# CLASSIFICATION/RECOGNITION

**Readings:**

**Szeliski: Chapter 14.1,14.3**

**Some Slides adapted from Univ. of Washington**

**SONKA et.al**

**Classification Chapter-9.2**



## What is pattern recognition?

"The assignment of a physical object or event to one of several prespecified categeries" -
- Duda & Hart

- A **pattern** is an object, process or event that can be given a name.
- A **pattern class** (or category) is a set of patterns sharing common attributes and usually originating from the same source.
- During **recognition** (or **classification**) given objects are assigned to prescribed classes.
- A **classifier** is a machine which performs classification.



## Examples of applications

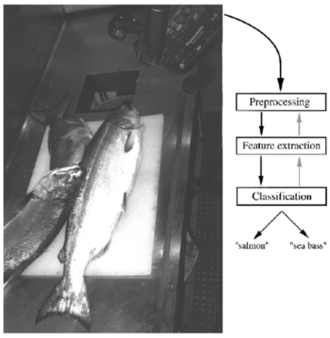
- **Optical Character Recognition (OCR)**
  - Handwritten: sorting letters by postal code, input device for PDA's.
  - Printed texts: reading machines for blind people, digitalization of text documents.
- **Biometrics**
  - Face recognition, verification, retrieval.
  - Finger prints recognition.
  - Speech recognition.
- **Diagnostic systems**
  - Medical diagnosis: X-Ray, EKG analysis.
  - Machine diagnostics, waster detection.
- **Military applications**
  - Automated Target Recognition (ATR).
  - Image segmentation and analysis (recognition from aerial or satelite photographs).

## Approaches

- **Statistical PR:** based on underlying statistical model of patterns and pattern classes.
- **Structural (or syntactic) PR:** pattern classes represented by means of formal structures as grammars, automata, strings, etc.
- **Neural networks:** classifier is represented as a network of cells modeling neurons of the human brain (connectionist approach).

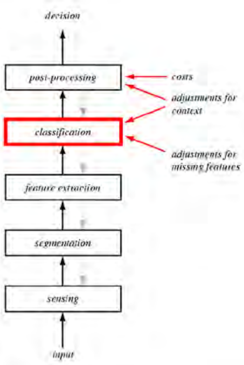


## Classification






## Pattern Recognition & IP

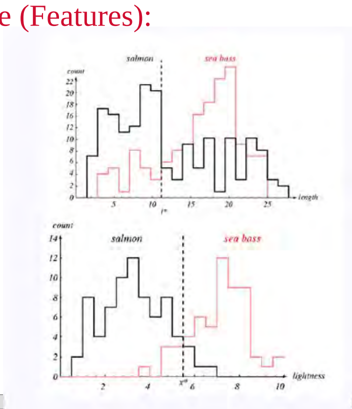


# Features, Patterns

# Featurespaces

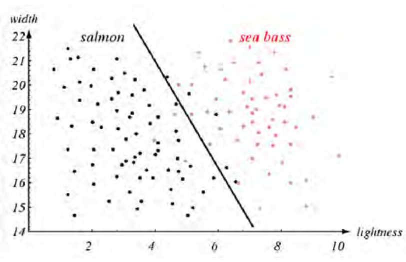


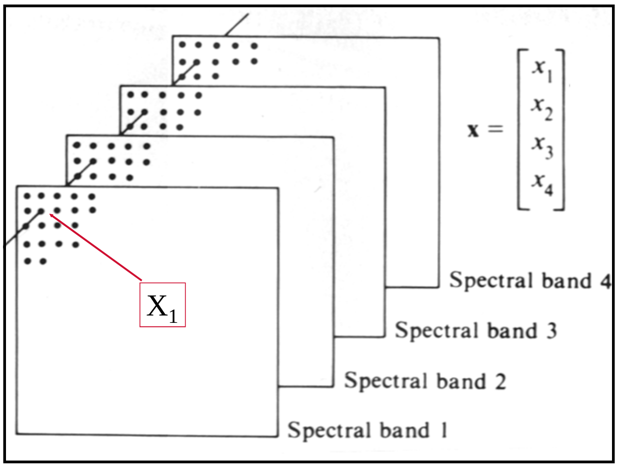
## Merkmale (Features):






