PATTERN RECOGNITION AND IMAGE UNDERSTANDING

The ultimate objective of many image analysis tasks is to discover meaning of the analysed image, e.g. categorise the objects, provide symbolic/semantic interpretation of the image or global **understanding of the image scene**.

Because these tasks are application specific no ready solutions are available. Hence, image analysis problems have strong resemblance to ***intelligent cognition*** rather than rely on application of "ready to use" set of standard processing techniques.

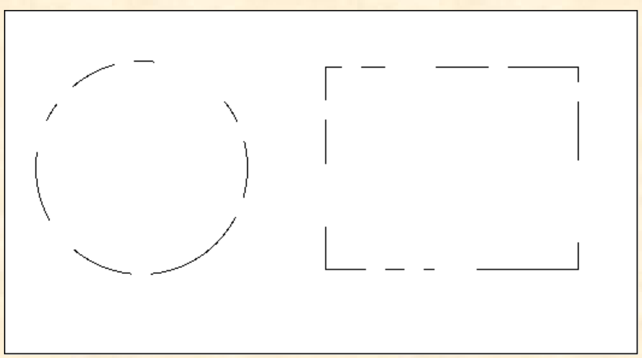
Image understanding

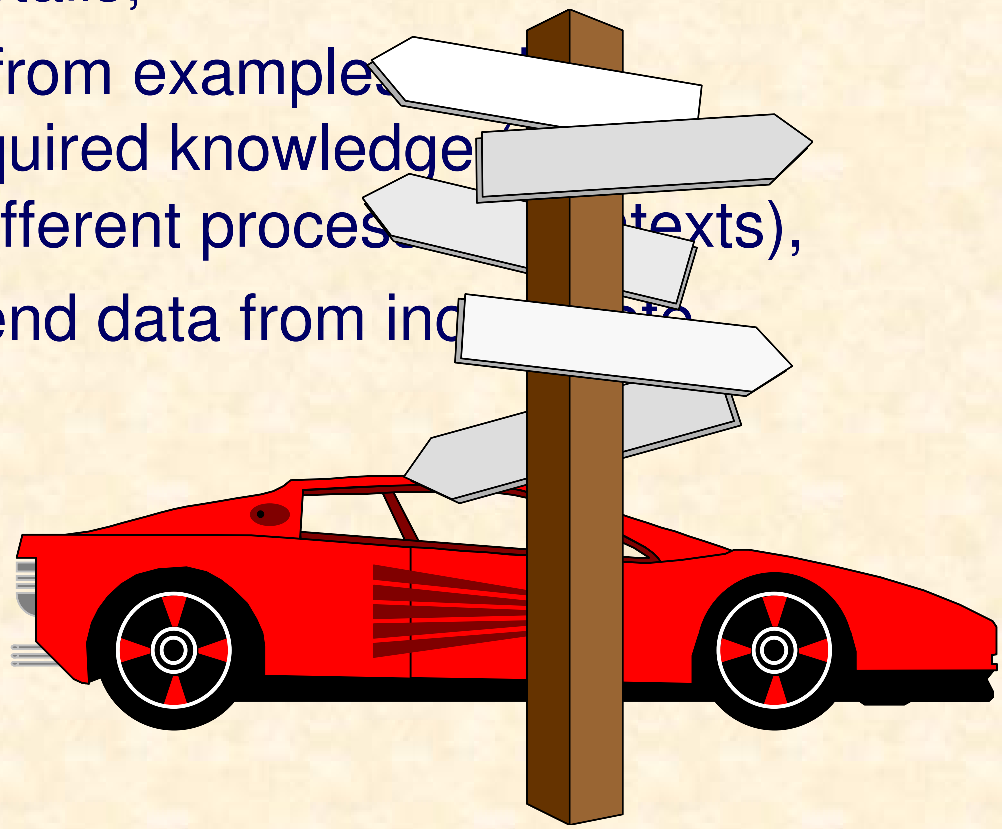
Intelligence:

⇒ ability to extract meaningful information from a set of irrelevant details,

⇒ capability to learn from example
generalise the acquired knowledge
can be useful in different proces texts),

⇒ ability to comprehend data from inc
information.

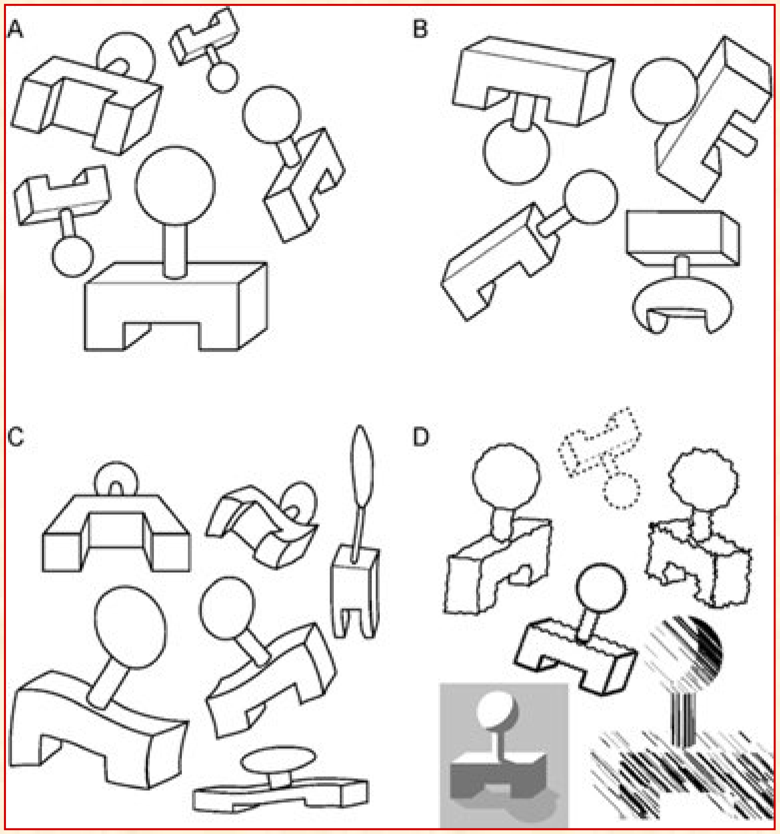




The concept of learning - example

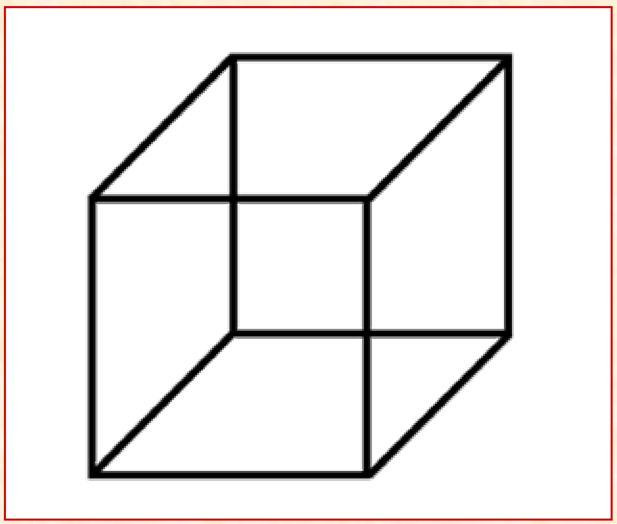
Emergence

An example of a difficult pattern recognition problem

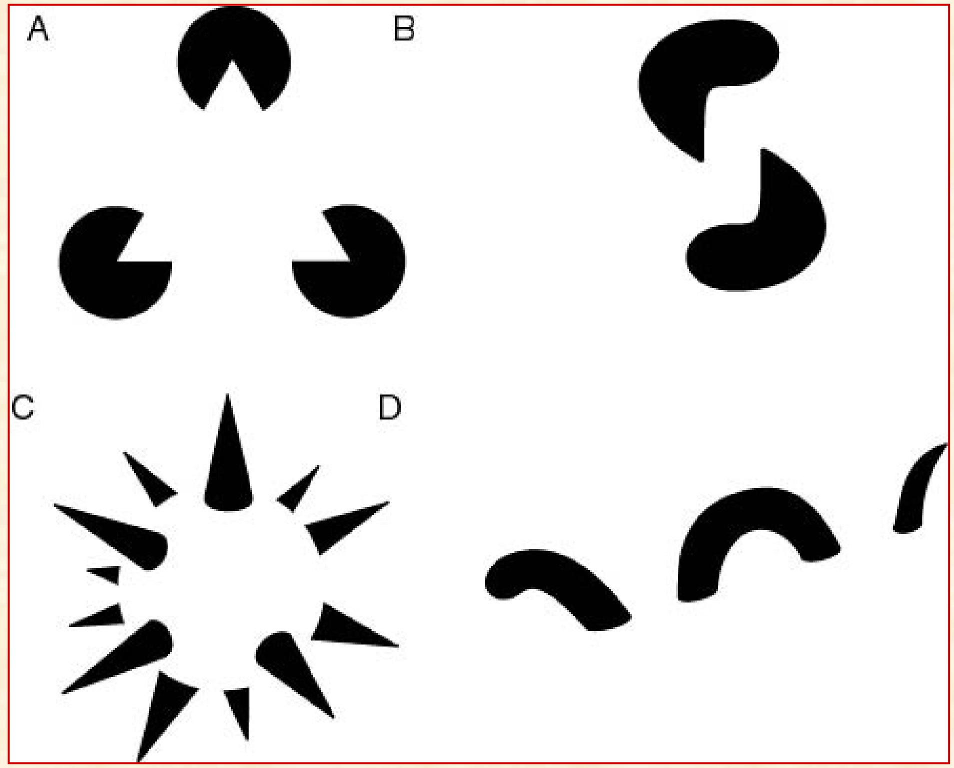


Invariance

Gestalt – Theory of mind and brain

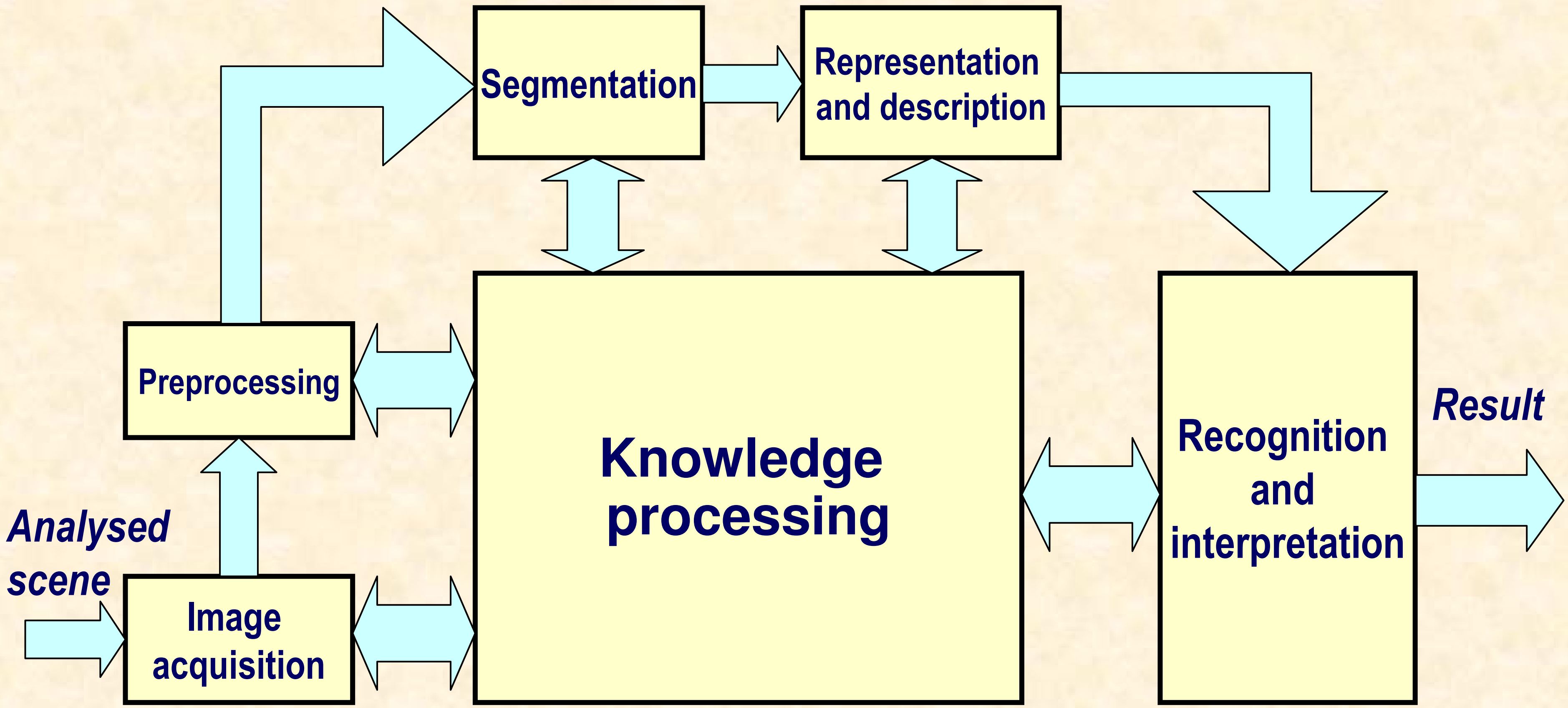


Multistability



Reification (illusory objects)

