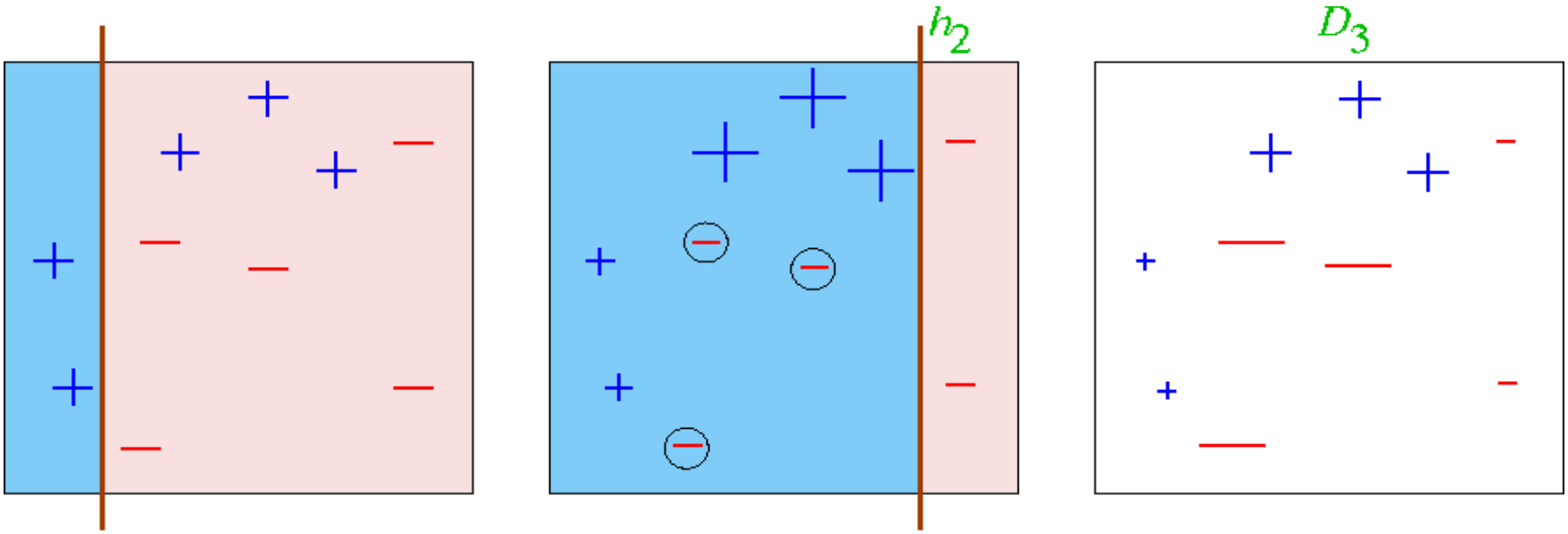
Boosting Example



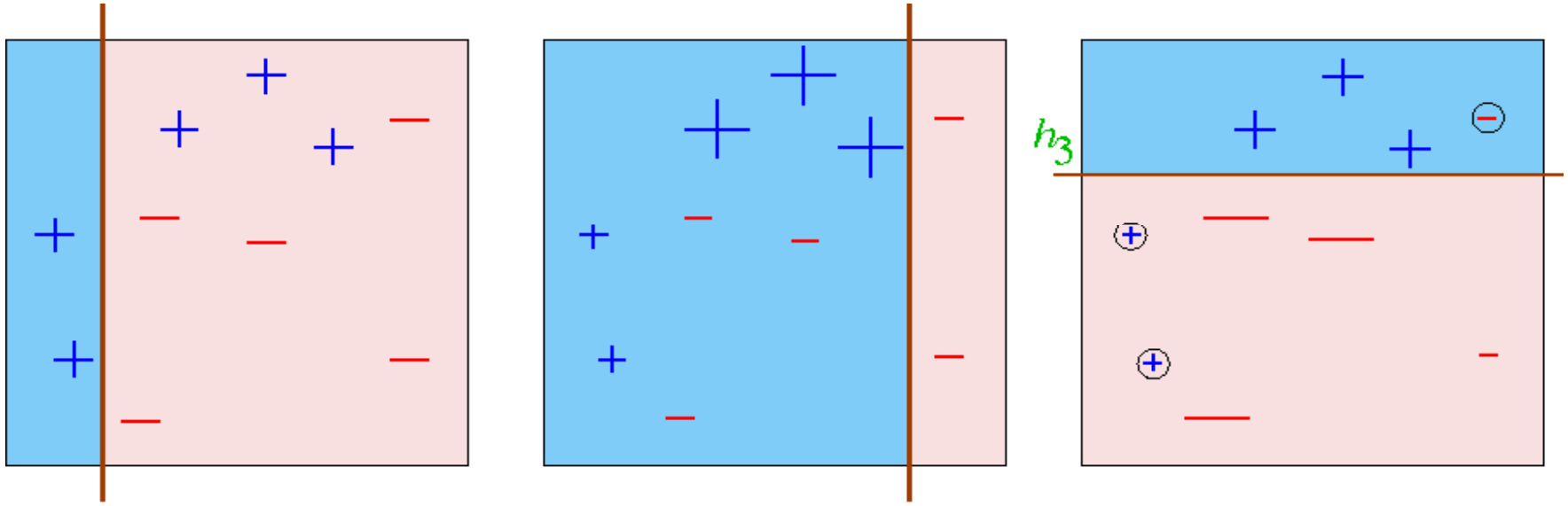
First classifier



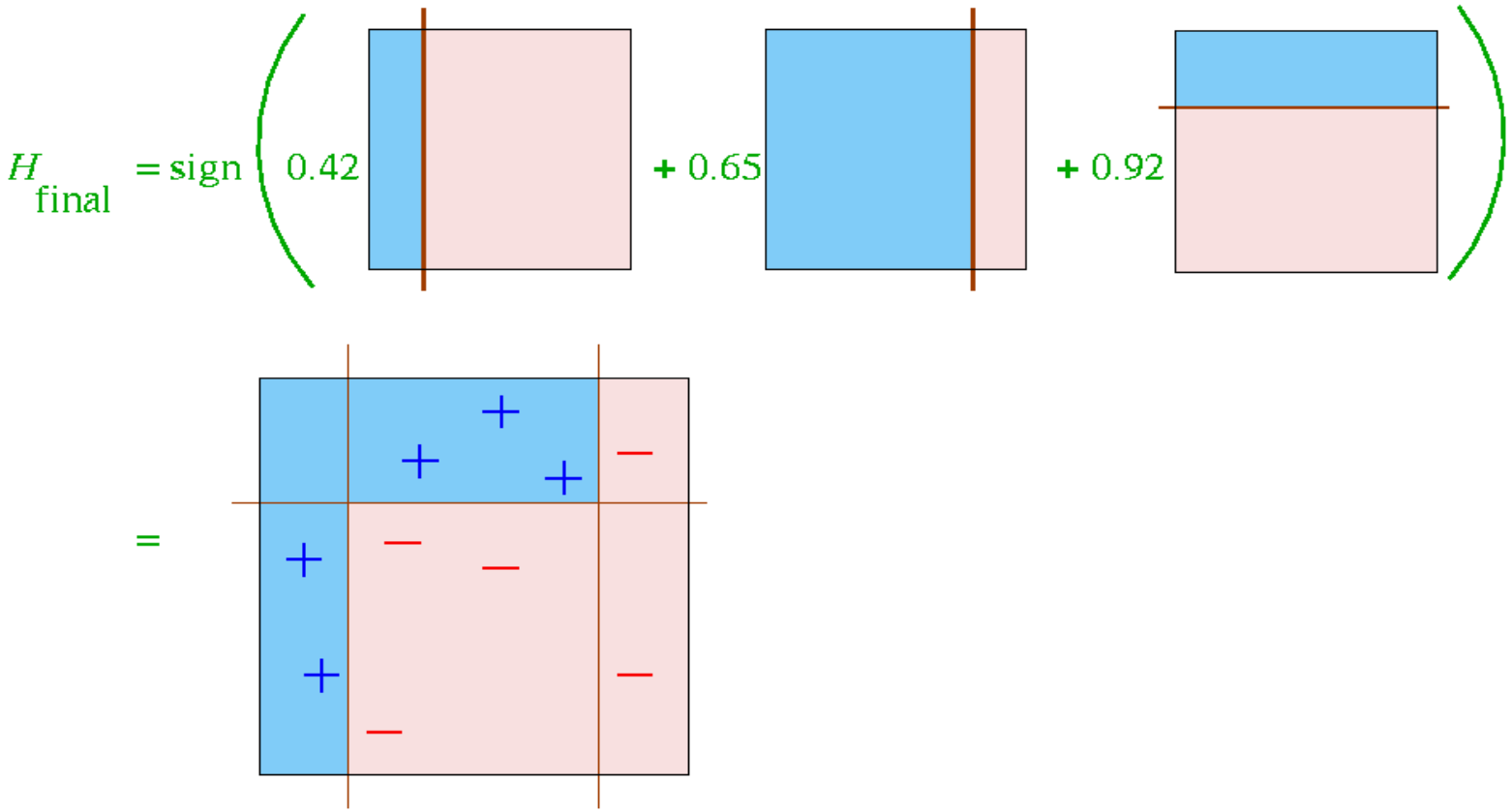
First 2 classifiers



First 3 classifiers



Final Classifier learned by Boosting



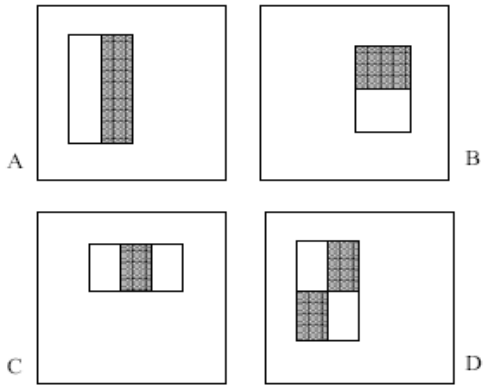
Viola Jones Face Recognition

- Three major contributions/phases of the algorithm :
  - Feature extraction
  - Classification using boosting
  - Multi-scale detection algorithm
- Feature extraction and feature evaluation.
  - Rectangular features are used, with a new image representation their calculation is very fast.
- Classifier training and feature selection using AdaBoost.
- A combination of simple classifiers is very effective

"Robust Real-Time Face Detection", Paul Viola, Michael J. Jones International Journal of Computer Vision 57(2), 137–154, 2004

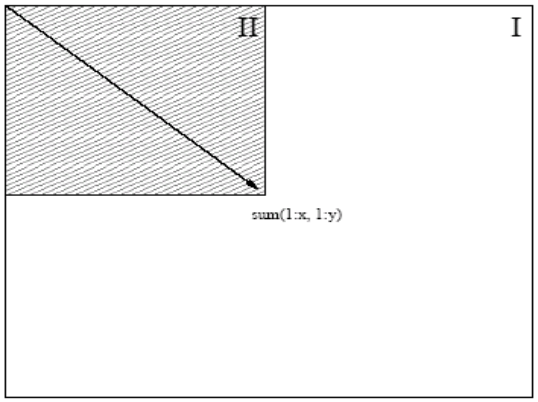
Features

- Extracted from a block of the image (obtained through search):
  - Four basic features (similar to wavelets)
  - They are easy to calculate.
  - The white areas are subtracted from the black ones.
  - A special representation of the sample called the integral image makes feature extraction faster.