## Featurespace






## Classification Principle



### Classification

Feature Vector → Class Label

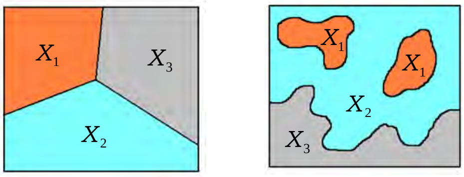
$$\Theta : X \rightarrow \Omega$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \rightarrow \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_c \end{pmatrix} = \Omega$$

## Classifier

A classifier partitions feature space $X$ into **class-labeled regions** such that

$X = X_1 \cup X_2 \cup \ldots \cup X_{|Y|}$ and $X_1 \cap X_2 \cap \ldots \cap X_{|Y|} = \{0\}$




The classification consists of determining to which region a feature vector **x** belongs to.

Borders between **decision boundaries** are called decision regions.



## A priori probability

$P(\omega_1)$ , $P(\omega_2)$ … a priori propability

sea bass salmon

$P(\omega_1) + P(\omega_2) = 1$ (Only $\omega_1$ and $\omega_2$ )

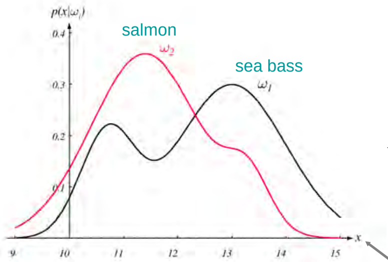
Simple Classification:

$$S = \begin{cases} \omega_1 & if \quad P(\omega_1) > P(\omega_2) \\ \omega_2 & else \end{cases}$$

Without Features



## Class-conditional probability density




$$\int_{-\infty}^{\infty} p(x|\omega_1)dx = 1$$

$$\int_{-\infty}^{\infty} p(x|\omega_2)dx = 1$$

Feature (e.g. Brightness)

## Posterior Probability

Bayes Theorem:

$$\boxed{P(\omega_j|\vec{x}) = \frac{p(\vec{x}|\omega_j)P(\omega_j)}{p(\vec{x})}} \quad \text{with} \quad p(\vec{x}) = \sum_{j=1}^{c} p(\vec{x}|\omega_j)P(\omega_j)$$

Classification:

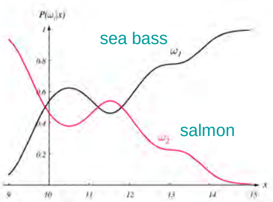
$$\boxed{S = \begin{cases} \omega_1 & falls \quad P(\omega_1|\vec{x}) > P(\omega_2|\vec{x}) \\ \omega_2 & sonst \end{cases}}$$

Error Probability:

$$P(error|\vec{x}) = \begin{cases} P(\omega_1|\vec{x}) & falls \quad S = \omega_2 \\ P(\omega_2|\vec{x}) & falls \quad S = \omega_1 \end{cases}$$



## Posterior Probability




$$P(\omega_1|\vec{x}) + P(\omega_2|\vec{x}) = 1$$

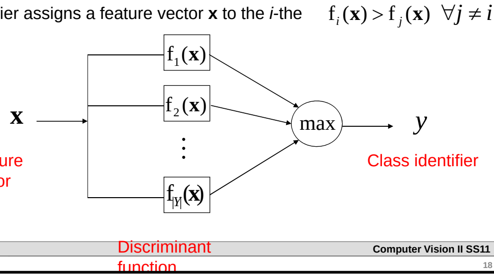


## Representation of classifier

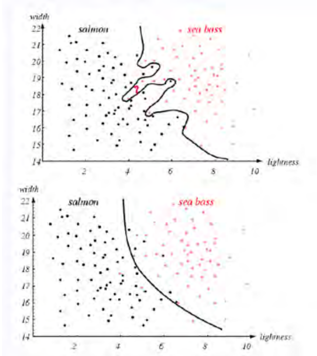
A classifier is typically represented as a set of discriminant functions

$$\mathrm{f}_i(\mathbf{x}): X \rightarrow \Re, i = 1, \ldots, |Y|$$

The classifier assigns a feature vector **x** to the *i*-the class if $\mathrm{f}_i(\mathbf{x}) > \mathrm{f}_j(\mathbf{x}) \quad \forall j \neq i$