Main parts of an image analysis system



Main parts of an image analysis system

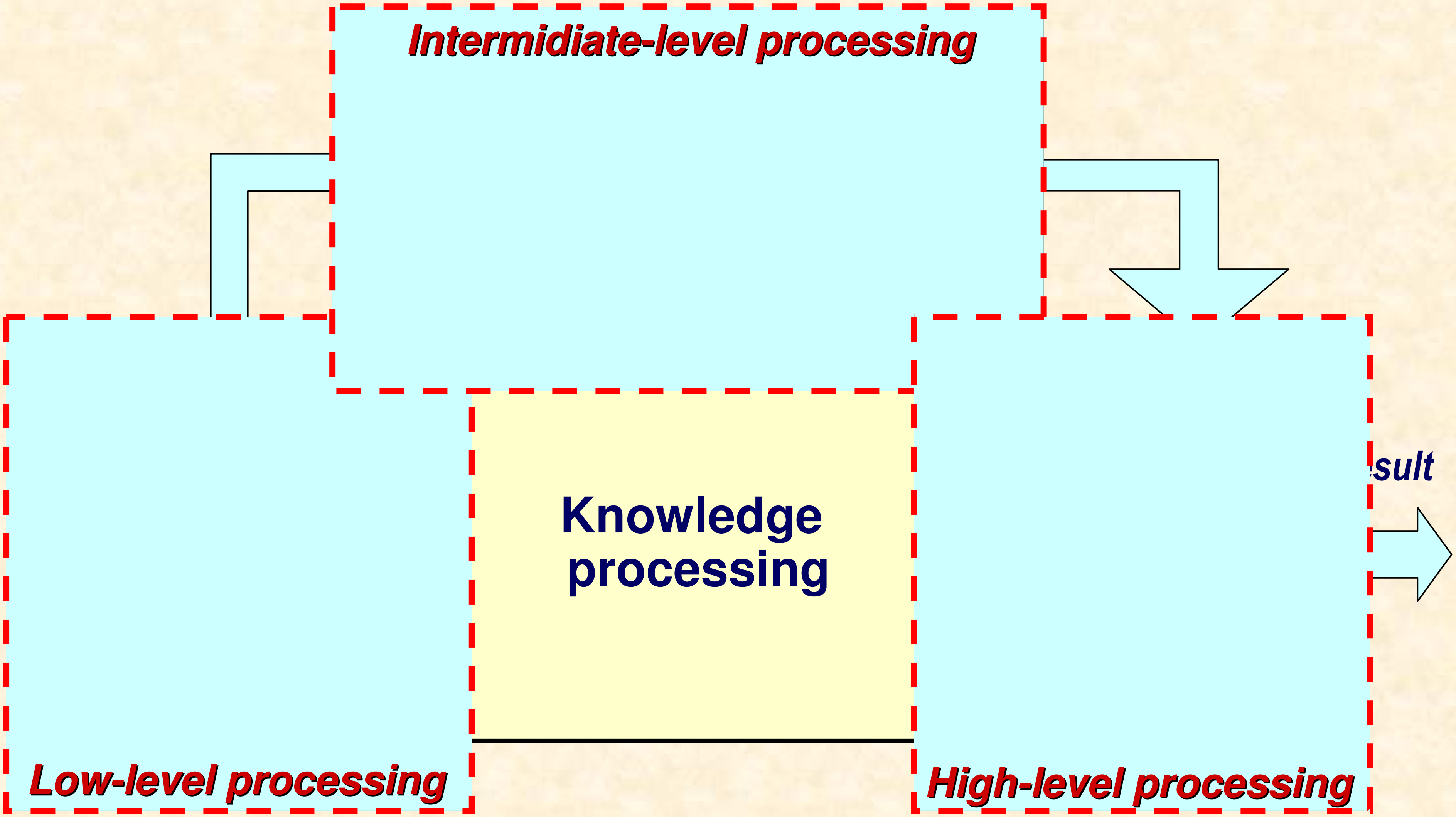


Image analysis system

A complete image analysis system comprises three major data processing modules:

Low-level processing- comprises image acquisition, image pre- processing and enhancement, e.g. noise reduction, contrast improvement, filtering, zooming, etc.

Intermediate level processing- deals with extraction and description of image components identified from a knowledge base, e.g. image segmentation, boundary and edge description, morphological processing.

High-level processing- involves classification, recognition and interpretation of the image scene.

Patterns and pattern classes

A **pattern** is a set of features that form a qualitative or structural description of an analysed object; in a mathematical sense a pattern is defined as a vector of features $\boldsymbol{x}=[x_1, x_2, \ldots, x_N]$.

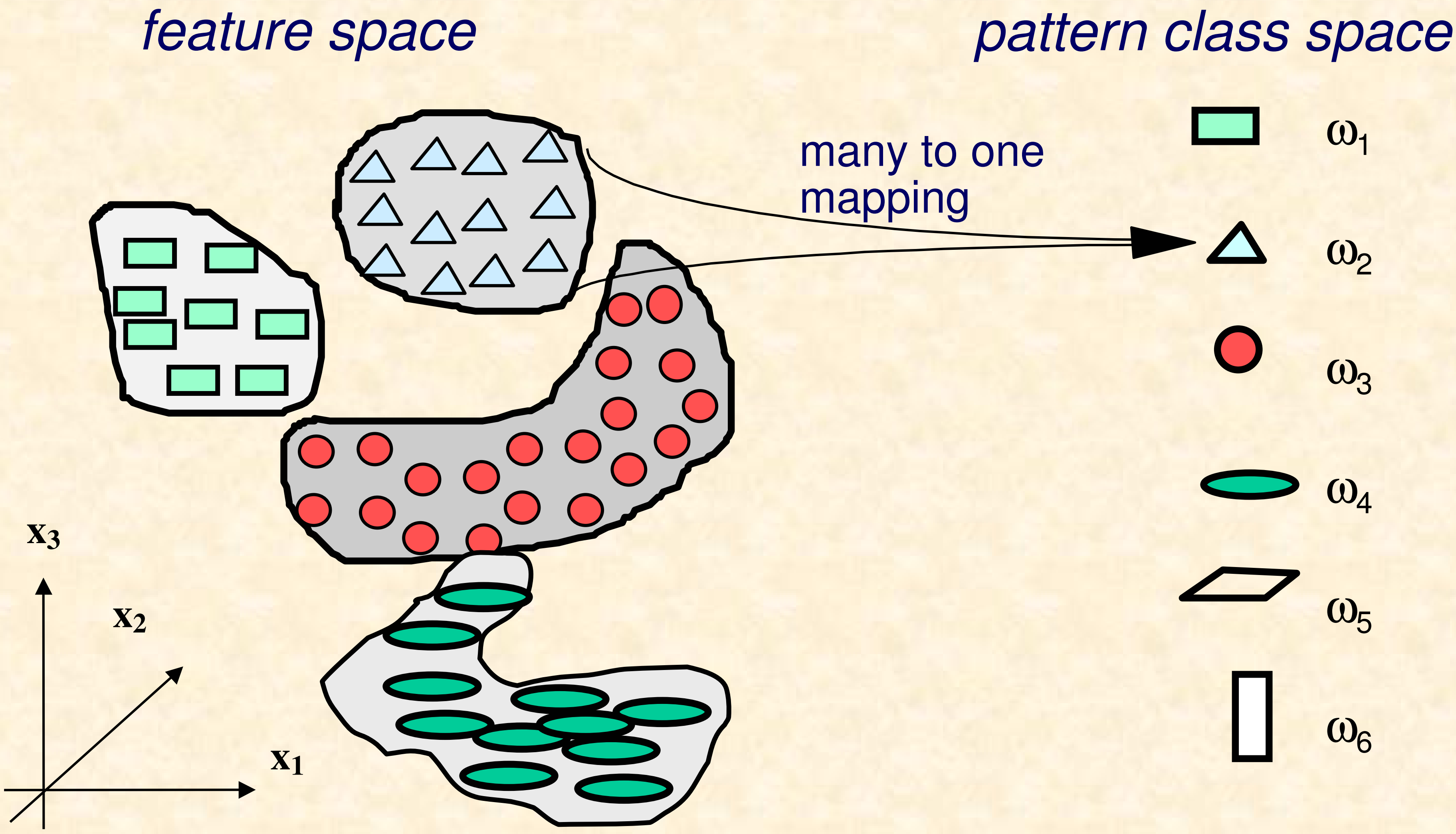
A **pattern class** is a group of patterns that have similar feature vectors (according to some similarity measure). Pattern classes are denoted $\omega_1, \omega_1, \ldots \omega_M$, where M is the number of classes.

Pattern recognition (alternatively termed pattern classification) is the task of assigning patterns to their respective classes. It is equivalent to establishing a mapping:

$$\mathbf{x} \rightarrow \omega$$

from the **feature space** X to the **pattern class space** Ω.

Pattern classification definition



Pattern classification problem

The mapping $\mathbf{x} \rightarrow \omega$ should point to the true class label ω.

The optimum among all the possible mappings $\mathbf{x} \rightarrow \omega$ is the one that generates minimum error rate.

This rate depends on the distribution characteristics of feature vectors in the pattern space X, i.e., their distributions, overlap between distributions, etc.

If the knowledge about distributions of the patterns is unavailable the classification system must rely on the learning paradigms, e.g. ***artificial neural networks***. In such approaches the knowledge is acquired from a sufficiently large number set of training samples drawn from the pattern examples

Selection of feature properties

- **discrimination** – features should assume considerably different values for patterns from different classes, e.g., diameter of fruits is a good feature for classification between grapefruits and cherries,
- **robustness** – features should assume similar values for all patterns belonging to the same class, e.g., colour is a poor feature for apples,
- **independence** – features used in a classification system should be uncorrelated, e.g., weight and size of a fruit are strongly correlated features,
- **small number of features –** complexity of data classification system grows substantially with the number of classified features, e.g., features that are strongly correlated should be eliminated.

Selection of feature properties

Correlation coefficient between features *X* and *Y*:

$$\gamma_{xy} = \frac{\frac{1}{P}\sum_{i=1}^{P}\left(x_i - \mu_x\right)\left(y_i - \mu_y\right)}{\sigma_x \sigma_y}$$

where: *P* – is the number of patterns, μ_x, μ_y are the average values for features *X* and *Y* correspondingly, and σ_x, σ_y are their standard deviations. The correlation coefficient assumes values in the range [1,1].

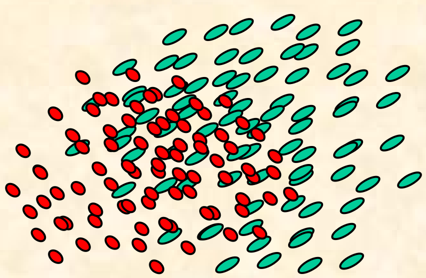
Selection of feature properties

A measure of **separation** for feature *X* between classes *j* and *k*:

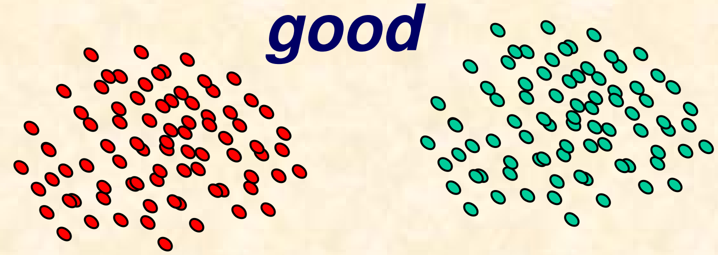
$$\hat{D}_{xjk} = \frac{\left| \mu_{xj} - \mu_{xk} \right|}{\sqrt{\sigma_{xj}^2 + \sigma_{xk}^2}}$$

Large value for this measure means feature X yields good separation between classes.