Integral Image

- Summed area tables



- A representation that means any rectangle's values can be calculated in four accesses of the integral image.

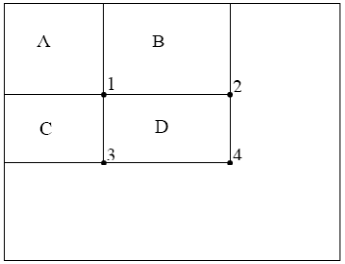
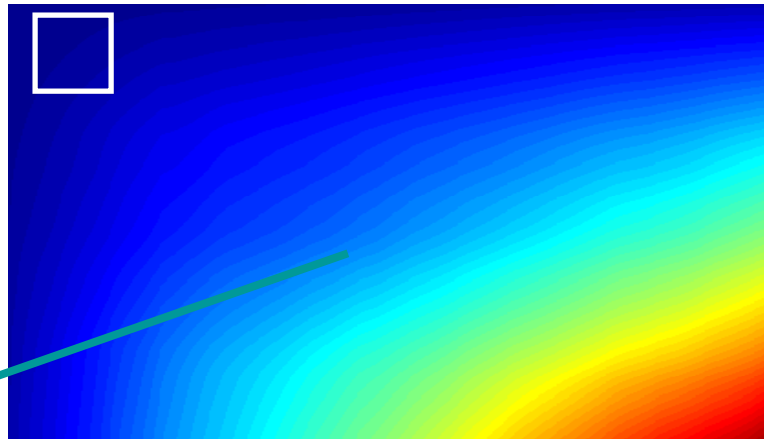
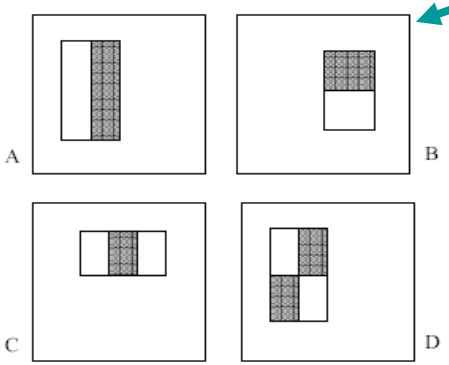


Figure 3: The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is $A + B$, at location 3 is $A + C$, and at location 4 is $A + B + C + D$. The sum within D can be computed as $4 + 1 - (2 + 3)$.

Feature Extraction





Detector response is calculated over several locations in the block and at several scales

Recall: Perceptron Operation

- Equations of “threshold” operation:
  - Output of a single perceptron is the signal of the weighted sum of the inputs.

$$o(x_1, x_2, \ldots, x_{d-1}, x_d) = 1 \quad (\text{if} \quad w_1 x_1 + \ldots w_d x_d + w_{d+1} > 0)$$

$$= -1 \quad (\text{otherwise})$$

Perceptron with just a Single Feature

- Equations of “thresholded” operation:

$$o(x_1) = 1 \quad (\text{if } w_1 x_1 + w_{d+1} > 0)$$

$$= -1 \quad (\text{otherwise})$$

- Equivalent to $x_1 > - w_{d+1} / w_1$

  i.e., equivalent to comparing the feature to a threshold

  Learning = finding the best threshold for a single feature

  Can be trained by gradient descent (or direct search)

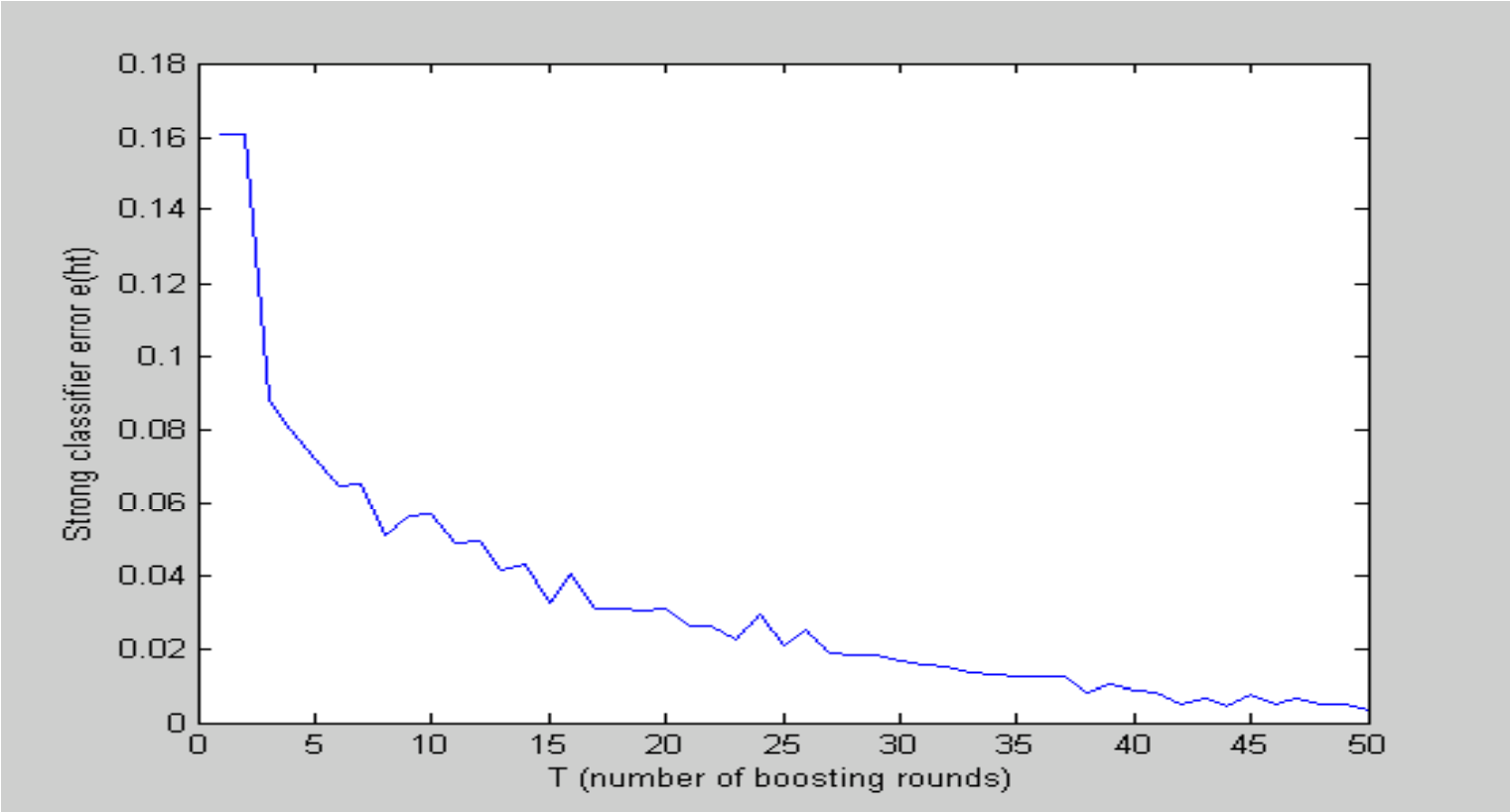
Boosting with Single Feature Perceptrons

- Viola-Jones version of Boosting:
  - "simple" (weak) classifier = single-feature perceptron
  - With K features (e.g., K = 160,000) we have 160,000 different single-feature perceptrons