# Decision Function:






## Bayes Classifier

Classifier: $\Theta_{Bayes}(\vec{x}) = S_{\max}(\Phi_{Bayes}, \vec{x})$

$$S_{\max}(\Phi, \vec{x}) = \omega_i \quad \text{falls} \quad \varphi_i(\vec{x}) = \max_{j=1,2,\ldots,c} \varphi_j(\vec{x})$$

Always optimal decision $\varphi_{Bayes_i}(\vec{x}) = P(\omega_i \mid \vec{x})$

not known!!

$$P(\omega_i \mid \vec{x})$$



## Bayes Classifier

Decision Function $\boxed{\varphi_{Bayes_i}(\vec{x}) = P(\omega_i \mid \vec{x})}$

Bayes Theorem:

$$P(\omega_i \mid \vec{x}) = \frac{p(\vec{x} \mid \omega_i) P(\omega_i)}{\sum_{k=1}^{c} p(\vec{x} \mid \omega_k) P(\omega_k)}$$

Decision Function

$$\varphi_{Bayes_i}(\vec{x}) = p(\vec{x} \mid \omega_i) P(\omega_i)$$

$$\ln \varphi_{Bayes_i}(\vec{x}) = \ln p(\vec{x} \mid \omega_i) + \ln P(\omega_i)$$

## Parametric Classifier

Assumption that all

$$p(\vec{x} \mid \omega_i)$$

a distributed according to a parametric model

Learning: Estimating model parameters



## Multivariate Normaldistribution

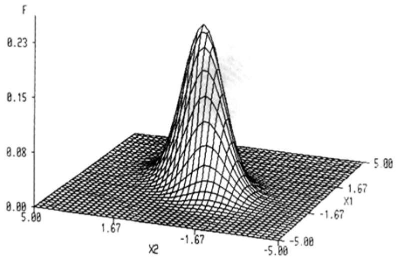
$$f(x_1, x_2, \ldots, x_l) = \frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu})\}$$

μ Mean
Σ Covariance
|Σ| Determinant

$$\Sigma = E\left[(\vec{x}-\vec{\mu})(\vec{x}-\vec{\mu})^T\right]$$
$$\vec{\mu} = E[\vec{x}]$$



## Bivariate Normalverteilung

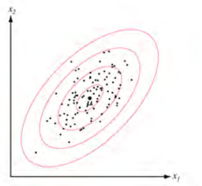


## 2D-Distribution

Equidistances are Ellipses (Mahalanobis Distance)

$$(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu}) = c^2$$

Center $(\mu_1 \mu_2)$:





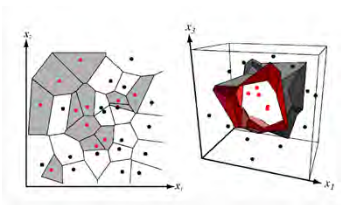
## Supervised Classification

We have labeled Training Data



## Nearest Neighbor Classification

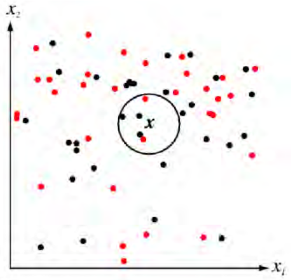
Assign Pattern class of closest training pattern



For n→∞ at maximum twice Bayes error rate.

## K-Nearest Neighbor Classifier

Assign Pattern x majority among k nearest neighbors

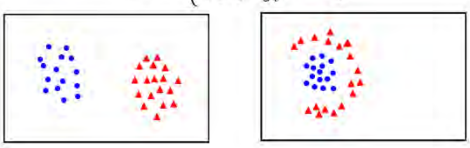





## Binary Classification

Given training data $(\mathbf{x}_i, y_i)$ for $i = 1 \dots N$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, -1\}$, learn a classifier $f(\mathbf{x})$ such that

$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

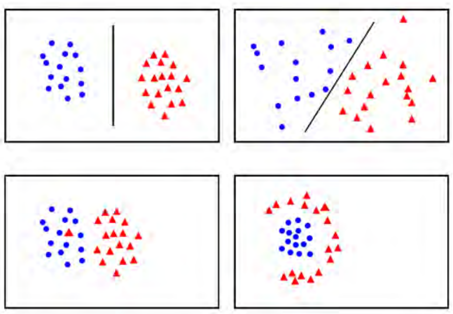




## Linear Separable

linearly separable

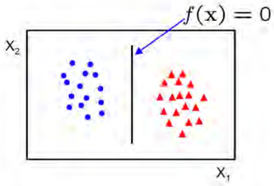
not linearly separable



## Linear Classifier

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



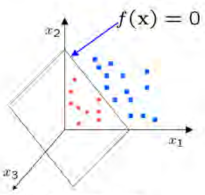

- in 2D the discriminant is a line
- $\mathbf{w}$ is the normal to the plane, and b the bias
- $\mathbf{w}$ is known as the weight vector