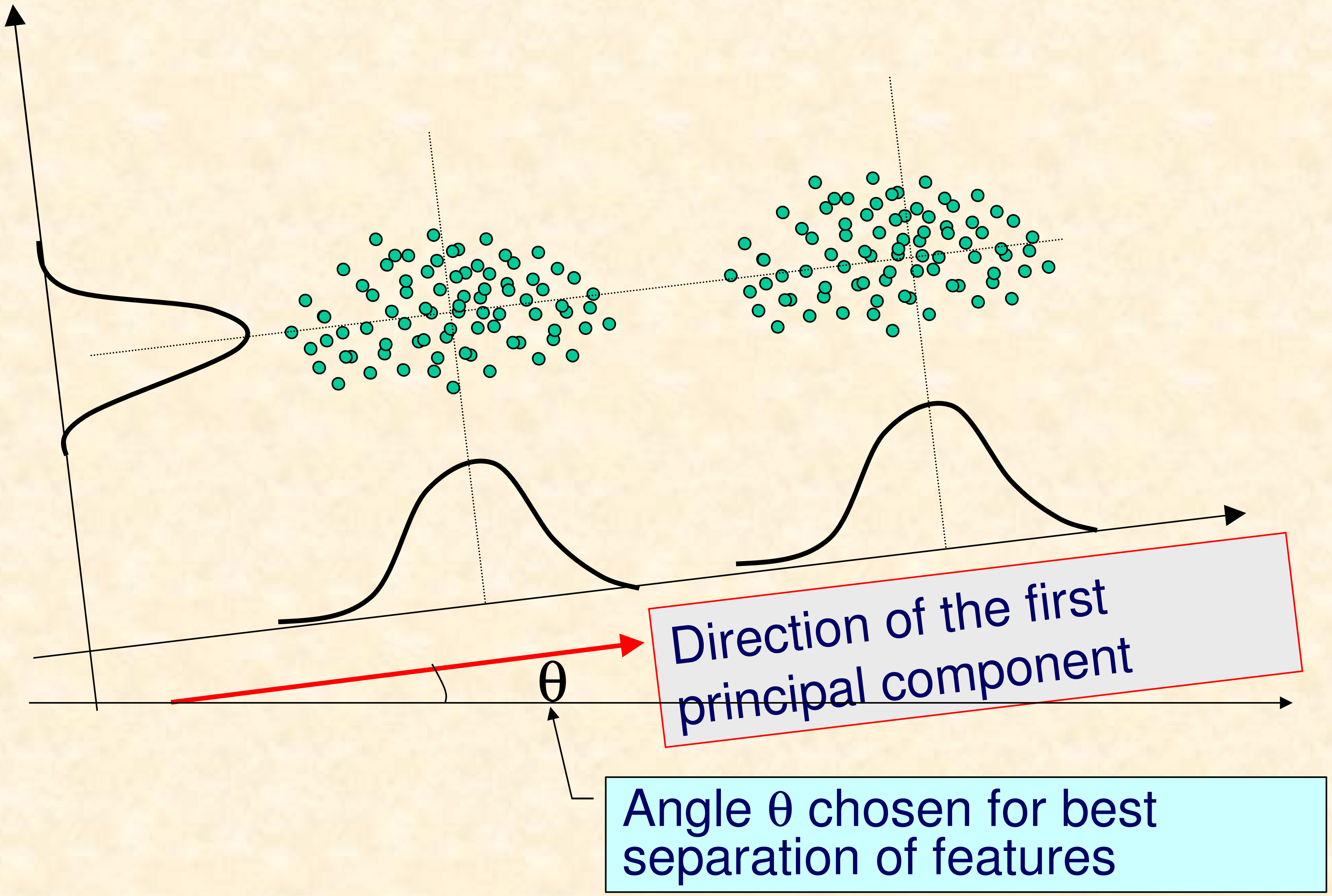
bad



good



Reduction of the number of features



Decision-theoretic methods

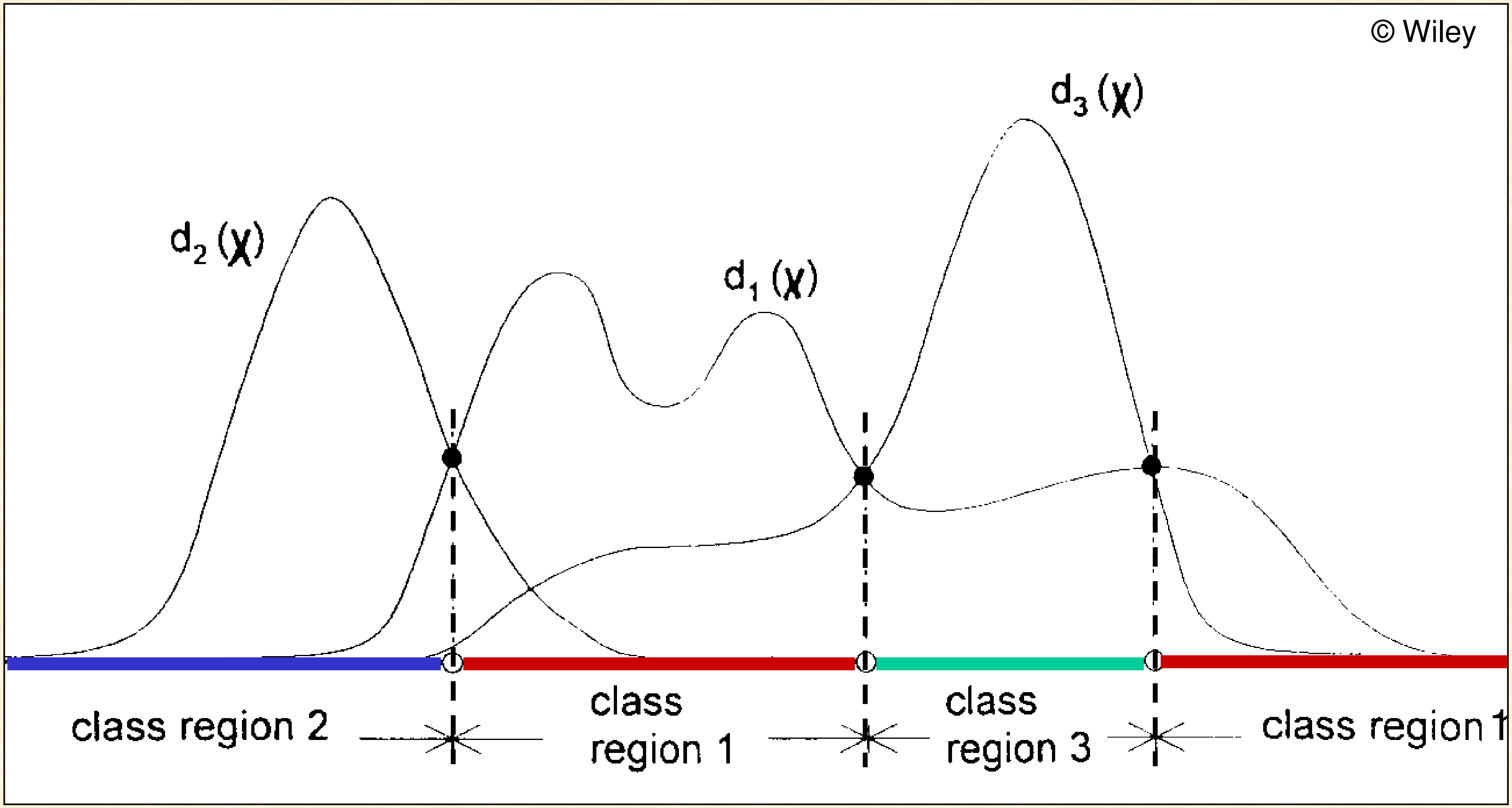
Pattern classification problem can be approached by means of the so called ***decision functions***. The number of these functions is equal to the number of classes.

Let $\mathbf{x}=[x_1, x_2, \ldots, x_N]^T$ represent an N-dimensional pattern vector. For M pattern classes $\omega_1, \omega_2, \ldots \omega_M$, we are searching for M decision functions $d_1(\boldsymbol{x}), d_2(\boldsymbol{x}), \ldots, d_M(\boldsymbol{x})$ with a property that, if a pattern $\boldsymbol{x}$ belongs to class ω_i then

$$d_i(\boldsymbol{x}) > d_j(\boldsymbol{x}) \quad j = 1, 2, \ldots, M; i \neq j$$

In other words, decision function $d_i(\boldsymbol{x})$, "wins the competition" for assigning the input feature vector $\boldsymbol{x}$ to the pattern class ω_i.

Decision functions



Decision functions

The decision boundary between two arbitrary classes i and j ($i \neq j$) is defined by the function:

$$d_{ij}(\boldsymbol{x}) = d_i(\boldsymbol{x}) - d_j(\boldsymbol{x}) = 0$$

Then, for patterns of class ω_i :

$$d_{ij}(\boldsymbol{x})>0$$

and for patterns of class ω_j:

$$d_{ij}(\boldsymbol{x})<0.$$

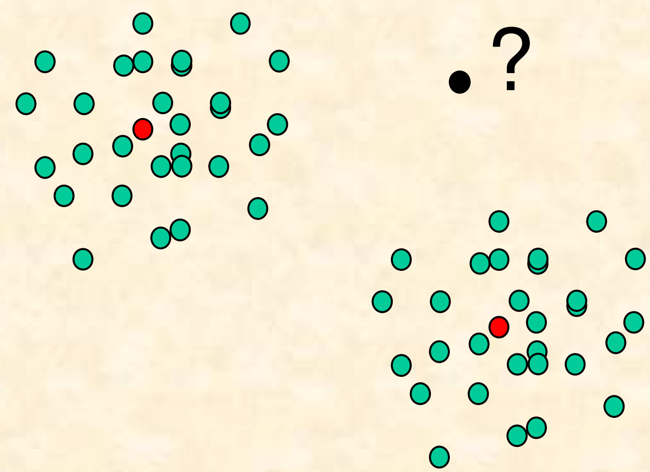
Minimum distance classifier

Assume that each pattern class is represented by a mean vector (also called a class *prototype*):

$$\boldsymbol{m}_i = \frac{1}{P_j} \sum_{x \in \omega_j} \boldsymbol{x}, \qquad\qquad j = 1,2,\ldots,M$$

where N_j is the number of pattern vectors from class ω_j.

Possible way to determine the class membership of an unknown pattern vector $\boldsymbol{x}$ is to assign it to the class of its closest prototype vector.



Minimum distance classifier

If the Euclidean distance is used the distance measure is of the form:

$$D_j(\boldsymbol{x}) = \left\| \boldsymbol{x} - \boldsymbol{m}_j \right\| \qquad j = 1,2,\ldots,M$$

and $\left\| \boldsymbol{x} \right\| = (\boldsymbol{x}^T \boldsymbol{x})^{1/2}$.

Feature vector ***x*** is assigned to class ω_j if $D_j(\boldsymbol{x})$ is the smallest distance.

Minimum distance classifier

The following distance function can be constructed

$$d_j(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{m}_j - \frac{1}{2}\boldsymbol{m}_j^T \boldsymbol{m}_j \qquad j = 1, 2, \ldots, M$$

and assigning $\boldsymbol{x}$ to class ω_j if $d_j(\boldsymbol{x})$ gives the largest value.

Minimum distance classifier

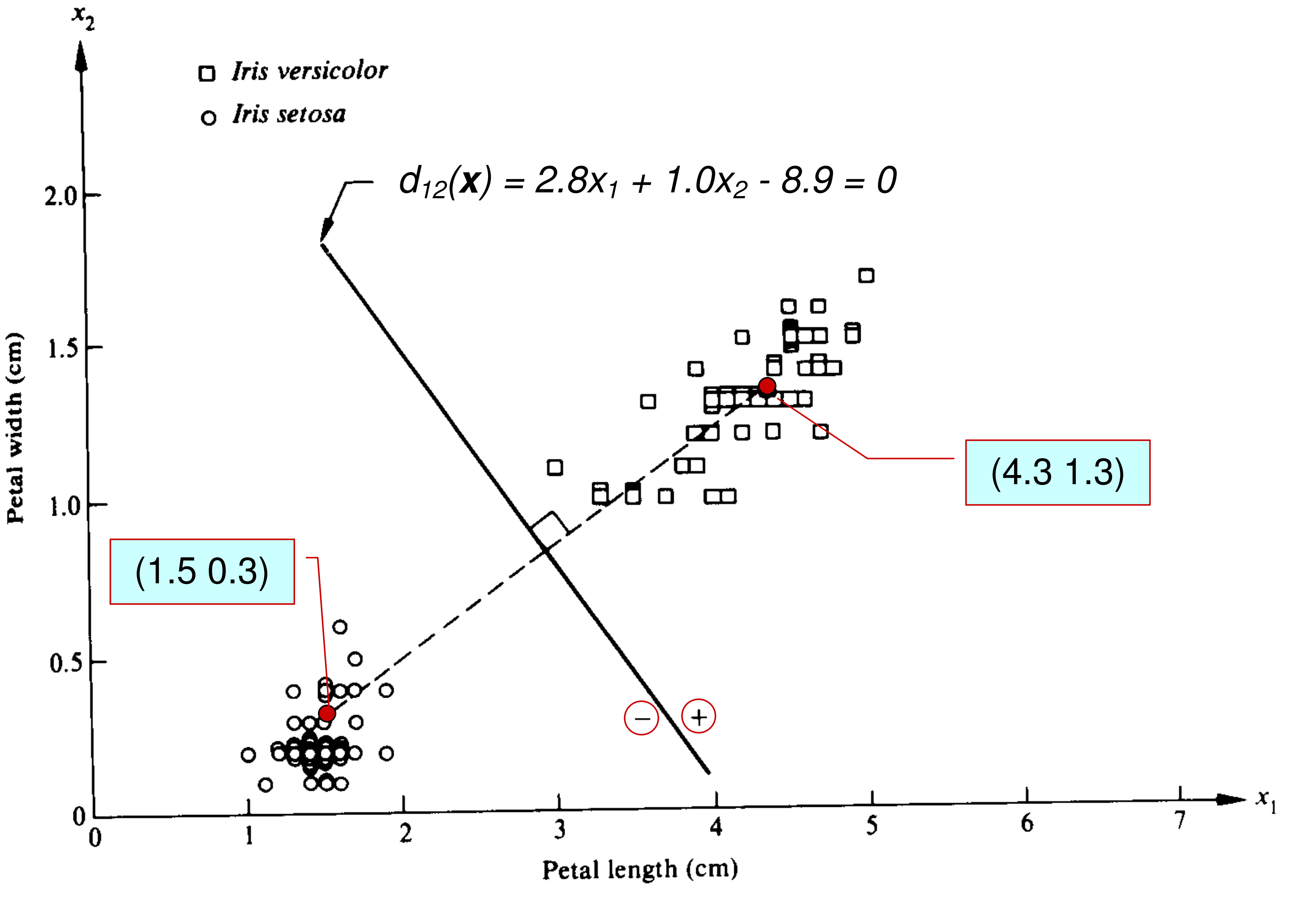
The decision boundary between classes ω_i and ω_j for a minimum distance classifier is:

$$d_{ij}(\boldsymbol{x}) = d_i(\boldsymbol{x}) - d_j(\boldsymbol{x}) =$$

$$= \boldsymbol{x}^T(\boldsymbol{m}_i - \boldsymbol{m}_j) - \frac{1}{2}(\boldsymbol{m}_i - \boldsymbol{m}_j)^T(\boldsymbol{m}_i - \boldsymbol{m}_j) = 0$$

The surface defined by this equation is the perpendicular bisector to the line joining $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$. For $N=2$ the bisector is a line, for $N=3$ it is a plane, and for $N>3$ it is called a *hyperplane*.

Minimum distance classifier



Minimum distance classifier

Example 1: Consider two classes denoted by ω_1 and ω_2, that have mean vectors $\mathbf{m}_1 = (4.3,\ 1.3)^T$ and $\mathbf{m}_2 = (1.5,\ 0.3)^T$. The decision functions for each of the classes are:

$$d_1(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_1 - 0.5\,\mathbf{m}_1^T\mathbf{m}_1 = 4.3x_1 + 1.3x_2 - 10.1$$

$$d_2(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_2 - 0.5\,\mathbf{m}_2^T\mathbf{m}_2 = 1.5x_1 + 0.3x_2 - 1.17$$

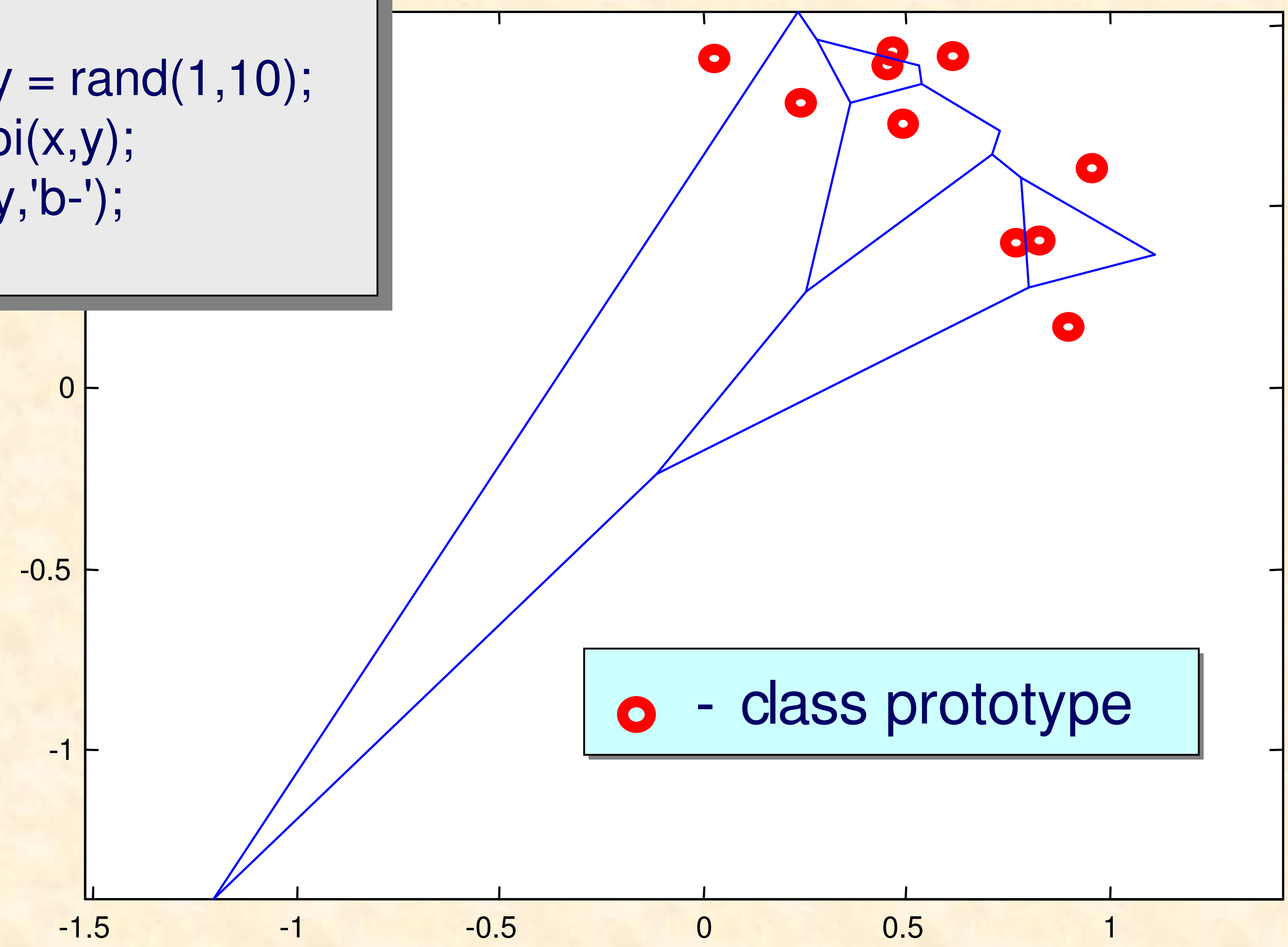
Equation for the decision boundary :