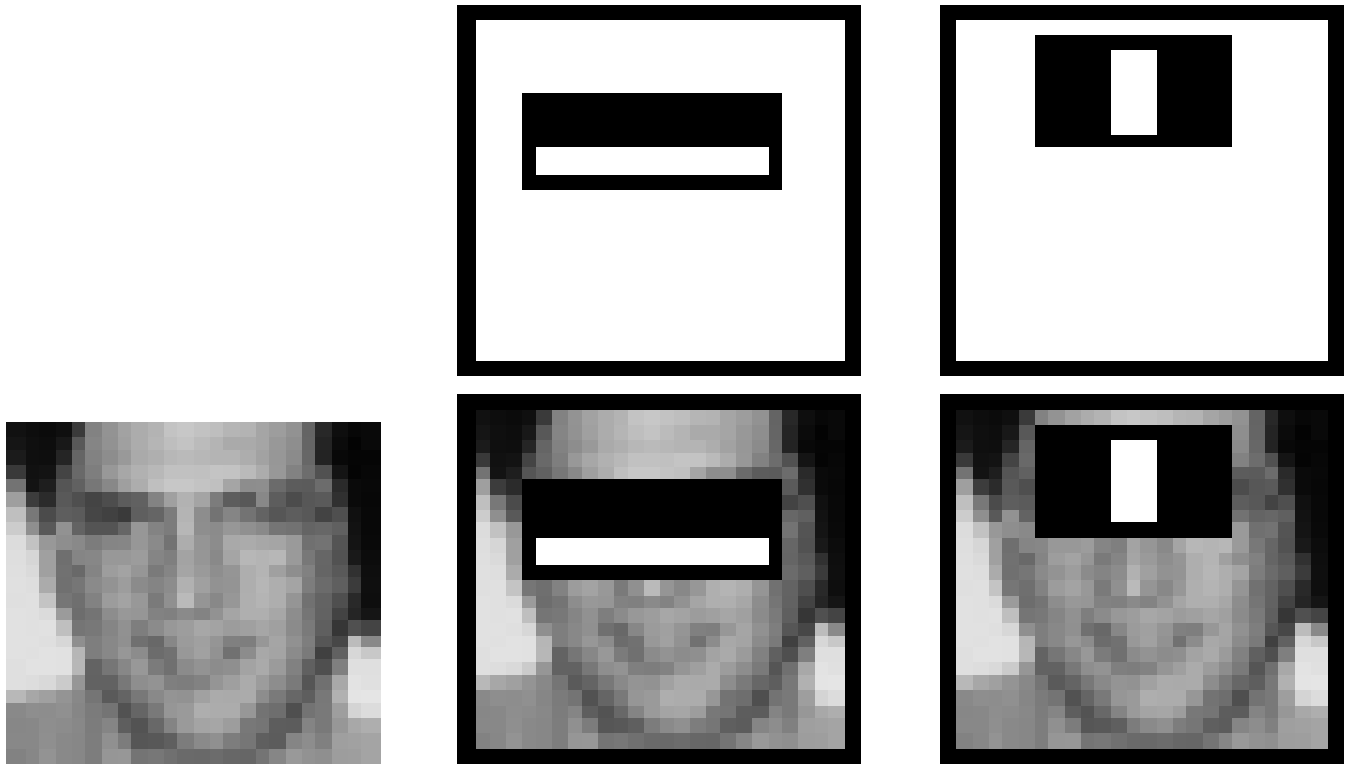
- At each stage of boosting
    - given reweighted data from previous stage
    - Train all K (160,000) single-feature perceptrons
    - Select the single best classifier at this stage
    - Combine it with the other previously selected classifiers
    - Reweight the data
    - Learn all K classifiers again, select the best, combine, reweight
    - Repeat until you have T classifiers selected

- Hugely computationally intensive
    - Learning K perceptrons T times
    - E.g., K = 160,000 and T = 1000

Reduction in Error as Boosting adds Classifiers

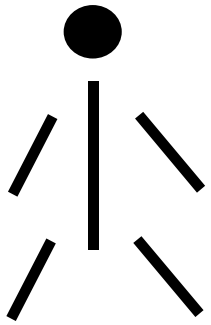


Useful Features Learned by Boosting



Part modelling

- An object can be further divided into parts:
  - Person:
    - Head, torso and limbs

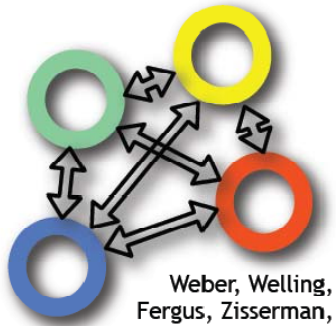


- Parts are easier to mode. However:
  - Miss-detection of parts can occur more frequently
  - To obtain the detection of the object we need to introduce and extra level of modelling: part spatial modelling

Constellation Object Model

- **Geometric relations between object parts**
  - Object parts do not occur independently
  - The spatial relation between parts can validate or eliminate object recognition hypothesis.

Constellation Model:
Fully connected shape model

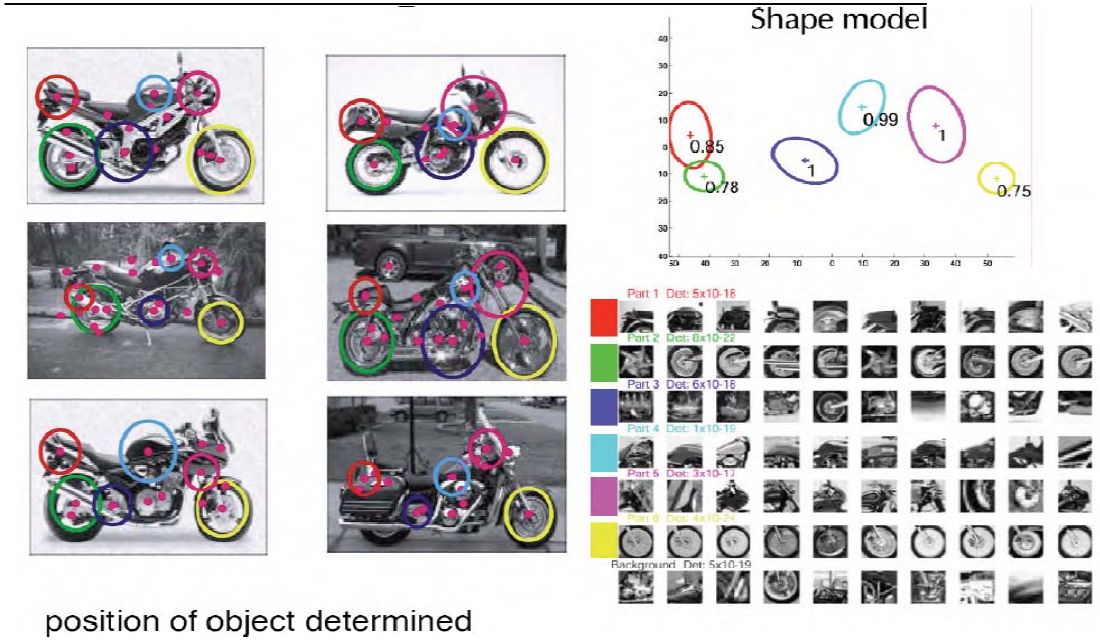


Weber, Welling, Perona '00
Fergus, Zisserman, Perona '03

  - Part's aspect and location is learned from labelled data.

Constellation Object Model

Motorbike Example [Perona, Caltech]



position of object determined

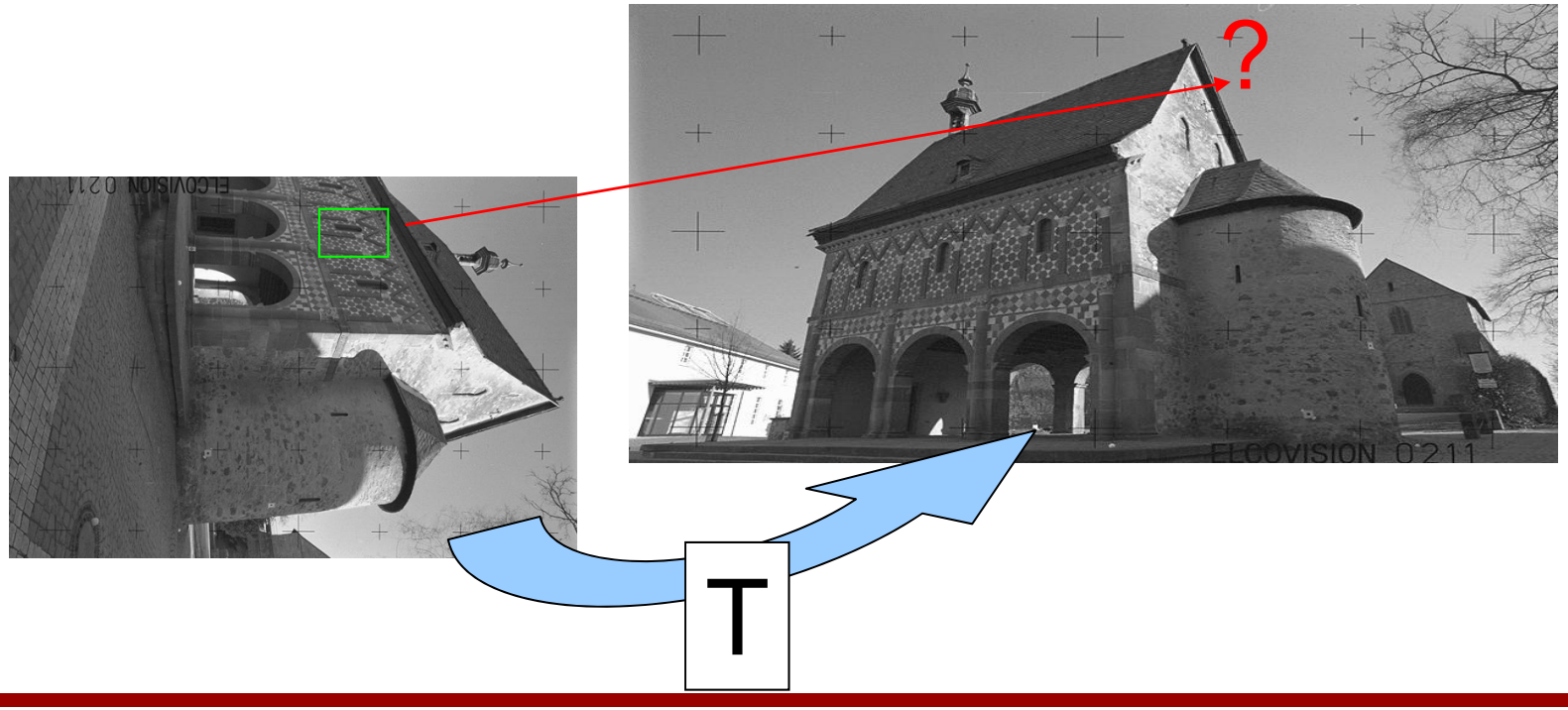
Position of parts (GMMs)

Parts apearence (PCA)
(Exhautive search detection)