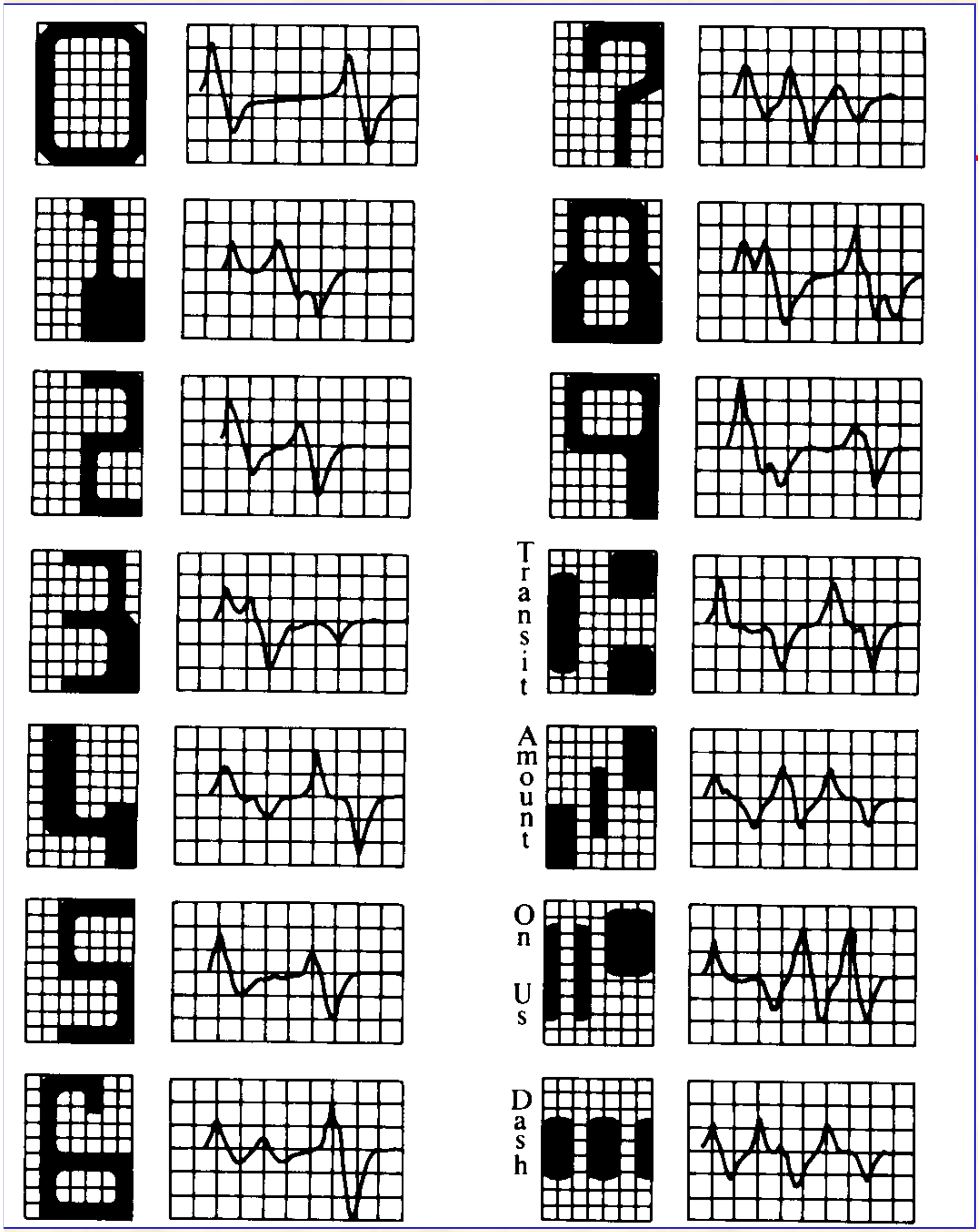
$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = 2.8x_1 + 1.0x_2 - 8.9 = 0$$

Class membership of a new feature vector is defined on the basis of the sign of $d_{12}(\mathbf{x})$.

Voronoi mosaic

```
%MATLAB
rand('state',0);
x = rand(1,10); y = rand(1,10);
[vx, vy] = voronoi(x,y);
plot(x,y,'r+',vx,vy,'b-');
axis equal;
```





Example 2:

Classification of American Bankers Association font character set.

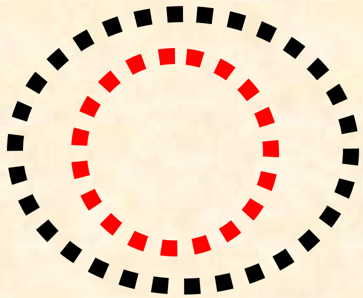
The design of the font ensures that the waveform corresponding to each character is distinct from that of all others.

There are N=14 prototype vectors in 10-D feature space.

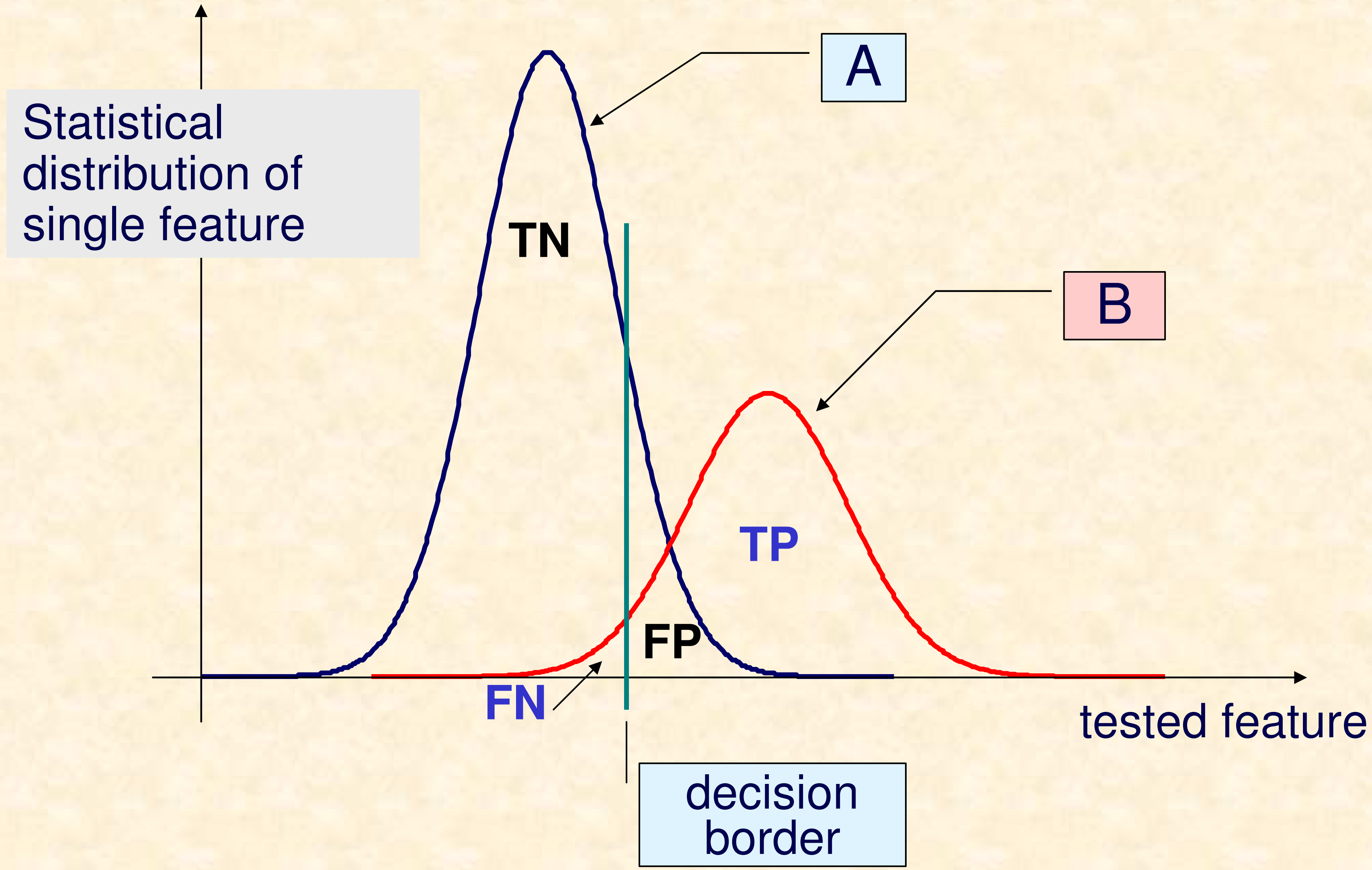
Minimum distance classifier

The minimum distance classifier works well on the following conditions:

- the distance between class means is large compared to the spread of each class.
- classes are linearly separable (class boundaries are: lines, planes or hyperplanes).



Evaluation of pattern classification quality



Evaluation of pattern classification quality

Confusion matrix

		True classification	
		Class B (wrong, ill, …)	Class A
Classifier output	Class B	**TP (**ang. *true-positive*)	**FP** (ang. *false-positive*)
	Class A	**FN** (ang. *false-negative*)	**TN** (ang. *true-negative*)

Evaluation of pattern classification quality

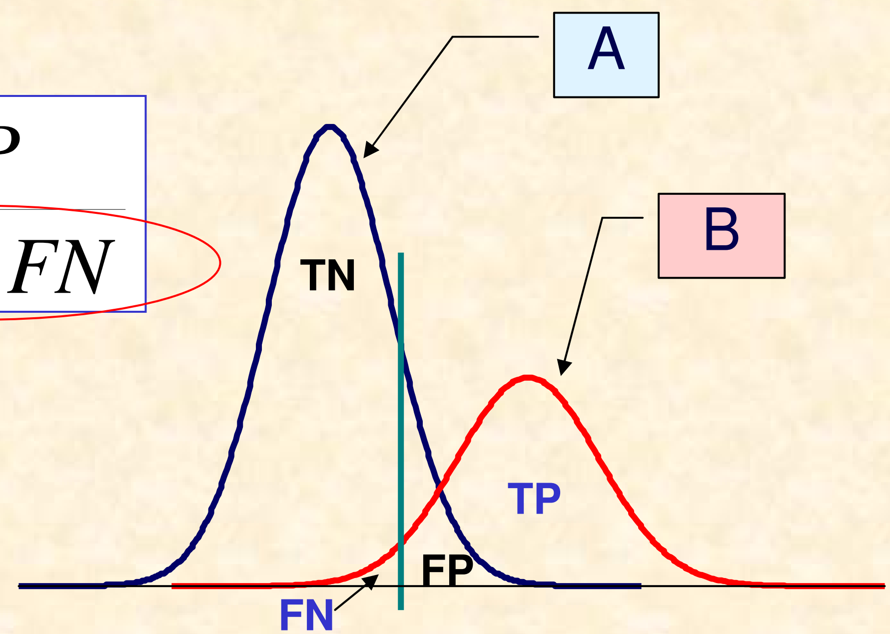
Accuracy:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Sensitivity:

$$SE = \frac{TP}{TP + FN}$$

All class B examples



Evaluation of pattern classification quality

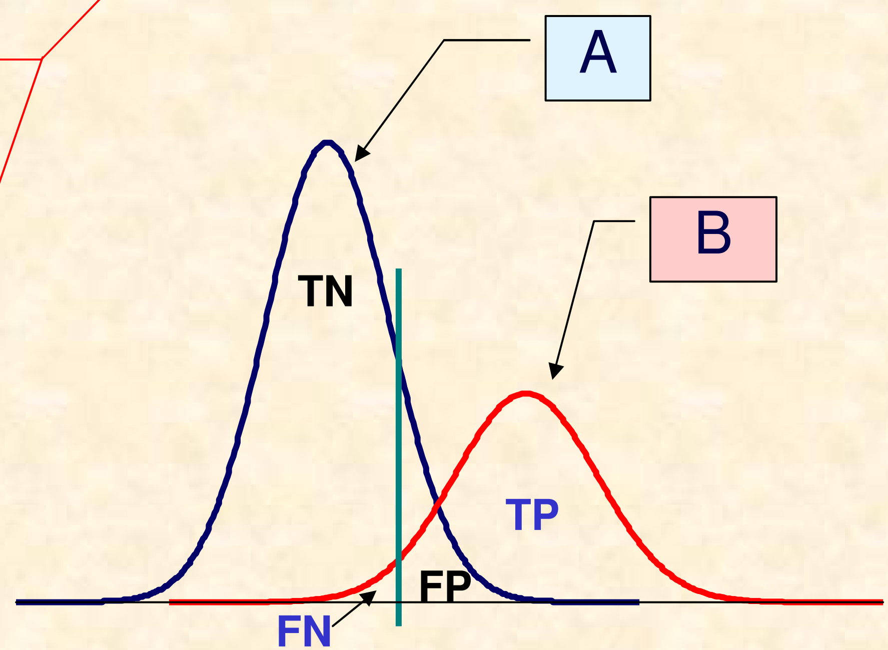
***Specifity*:**

$$SP = \frac{TN}{FP + TN}$$

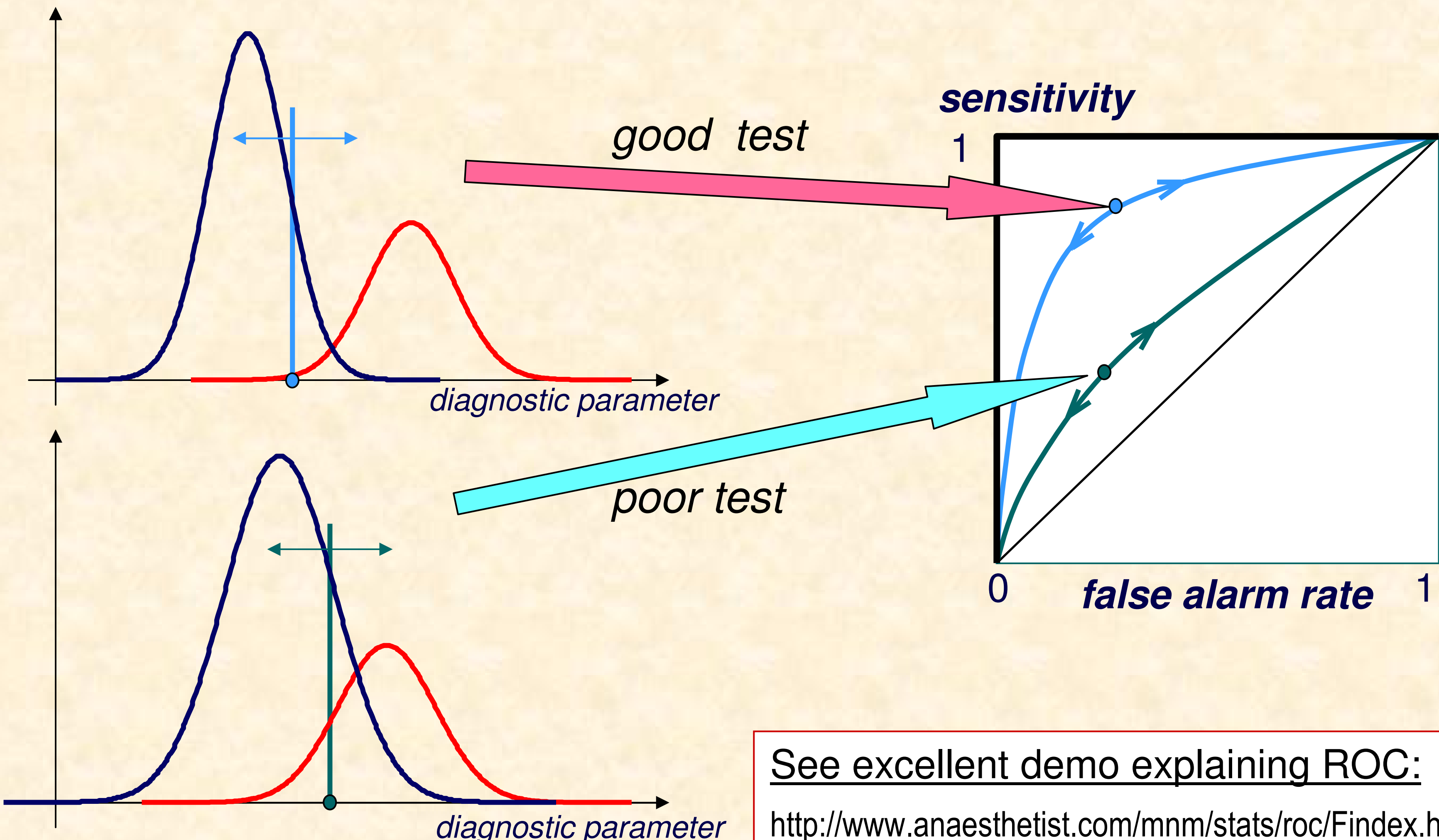
All class A examples

***False alarm rate*:**

$$FA = 1 - SP = \frac{FP}{FP + TN}$$



Receiver Operator Characteristic (ROC curve)



See excellent demo explaining ROC:

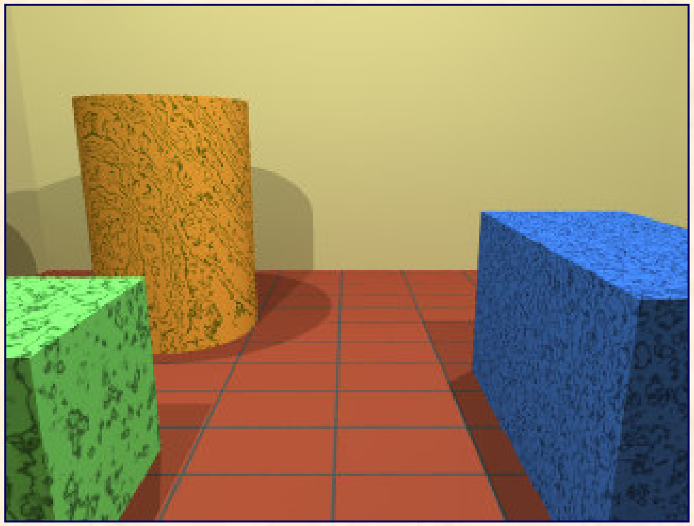
http://www.anaesthetist.com/mnm/stats/roc/Findex.htm

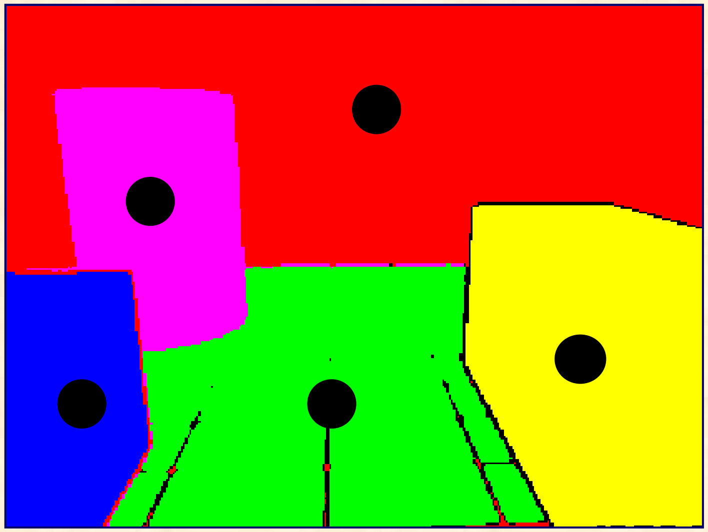
Types of classifiers

Parametric: probability density functions are known and only parameters of these functions are unknown e.g., the Gaussian distribution is defined by its mean μ and standard deviation σ), The Bayes classifier is a parametric classifier, i.e., requires knowledge about density function distributions

Nonparametric: probability density functions are not known and must be estimated from a sufficiently large measurement data.

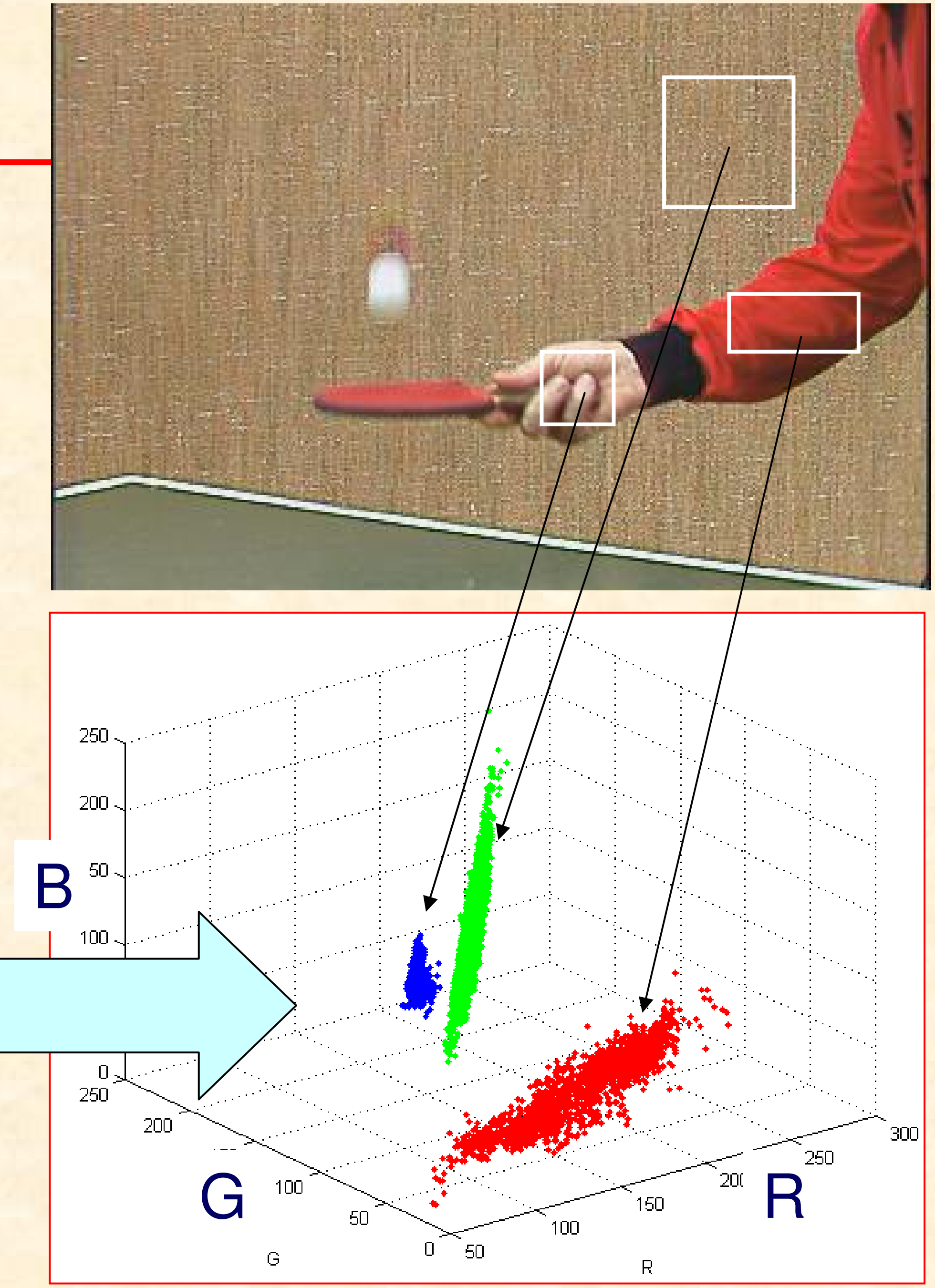
Image segmentation example



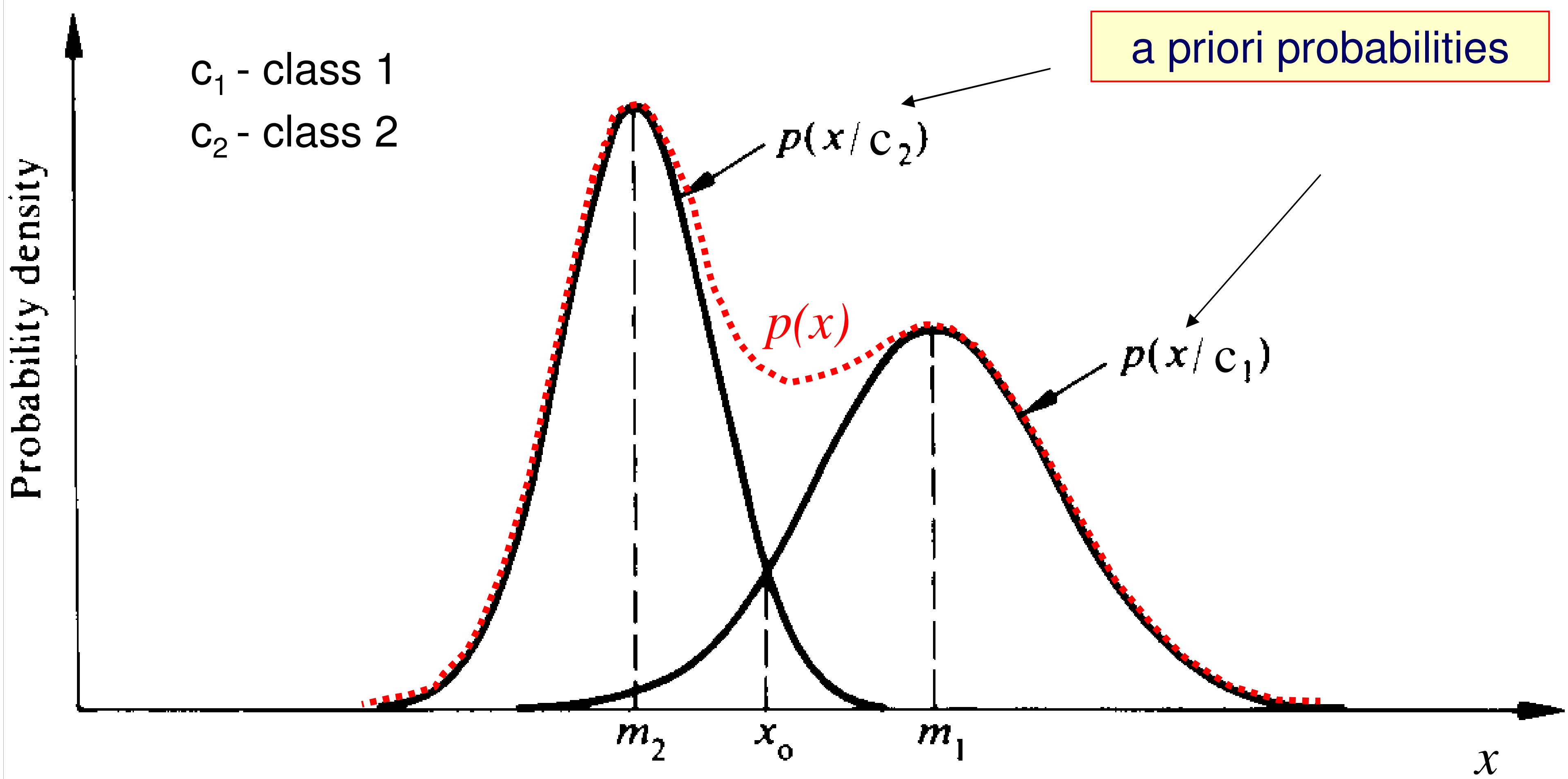


Colour based segmentation {R,G,B}

Statistical distribution of colour components for the indicated image regions



The Bayes classifier



The Bayes classifier

From the probability theory the following holds:

$$p(a/b) = \frac{p(a)\,p(b/a)}{p(b)}$$

hence:

a priori probability

$$p(c_i/\boldsymbol{x}) = \frac{P(c_i)\,p(\boldsymbol{x}/c_i)}{p(\boldsymbol{x})}$$

Bayes rule

a posteriori probability

Maximum likelihood estimator

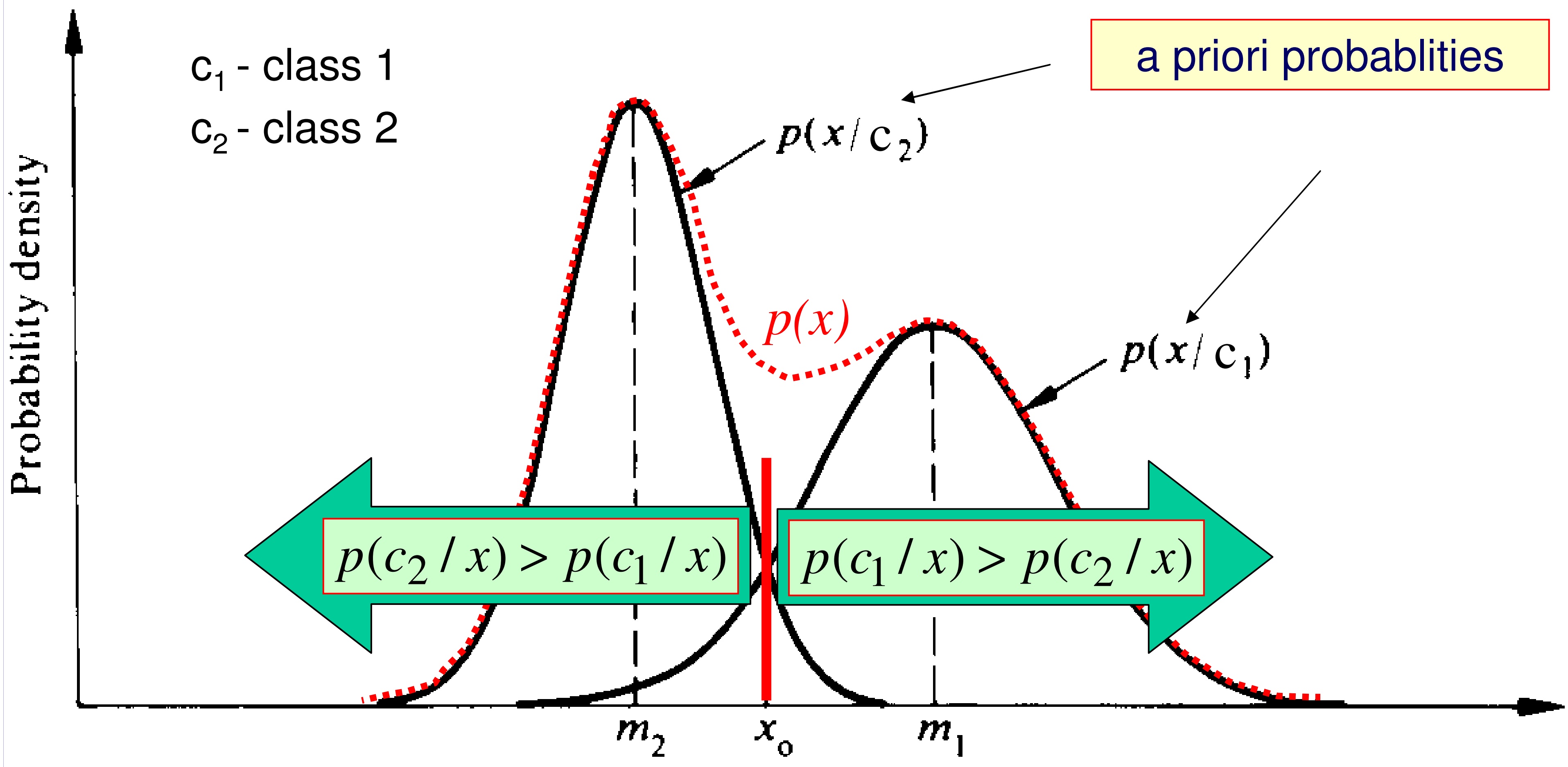
An image pixel is assigned to i th class for which:

$$L_i = \arg\max_i \left\{ p(c_j / \boldsymbol{x}) = \frac{P(c_i)\, p(\boldsymbol{x} / c_i)}{p(\boldsymbol{x})} \right\},\ i = 1,2,\ldots,N$$

i.e.