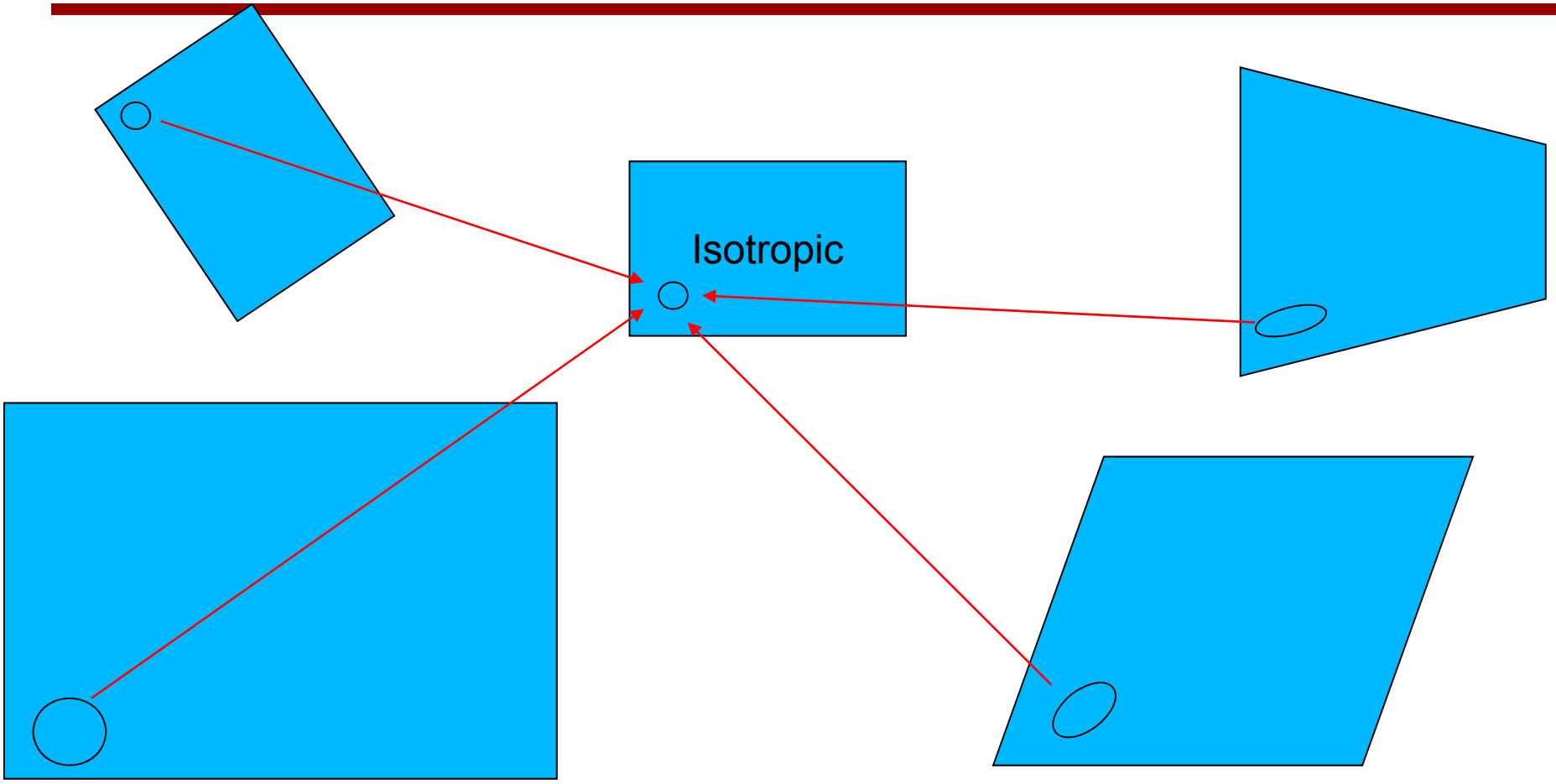
- **Problem: Objects must be framed and appear from the same viewpoints**

wide-baseline matching

- Local point detectors were first created to help solve the wide-baseline matching problem.



Shape projection



Definition of Local Point Detectors/Descriptors

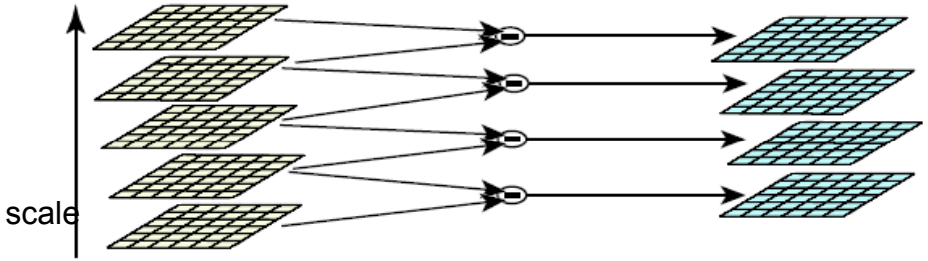
- Local Point Detectors
  - Detectors that “fire” at specific locations in the image
  - Define areas that are invariant to certain transformations
- Local Descriptors
  - Highly specific, must describe a local area with high discriminative power.

- Invariance to transformation can come from either the point detector or the local descriptor.

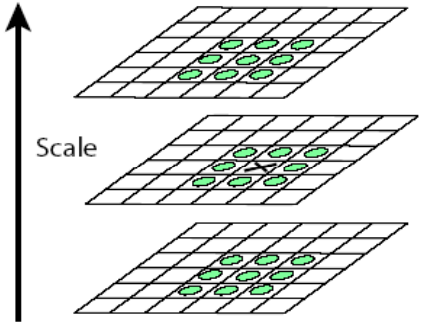
Anatomy of a Local Point Detector

- The DOG detector

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma). \end{aligned}$$



•Based on the search for local maxima and minima in the image.
•Detects blob like regions.

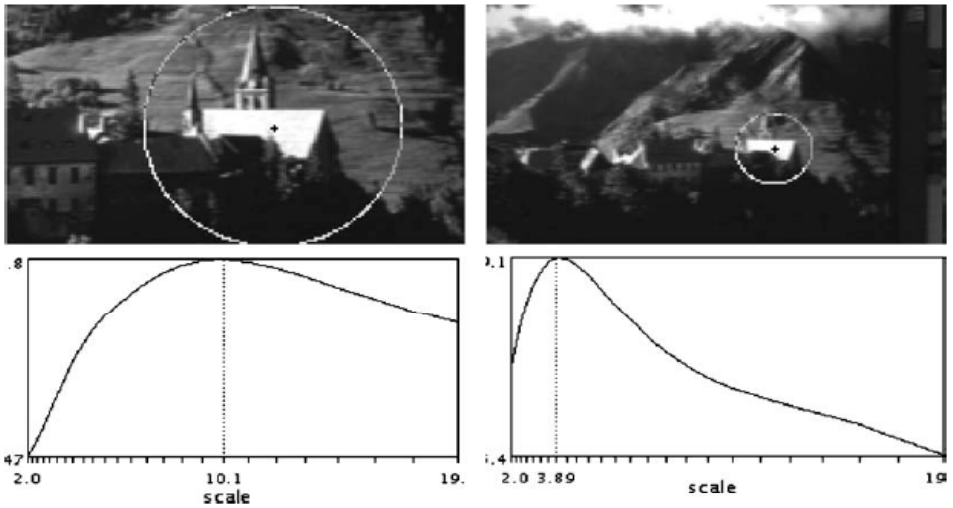


•Local maxima is searched in both space and scale neighborhood

Extending to Scale

- Automatic scale selection
  - We can find the specific scale for a point by using the maximum of the Laplacian of the Hessian = *Trace(H)*

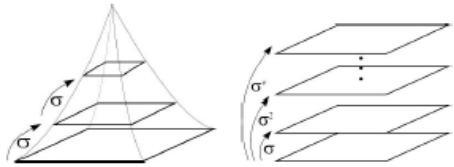
$$\text{Laplacian} \quad \sigma_D^2 |L_{xx}(\mathbf{x}, \sigma_D) + L_{yy}(\mathbf{x}, \sigma_D)|$$



Lindeberg, ``Feature detection with automatic scale selection". *IJCV*, V 30 (2), pp. 77-116, 1998

Detecting corners

- **The multi-scale Harris detector**



$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix}$$

$$= \sigma_D^2 \, g(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}$$

$$cornerness = \lambda_1 \times \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

$$cornerness = \det(\mu(\mathbf{x}, \sigma_\mathbf{I}, \sigma_\mathbf{D})) - \alpha \mathrm{trace}^2(\mu(\mathbf{x}, \sigma_\mathbf{I}, \sigma_\mathbf{D}))$$

•Based on the second moment matrix.
•The matrix describes the local gradient distribution.
•Eigen-values represent the change of the signal's strength in the local neighborhood.
•Based on this matrix we can extract point where the signal changes significantly in both orthogonal directions – corners…

Local maxima of cornerness define the local interest points

M. Krystian and C. Schmid, "Scale & Affine Interest Point Detectors", IJCV, V 60(1), p. 63-86, 2004.

C. Harris and M. Stephens, "*Combined corner and edge detector*", Proc. Fourth Alvey Vision Conference, pp. 147-151, 1988.

Non-isotropic scale

- The Affine-Harris detector

$$\Sigma_I = s\,\Sigma_D$$

$$\mu(\mathbf{x}, \Sigma_I, \Sigma_D) = \det(\Sigma_D)\, g(\Sigma_I) * ((\nabla L)(\mathbf{x}, \Sigma_D)(\nabla L)(\mathbf{x}, \Sigma_D)^T)$$

The second moment matrix can be used to estimate the anisotropic shape of the local neighborhood

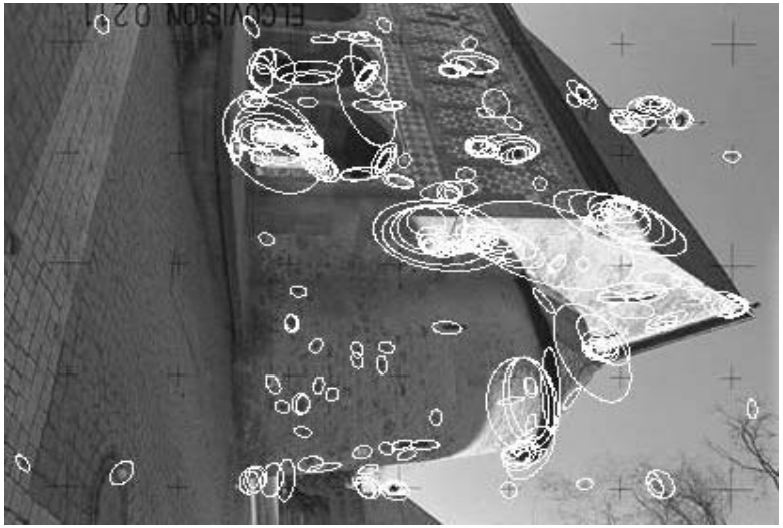
If Q is close to one then the local neighborhood is isotropic.