$$L_i > L_j,\ \ j \neq i,\ i = 1,2,\ldots,N$$

The Bayes classifier



The Bayes classifier

A priori probability for 1D Gaussian distribution:

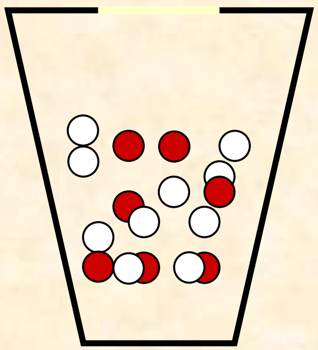
$$p(x / c_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x - m_i)^2}{2\sigma_i^2} \right] \quad i = 1,2$$

A posteriori probability:

$$p(c_i / x) = p(x / c_i) P(c_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x - m_i)^2}{2\sigma_i^2} \right] P(c_i) \qquad i = 1,2$$

The Bayes classifier - example

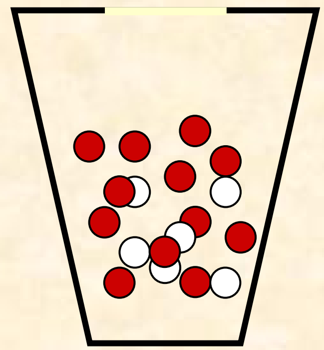
Pot A



20 – red

20 – white

Pot B



30 – red

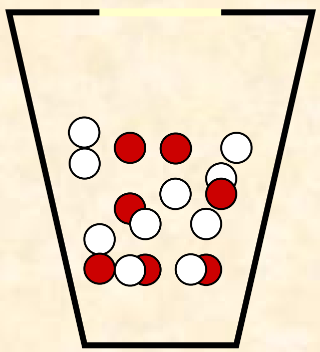
10 – white

We pick a pot randomly and then the ball randomly.

Question: From which pot the ball was picked?

The Bayes classifier - example

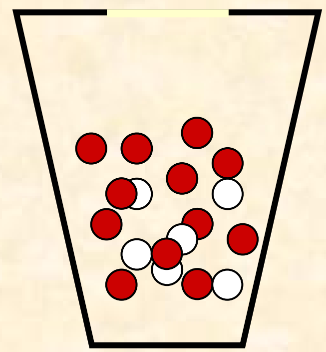
Pot A



20 – red

20 – white

Pot B



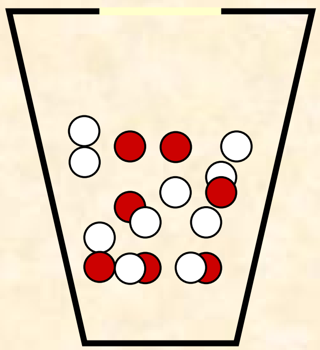
30 – red

10 – white

Before we see the colour of the ball we picked (a priori knowledge) we assume the probability of choosing each of the pots is equal, i.e. 0.5.

The Bayes classifier - example

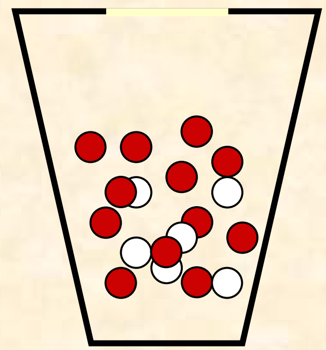
Pot A



20 – red

20 – white

Pot B



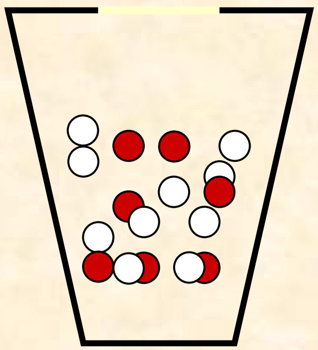
30 – red

10 – white

Suppose we picked a red ball. Can we verify our first hypothesis having this a posteriori knowledge?
Yes, the answer comes from the Bayes theorem.

The Bayes classifier -example

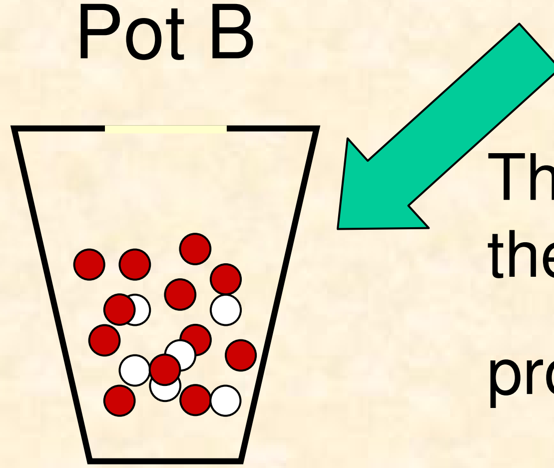
Pot A



20 – red

20 – white

Pot B



30 – red

10 – white

That was the pot!

probably

$$p(A/\mathbf{red}) = \frac{P(A)p(\mathbf{red}/A)}{P(A)p(\mathbf{red}/A) + P(B)p(\mathbf{red}/B)} = \frac{0.5 \times 0.5}{0.5 \times 0.5 + 0.5 \times 0.75} = 0.4$$

$$p(B/\mathbf{red}) = \frac{P(B)p(\mathbf{red}/B)}{P(A)p(\mathbf{red}/A) + P(B)p(\mathbf{red}/B)} = \frac{0.5 \times 0.75}{0.5 \times 0.5 + 0.5 \times 0.75} = 0.6$$

Multivariate Gaussian distribution

A priori probability for 2D Gassian distribution $\boldsymbol{x}^T=[x_1\ x_2]$:

$$p(\boldsymbol{x}/c_i)=\frac{1}{(2\pi)^{1/2}\left|\sum_i\right|^{1/2}}\exp\left[-\frac{(\boldsymbol{x}-\boldsymbol{m}_i)^T\sum_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)}{2}\right] \qquad i=1,2,\ldots,N$$

Where i is class index of N given classes

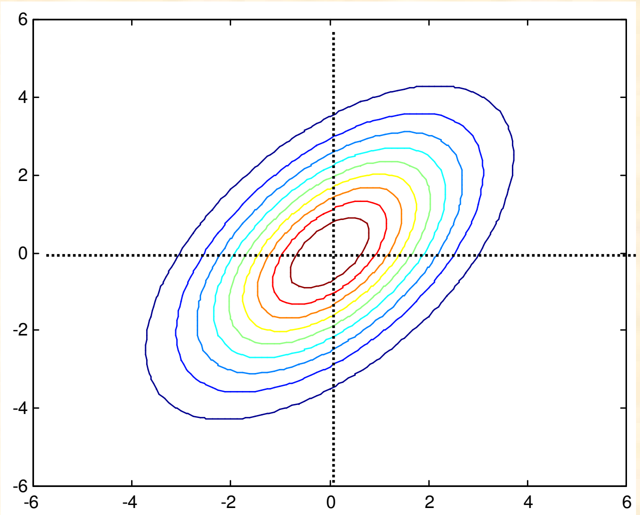
and the covariance matrix:

$$\Sigma_i=\begin{bmatrix}\sigma_{x1}^2 & \sigma_{x1}\sigma_{x2}\\ \sigma_{x2}\sigma_{x1} & \sigma_{x2}^2\end{bmatrix}_i$$

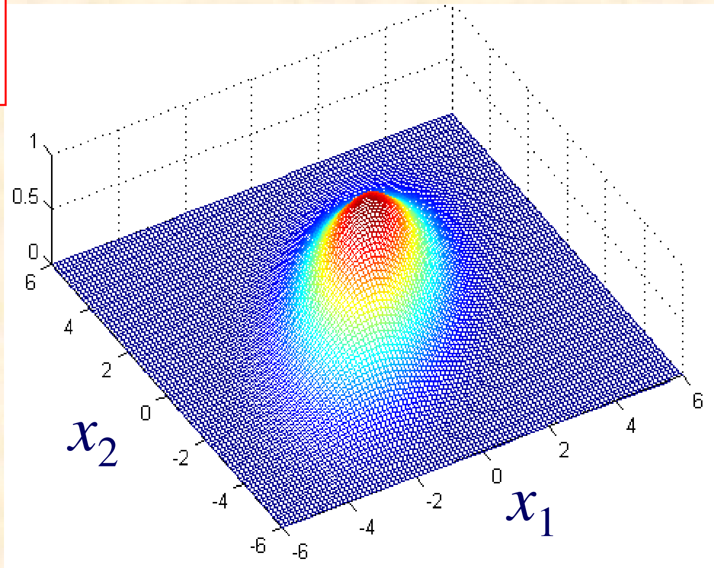
2D Gaussian distribution - example

Let $m_1 = m_2 = 0$:

$$\Sigma = \begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix} \longrightarrow \Sigma^{-1} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{8} \end{bmatrix} \longrightarrow |\Sigma| = 8$$



$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{1/2} \left|\sum\right|^{1/2}} \exp\left[-\frac{1}{2}\left(\boldsymbol{x}^T \sum\nolimits^{-1} \boldsymbol{x}\right)\right] = A\exp(B)$$

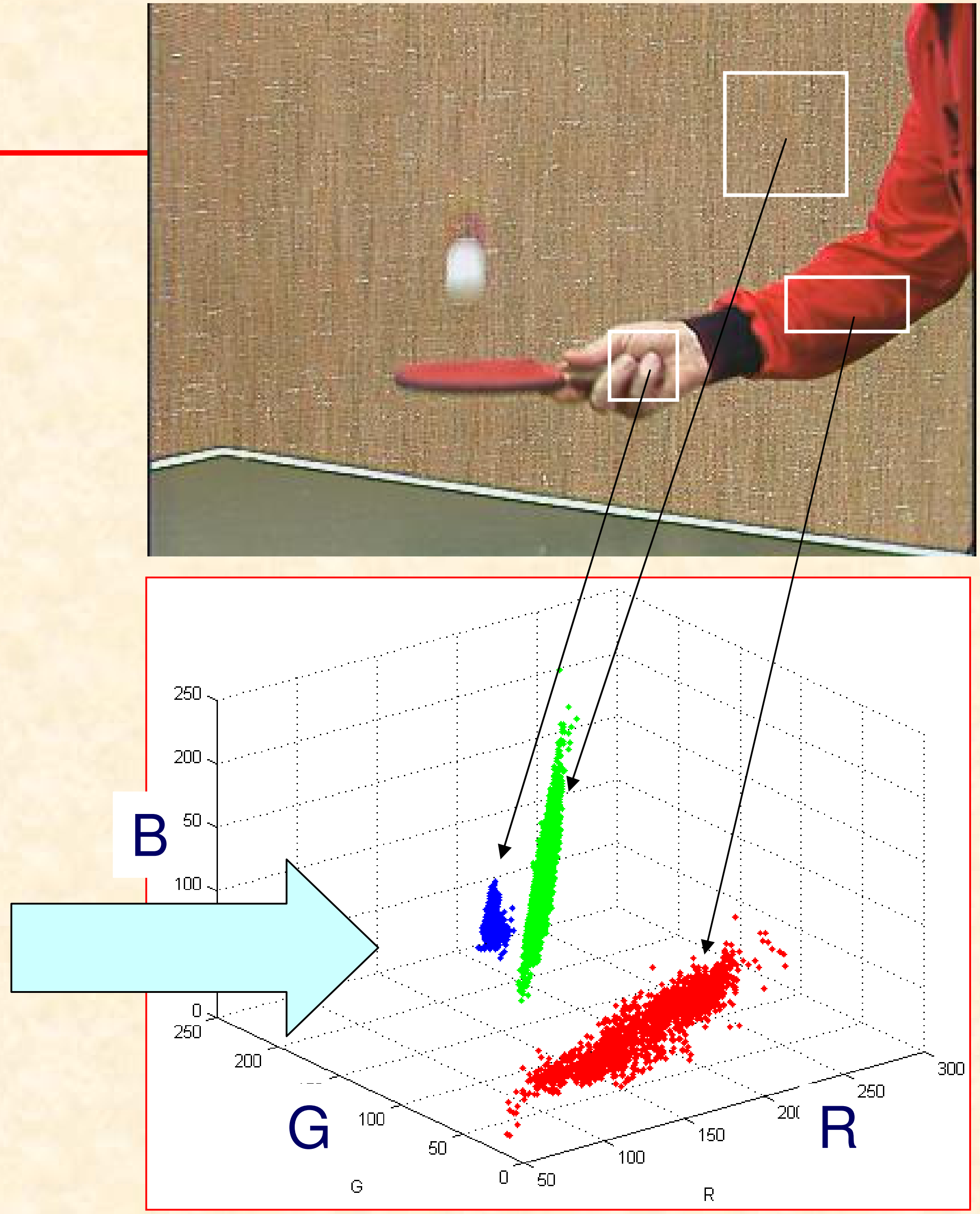


$$A = \frac{1}{(2\pi)^{1/2} 8^{1/2}} = \frac{1}{4\sqrt{\pi}}$$

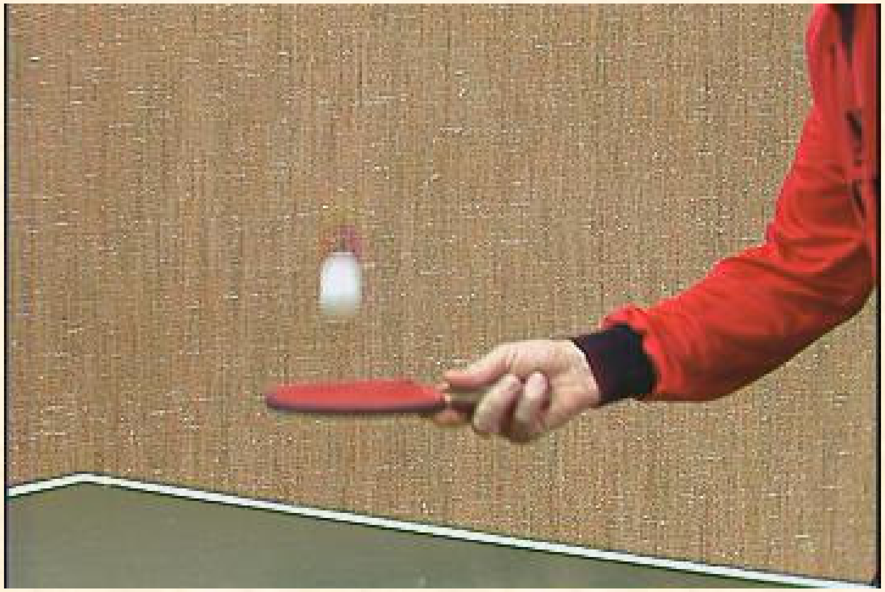
$$B = -\frac{1}{2}\left(\frac{1}{2}x_1^2 - \frac{1}{2}x_1 x_2 + \frac{3}{8}x_2^2\right)$$

Colour based segmentation {R,G,B}

Multivariate Gaussian distribution



Selection of the feature vector



Feature vector:

$$\theta = \{x, y, R, G, B, u, v\}$$

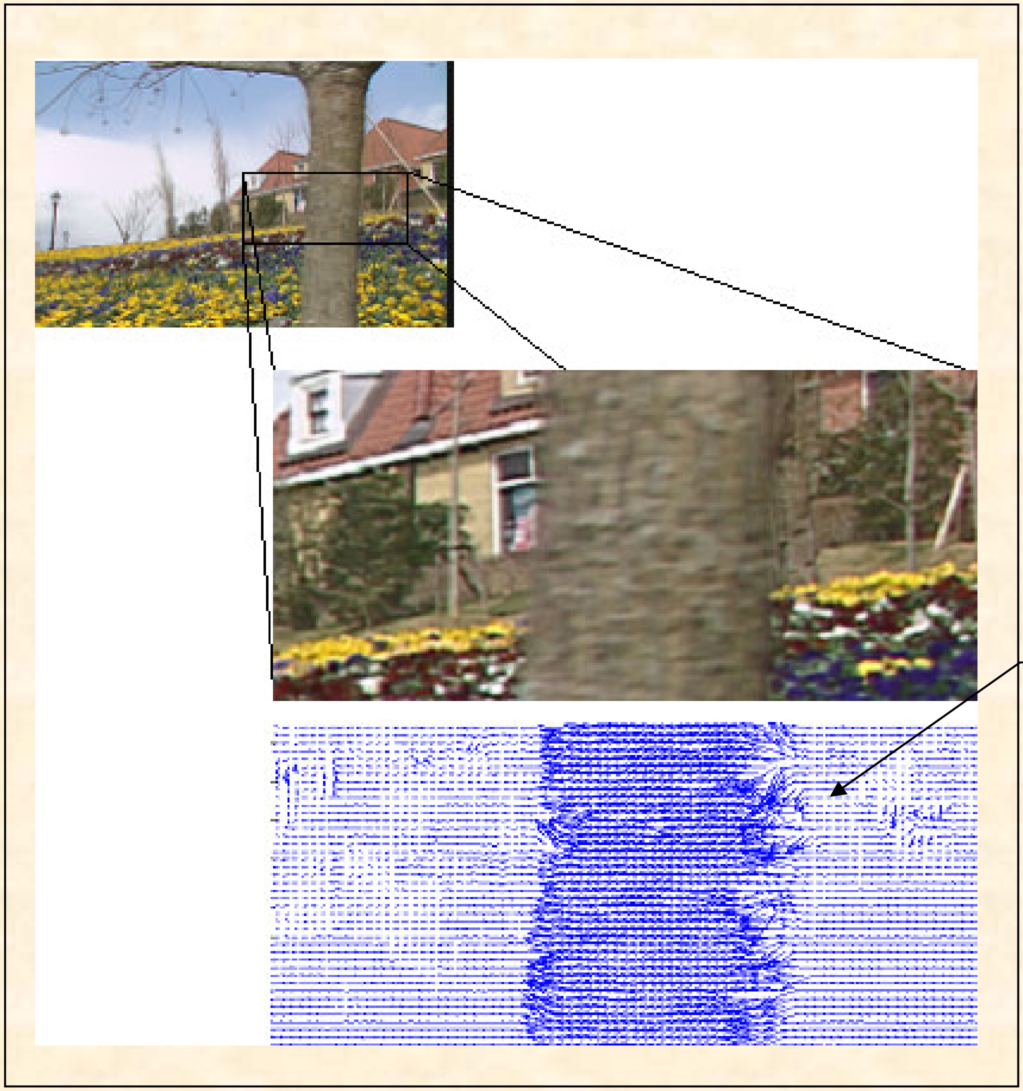
$$u = \frac{dx}{dt}$$
$$v = \frac{dy}{dt}$$

coordinates (x, y)

colour (R, G, B)

optical flow (u, v)

Optical flow features

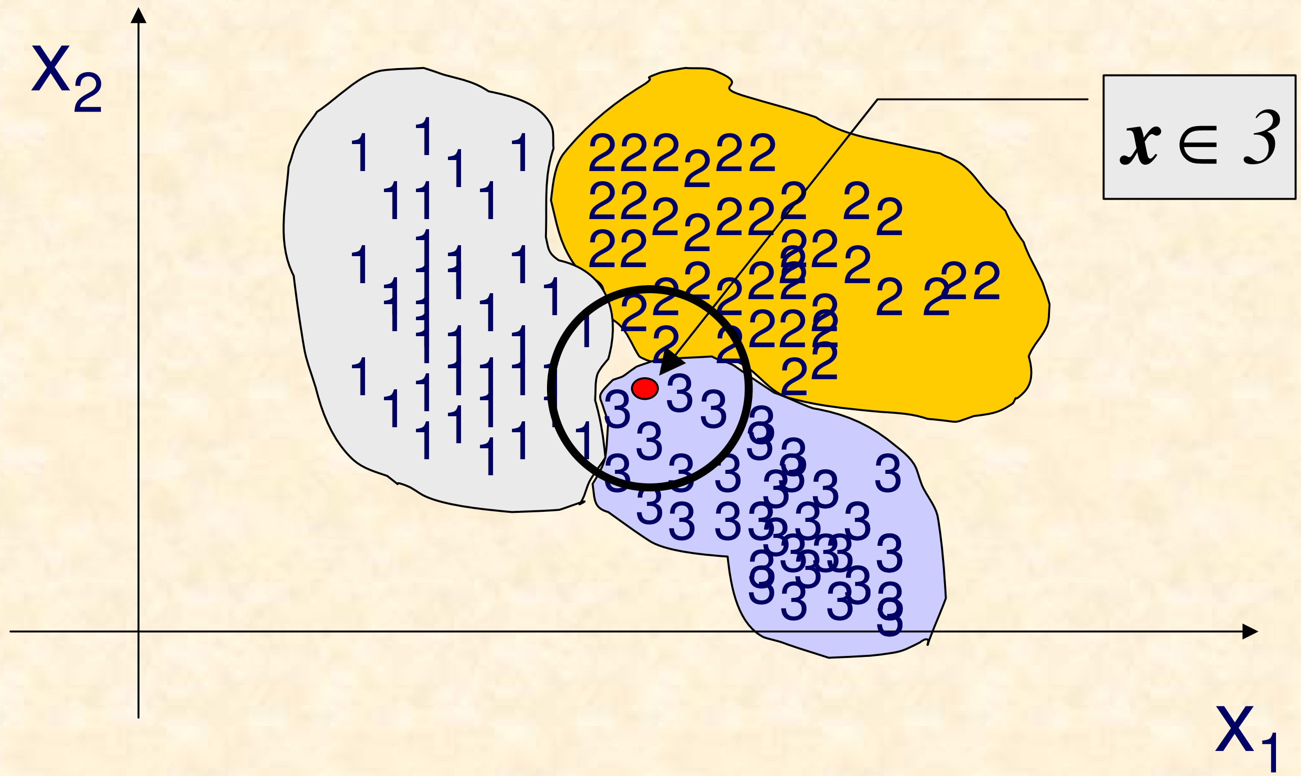


Optical flow based segmentation

$\{u, v\}$

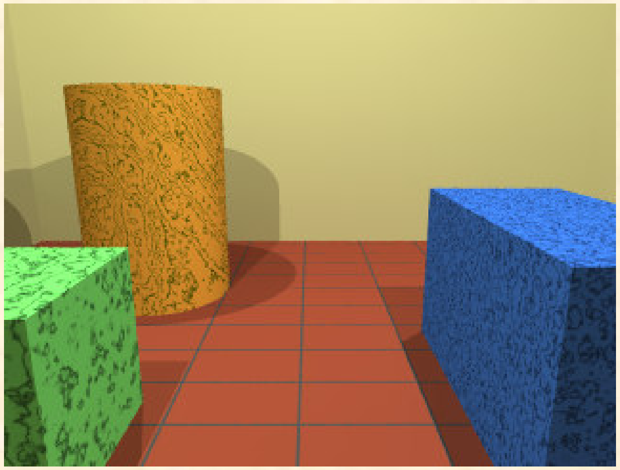
optical flow vector field

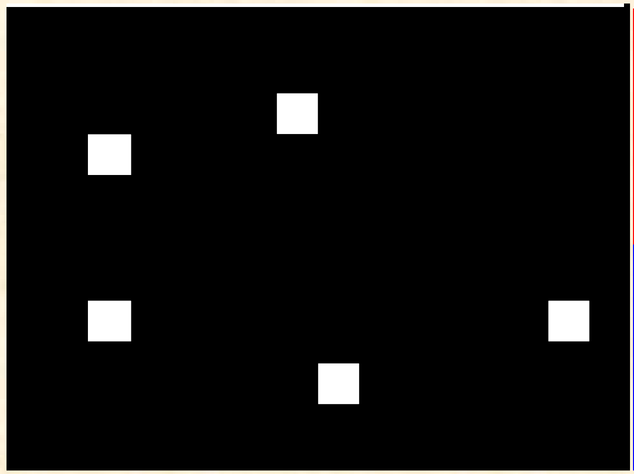
k-Nearest Neighbours Classifier (k-NN)

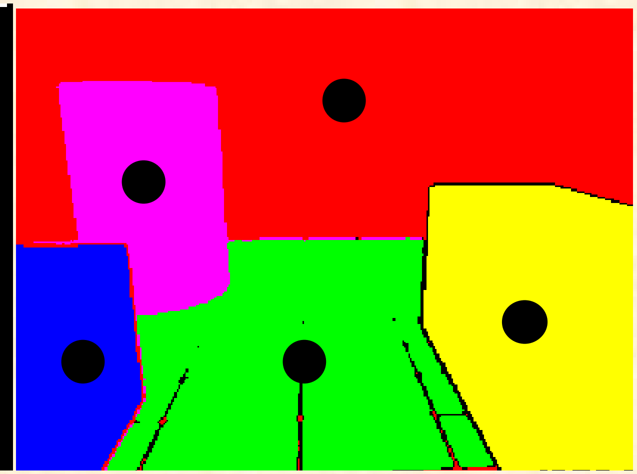


In each point of the feature space, a number of k neighbours is counted. The point is assigned to the class from which there is the largest number of points in the neighbourhood.

Image segmentation examples







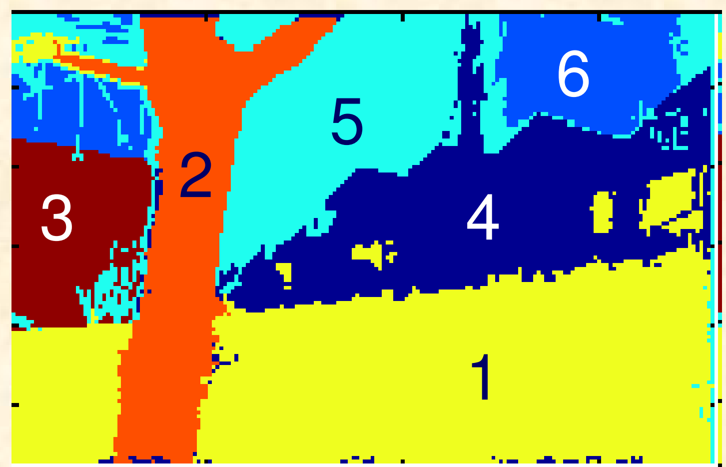
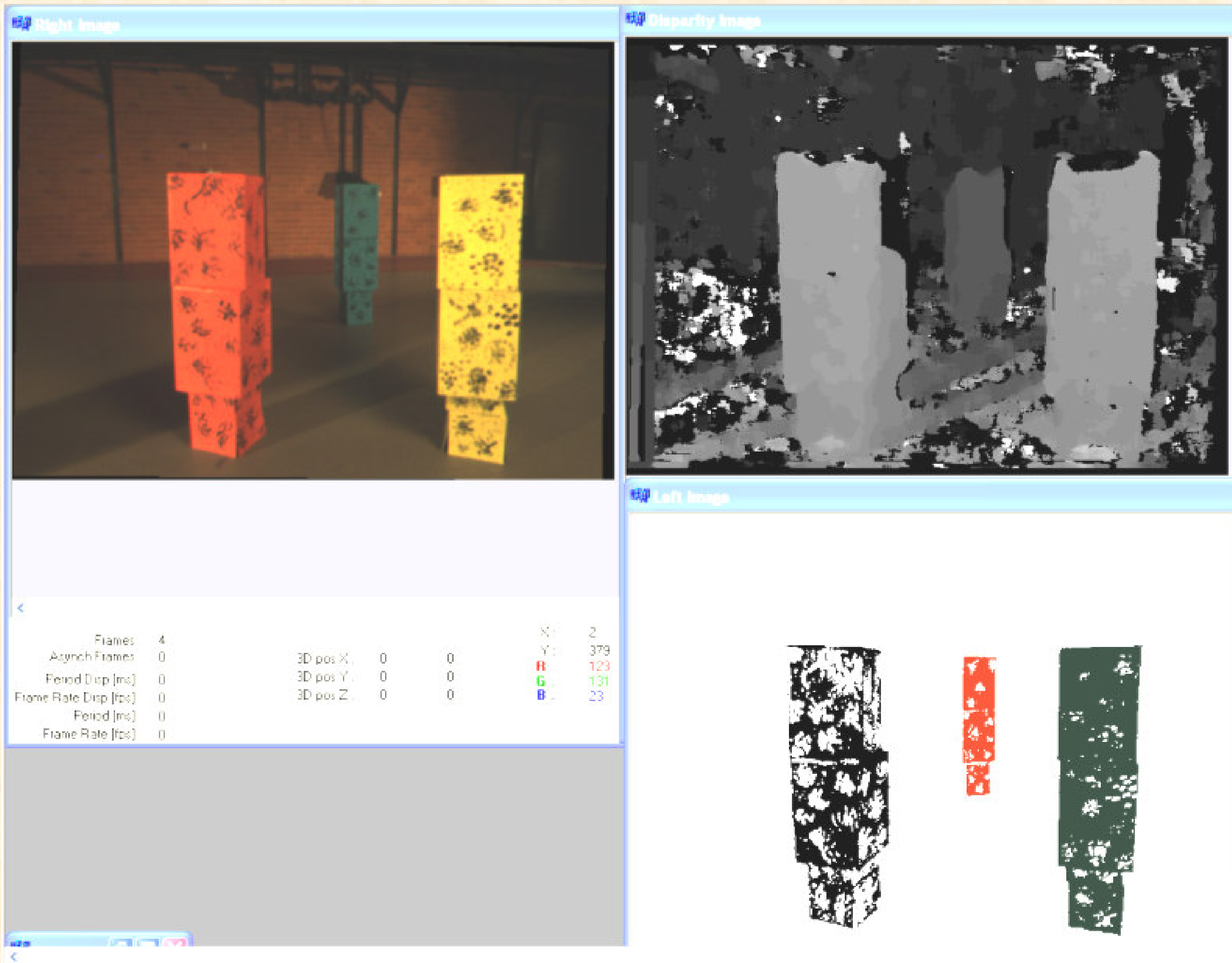
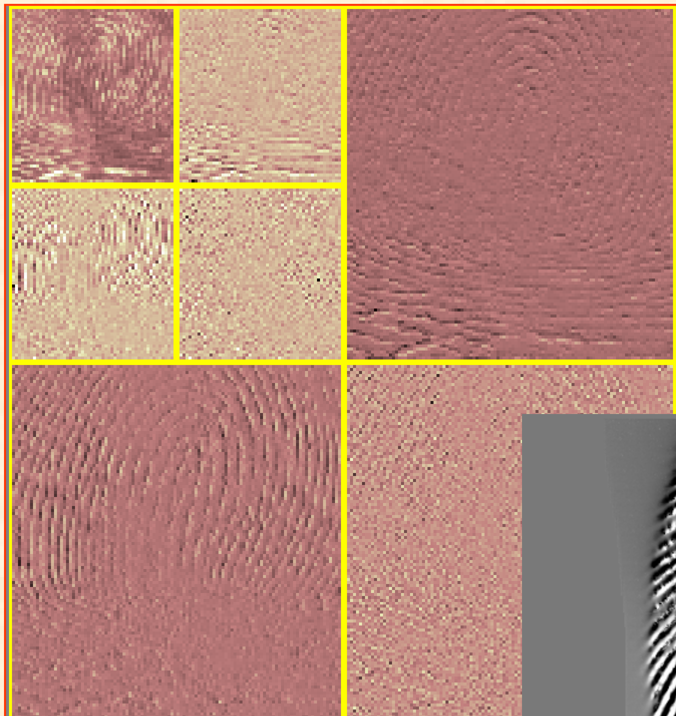