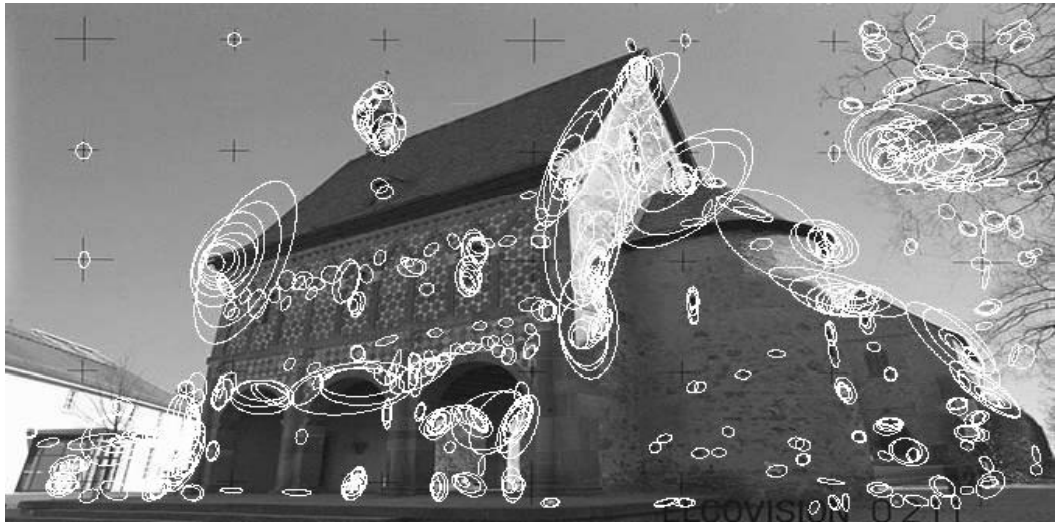
$$Q = \frac{\lambda_{\min}(\mu)}{\lambda_{\max}(\mu)}$$

We can transform any local neighborhood to its isotropic equivalent by using the Eigen-vectors and Eigen-values.

Lindeberg, ``Feature detection with automatic scale selection''. *IJCV*, V 30 (2), pp. 77-116, 1998

Affine multi-scale Harris detector results



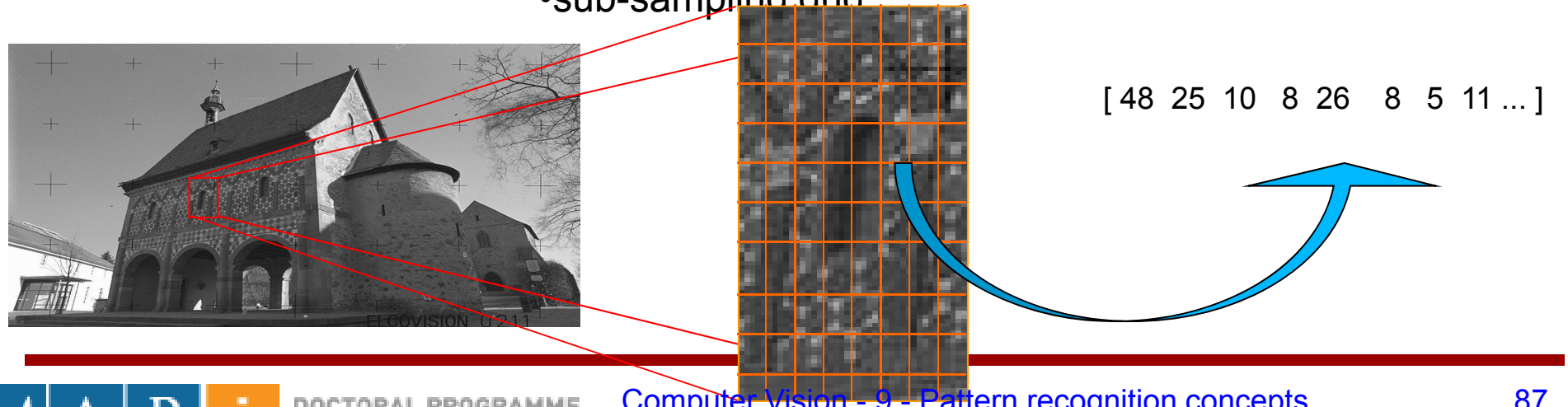


Local Descriptors

- Sampling the local image area

•Since local Point detectors provide invariance to rotation, scale and affine transformations. It has become an option to use the grayscale values of the image patch as a local descriptor.

•compensate for luminance variations
•sub-sampling grid

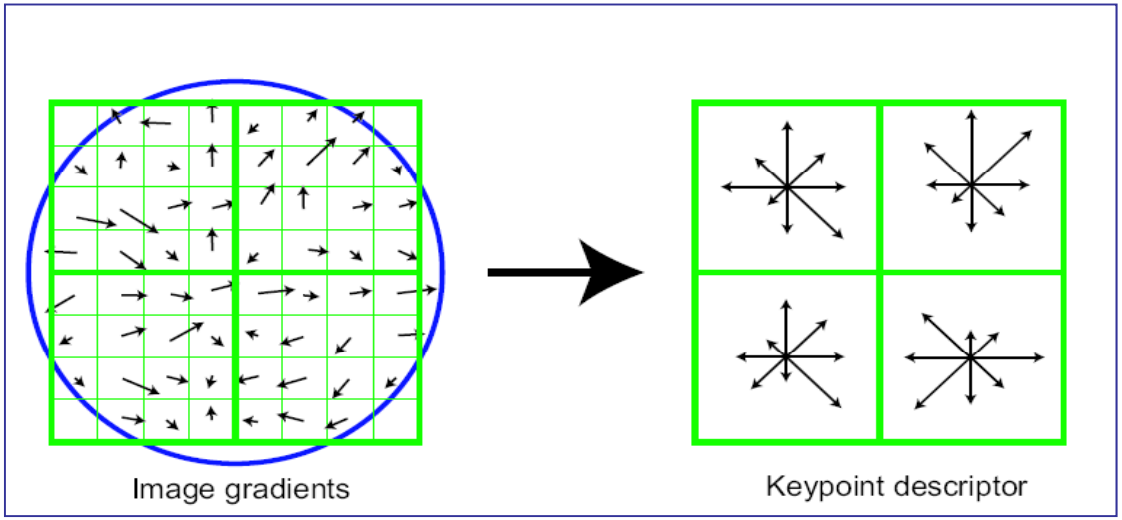


Local Descriptors

- SIFT descriptors

•Similar to a patch sub-sampling of the image. It is however performed in the gradient domain and it is stored as an histogram.

D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV, 60(2):91-110, 2003.



Constellation Object Model

- **Costellation modeling based on local descriptors:**
  - **Local descriptors are chosen to detect part's locations**
  - **Part learning is faster and more robust**
  - **Part's spatial relationship is still based on a frame of view (exhaustive search)**

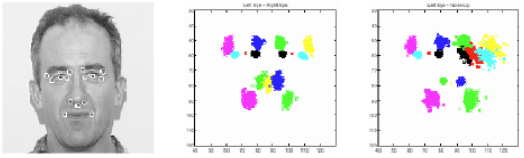
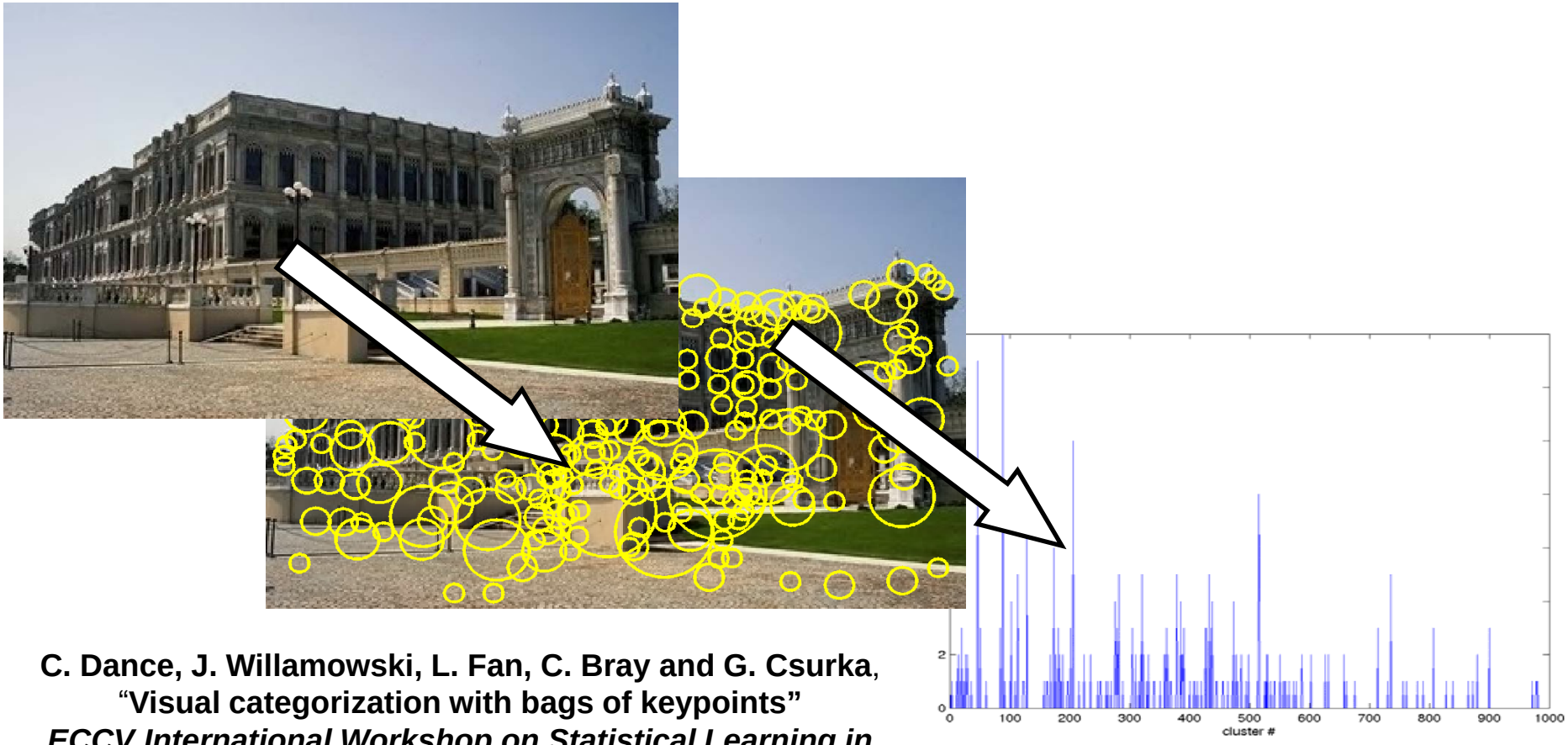


Figure 21.22. Perona's feature detector responses

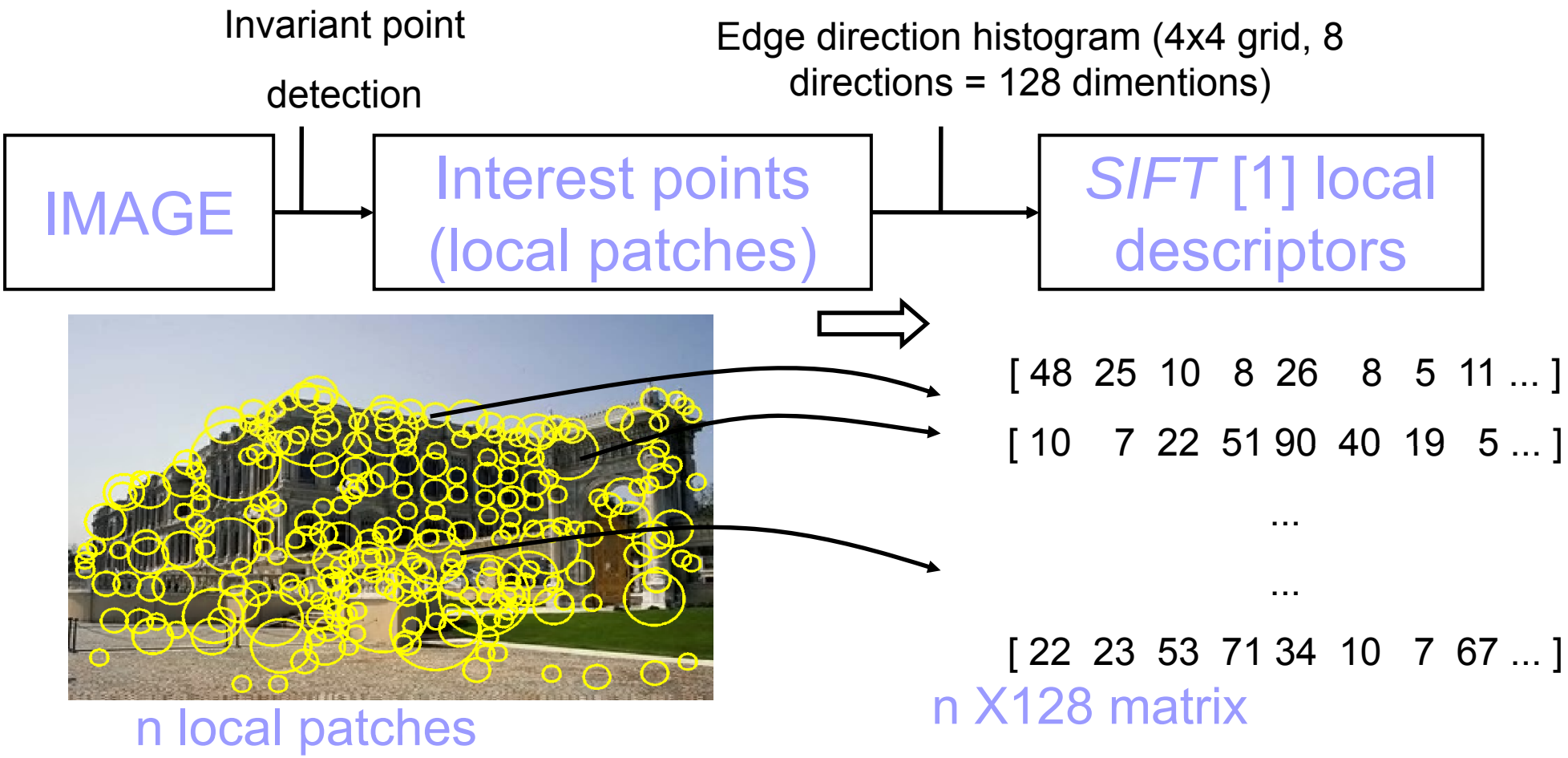
Figure 21.23. Perona's feature detector outputs

Parts with no spatial relations

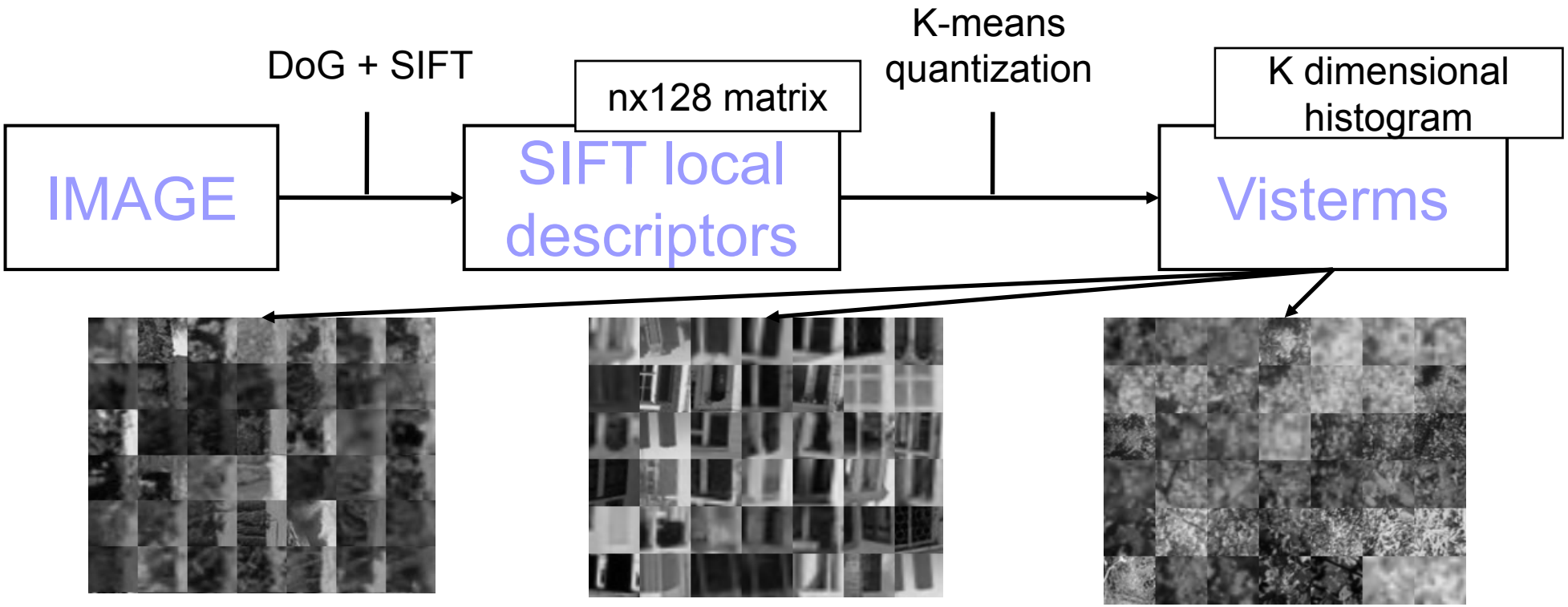


C. Dance, J. Willamowski, L. Fan, C. Bray and G. Csurka, "Visual categorization with bags of keypoints" ***ECCV International Workshop on Statistical Learning in Computer Vision , Prague, 2004.***

Local descriptors extraction

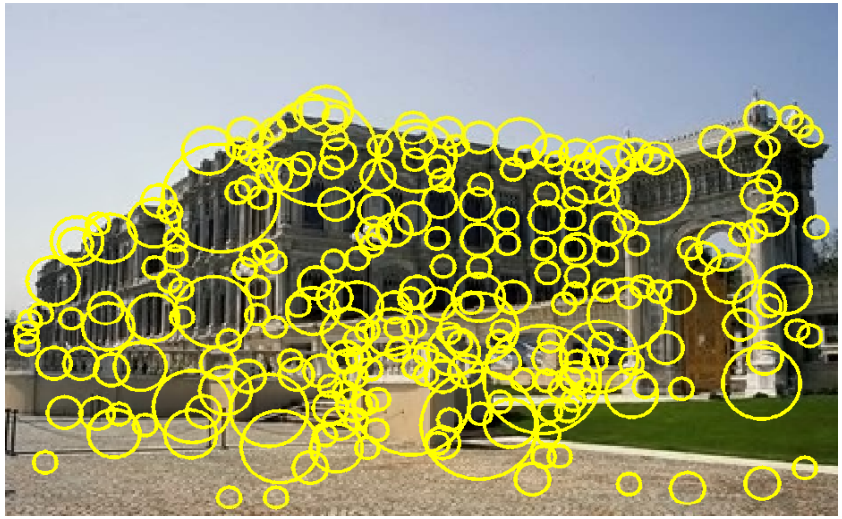


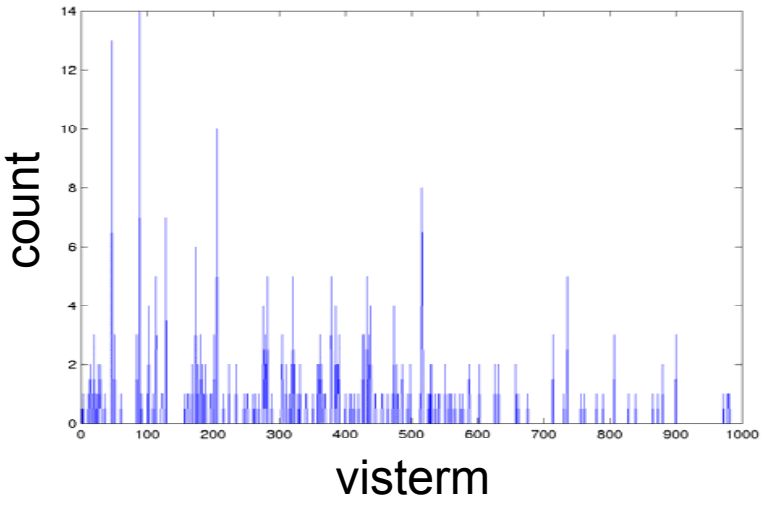
Quantizing local descriptors



- Visterms ⇔ local image patterns
- Image is now represented by a visterm histogram
- K is the vocabulary size (chosen by cross validation, depends on task)