Image segmentation examples

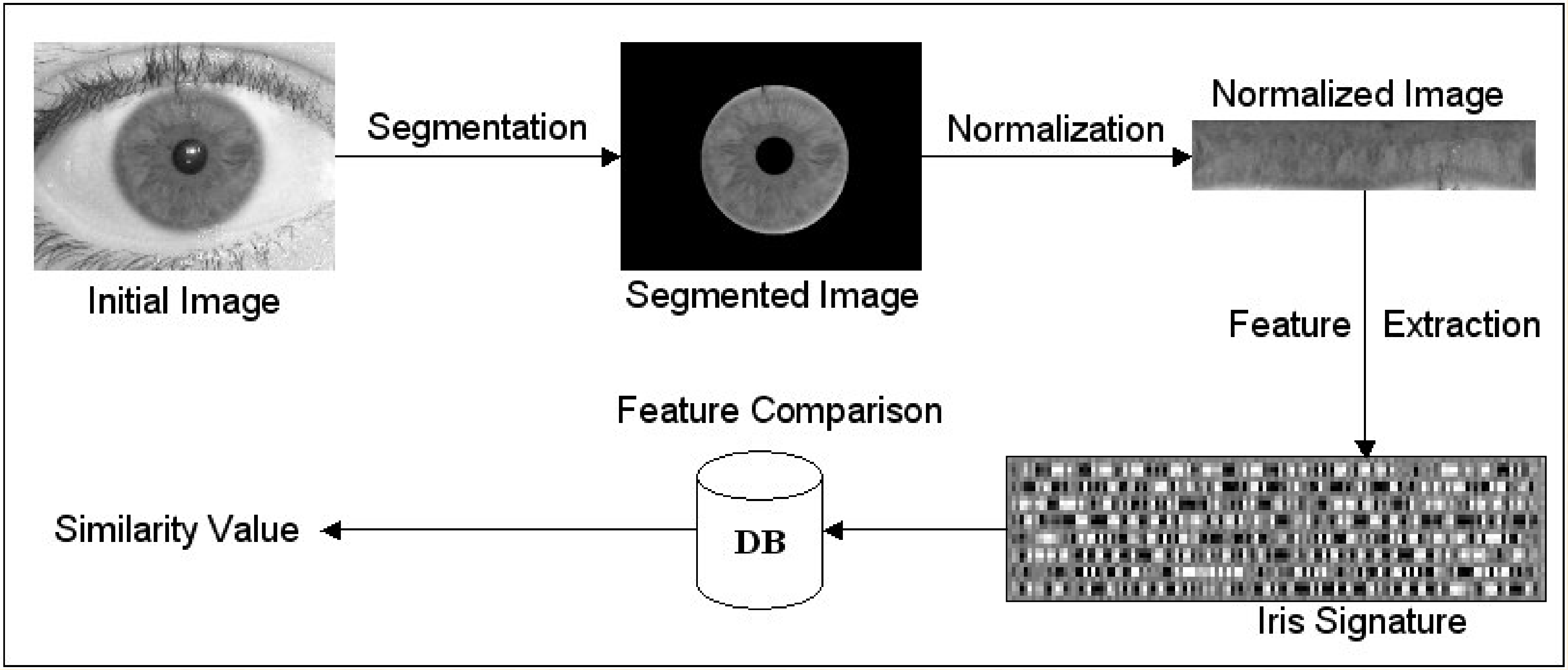


Fingerprint recognition



FBI image database of fingerprints (1992)

Iris recognition (biometry)



The patented Daugman's algorithm

Analysis of biomedical images

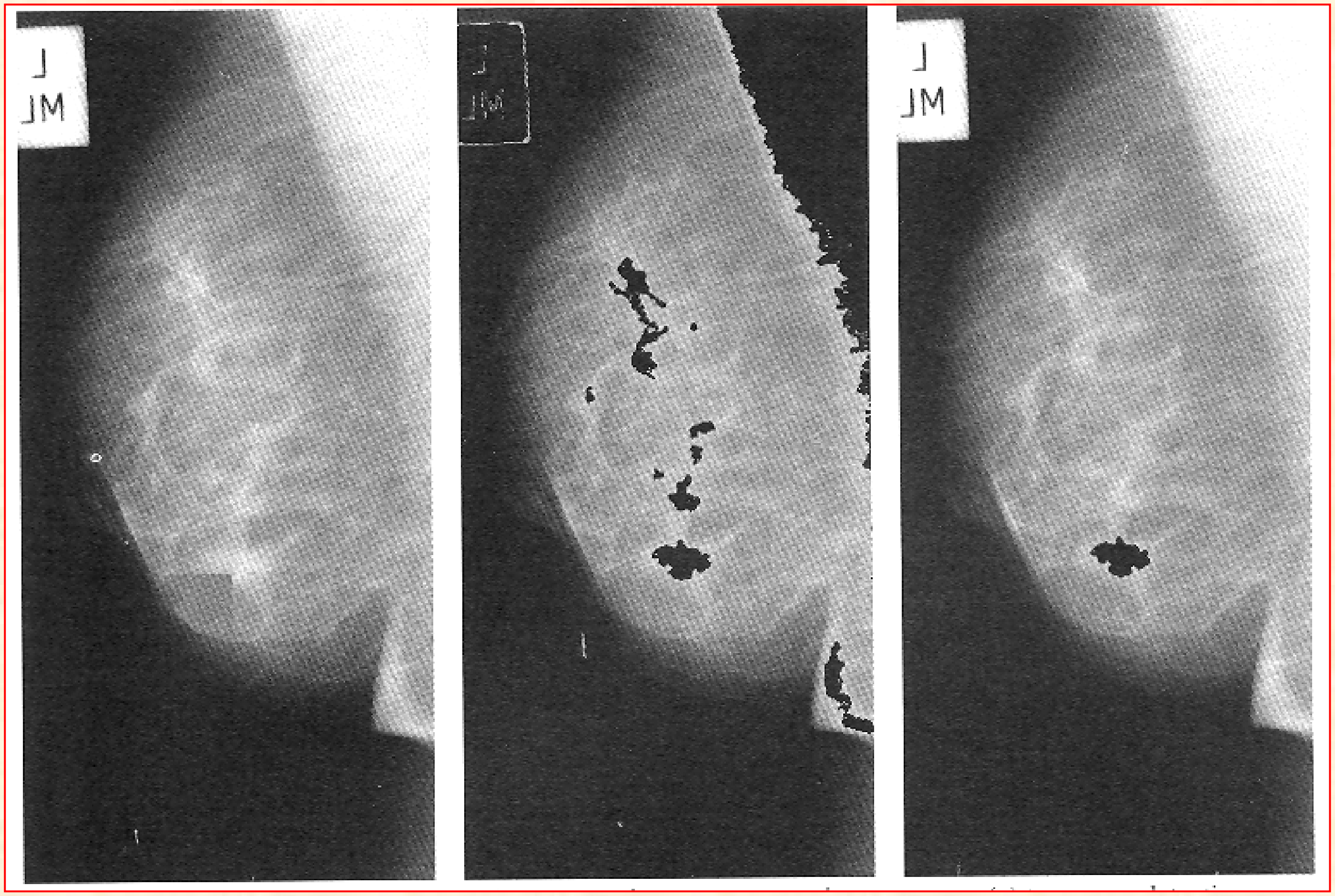
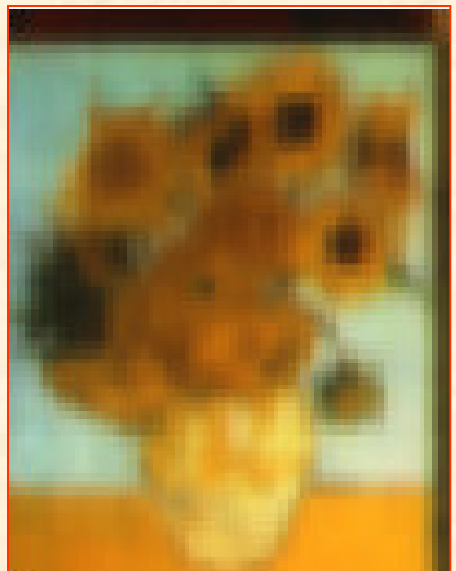
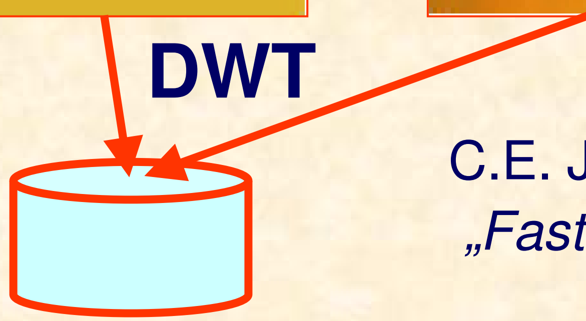


Image database quering

„Idea" of the search image or a copy of the image

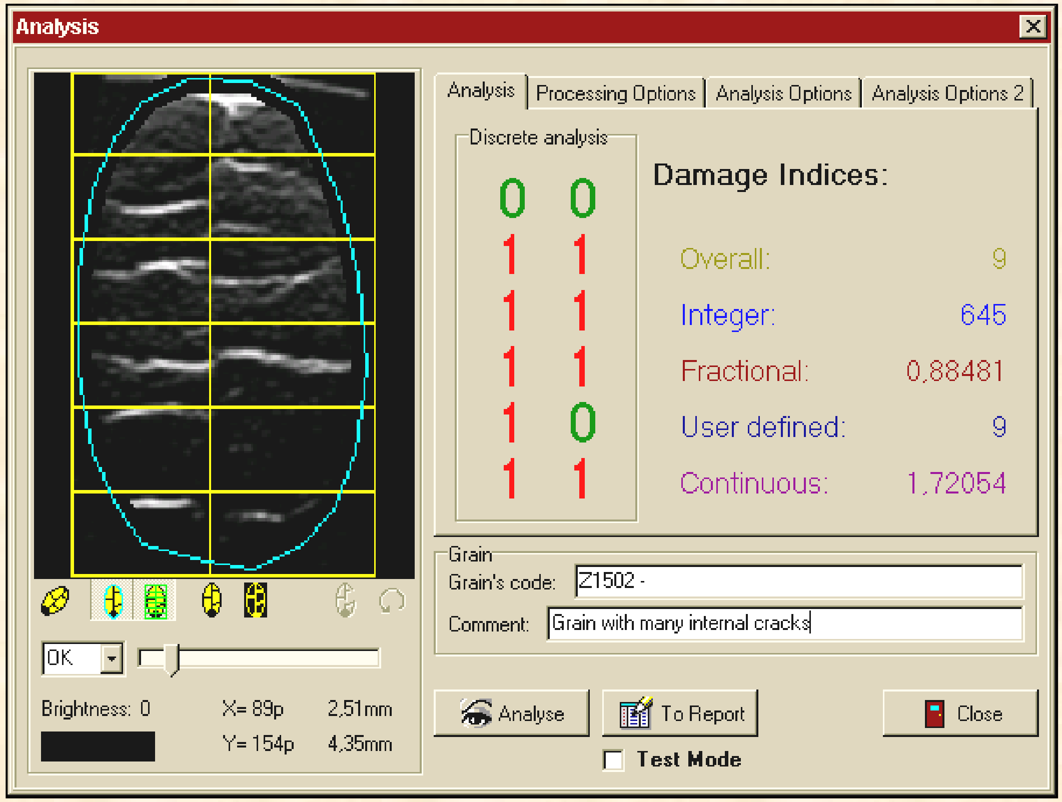
database hit





C.E. Jacobs, A. Finkelstein, D.H. Salesis,
„Fast multiresolution image quering", 1999

Analysis of X-ray images of grains



P. Strumillo, J. Niewczas, P. Szczypiński, P. Makowski, W. Wozniak, “Computer system for analysis of X–ray images of wheat grains”, *International Agrophysics*, 1999, vol. 13, No. 1, pp. 133–140.

Automatic text detection