Bag-of-visterms representation





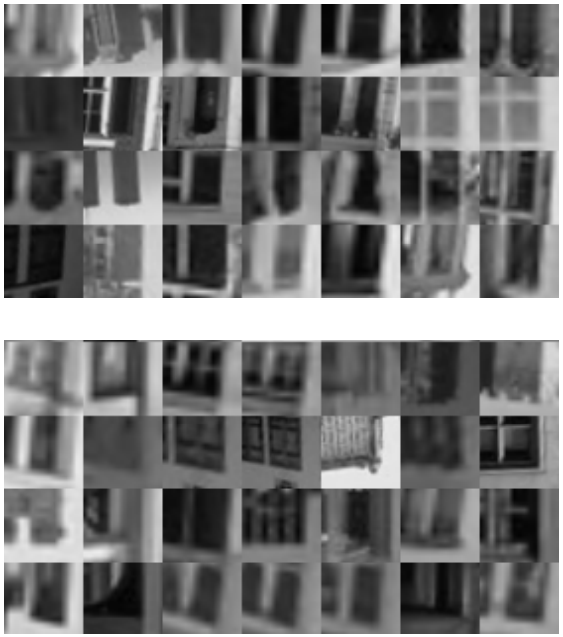
- Spatial information is discarded

$\Rightarrow$ analogy with text's bag-of-words:

bag-of-visterms(BoV)

Visterm ambiguity

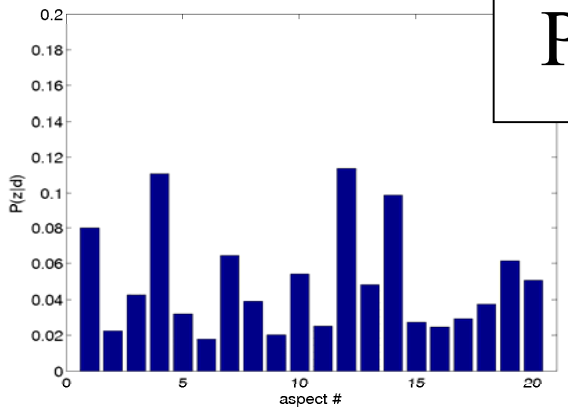
Synonymy



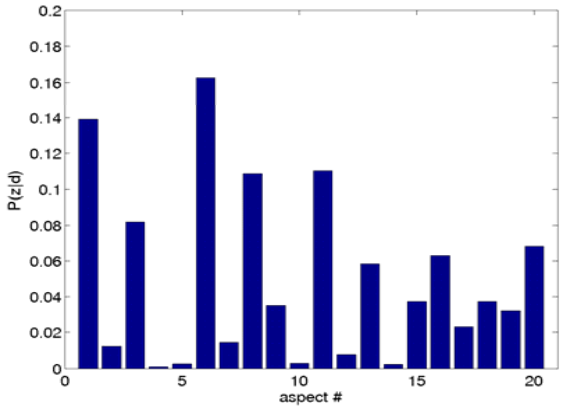
Polysemy

Disambiguating a representation from bag-of-visterms : PLSA

Aspect based image representation



$$P(z_k \mid d_i)$$



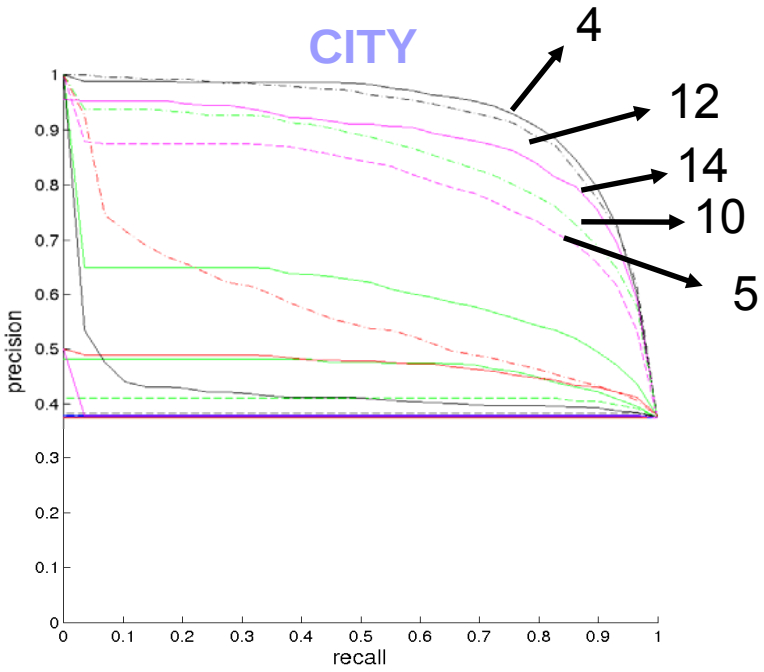
Aspect-based image ranking

- Given an aspect z_k, images can be ranked with respect to (for each aspect):

$$P(d \mid z_k) = P(z_k \mid d)P(d)/P(z_k)$$

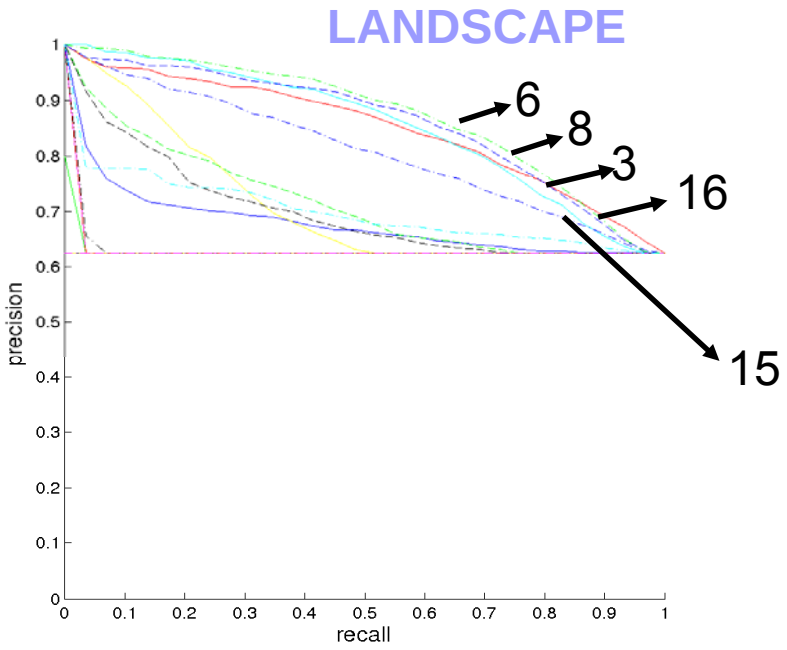
Precision/Recall

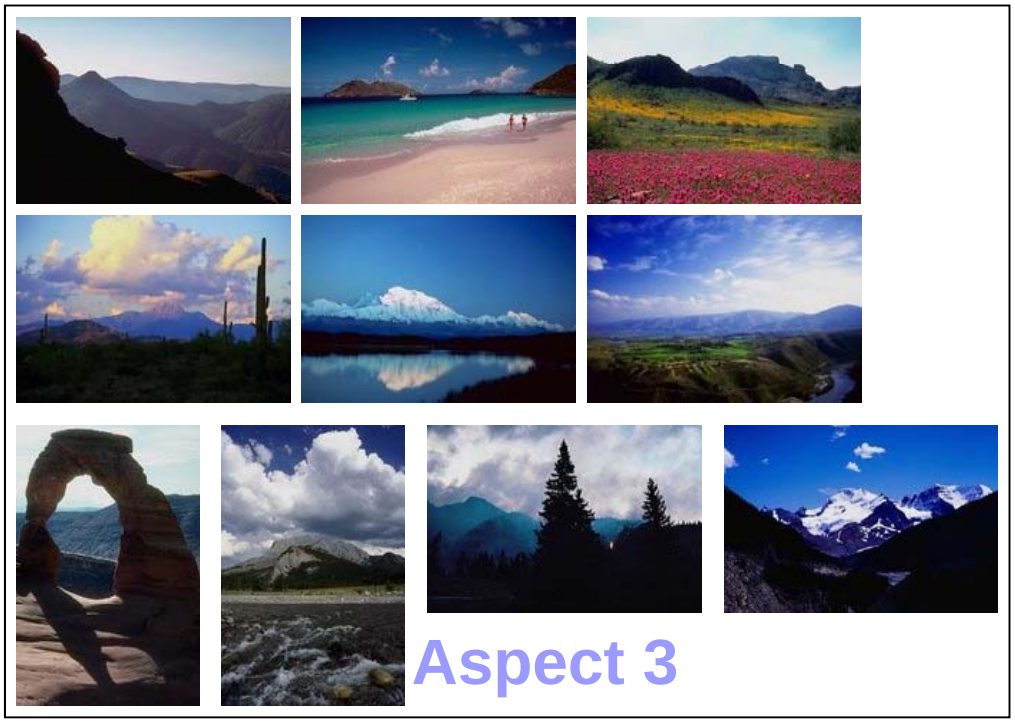
- Although not all, most aspects have a semantic meaning.



Precision/Recall

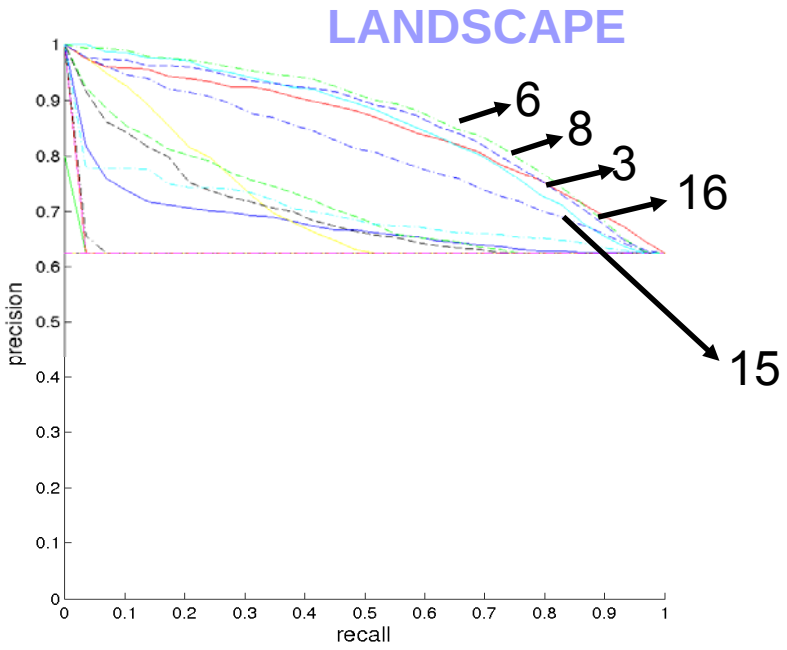
- Although not all, most aspects have a semantic meaning.





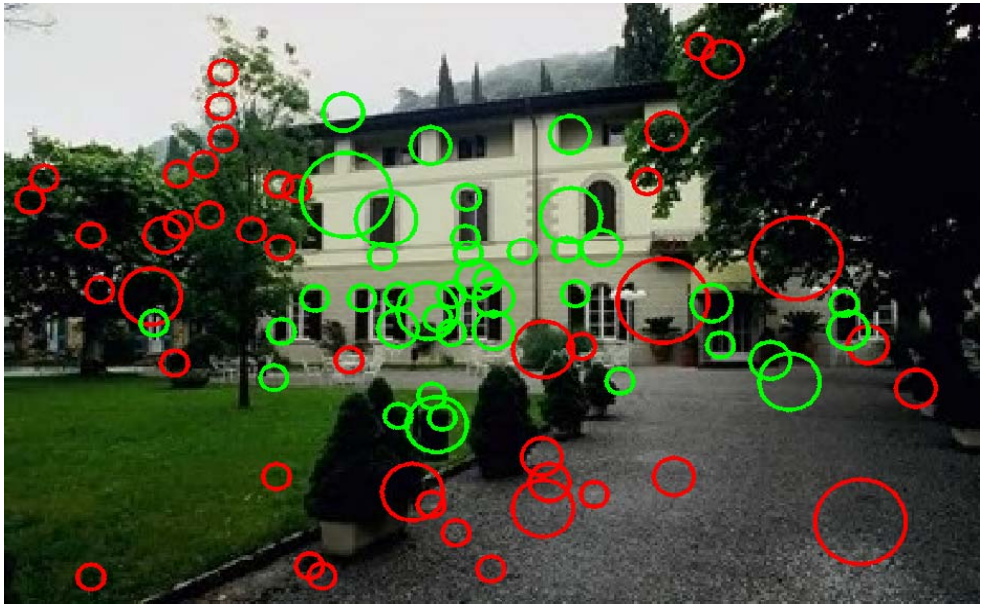
Precision/Recall

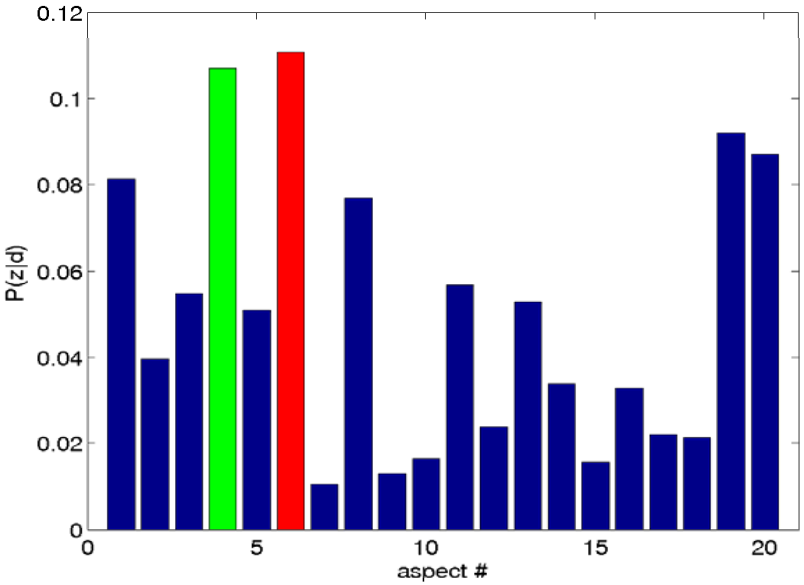
- Although not all, most aspects have a semantic meaning.



Aspect based segmentation

- Using $P(v_j \mid z_k)$ we can get segmentation of a scene.

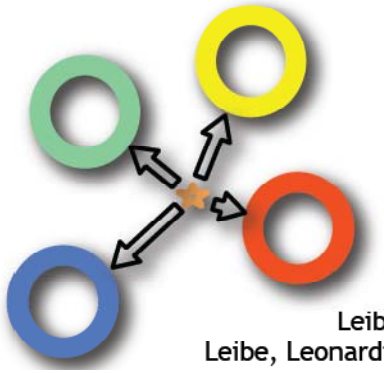




Implicit Shape Model

- **Object detection is based on voting by detected parts:**
  - Parts modelling is performed by local feature clustering
  - Part's spatial co-occurrence is learned from labelled images.
  - Object are detected using a framework similar to the generalized Hough transform.
- **Does not require exhaustive search for possible object positions.**
- **Requires supervised complex training (labelled data)**

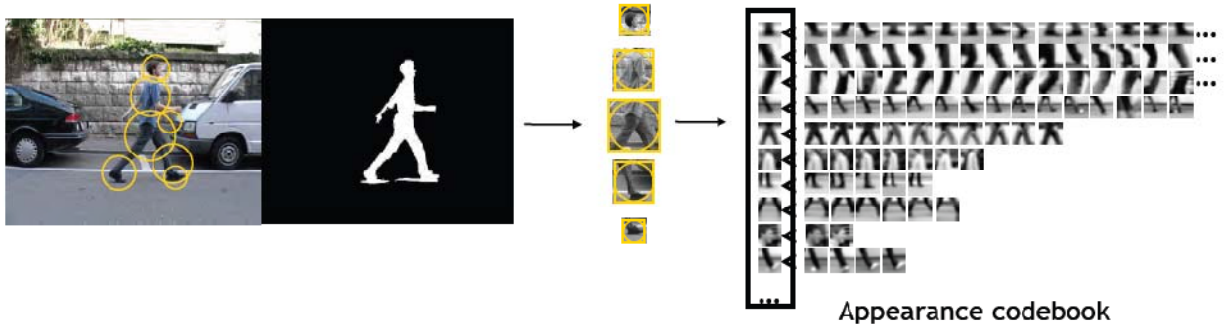
Implicit Shape Model:
Star-Model w.r.t. Reference Point



Leibe, Schiele '03
Leibe, Leonardis, Schiele '04

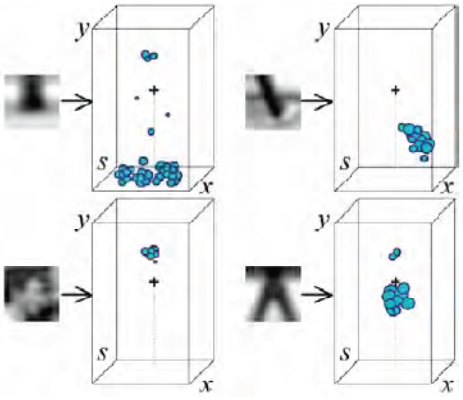
Implicit Shape Model

Implicit Shape Model - Representation



Appearance codebook

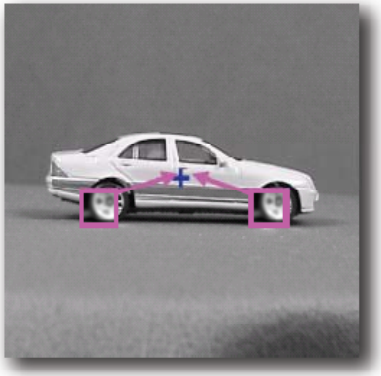
- Learn appearance codebook
  - Extract features at interest points (e.g. DoG)
  - Agglomerative clustering ⇒ codebook
- Learn codebook distributions (position & scale)
  - Match codebook to training images
  - Record matching positions on object



Spatial occurrence distributions

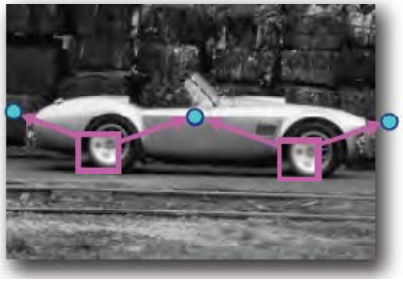
Implicit Shape Model

- For every codebook, store possible "occurrences"



- Relative position
- Scale
- Object Identity
- Pose

- For a new image, let the matched codebook vote for possible object positions



Implicit Shape Model