## Linear Classifier

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$




- in 3D the discriminant is a plane, and in nD it is a hyperplane

For a K-NN classifier it was necessary to `carry' the training data
For a linear classifier, the training data is used to learn **w** and then discarded
Only **w** is needed for classifying new data



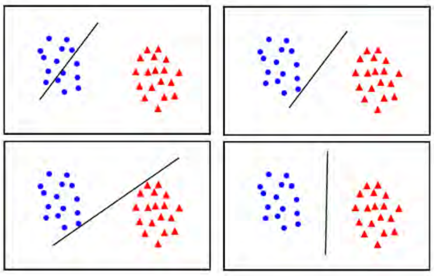
## How to find parameters

Various Algorithms to find parameters w:

- Perceptron and Variants
- LMS
- Linear Discriminant Analysis (LDA)

- Support Vector Machines SVMs

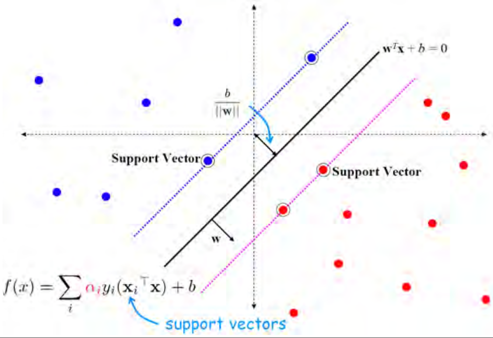
## Best w?



- maximum margin solution: most stable under perturbations of the inputs



## Support Vector Machines






## Example Object Detection

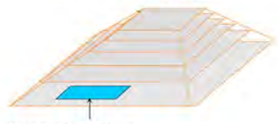
Objective: detect (localize) standing humans in an image

- cf face detection with a sliding window classifier

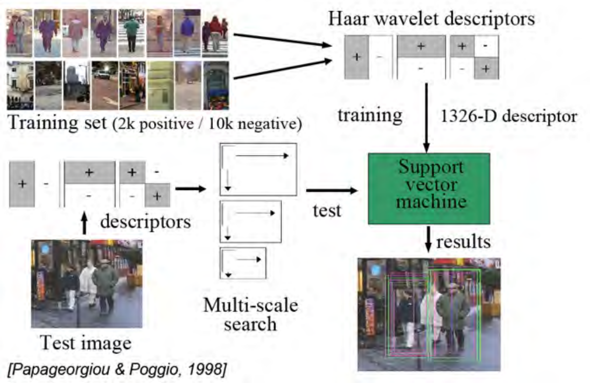
- reduces object detection to binary classification
- does an image window contain the object or not?

## Detection Phase

- Scan image(s) at all scales and locations
- Extract features over windows
- Run window classifier at all locations
- Fuse multiple detections in 3-D position & scale space
- Object detections with bounding boxes



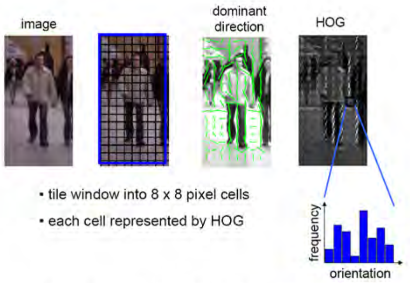



*[Papageorgiou & Poggio, 1998]*



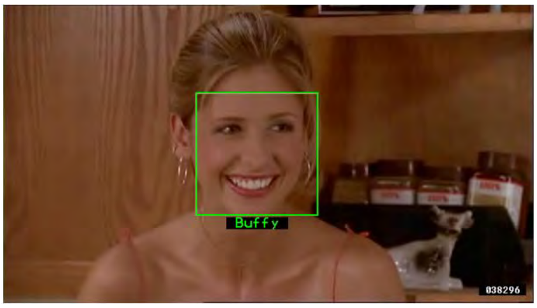
## Feature HOG: Histogram of Oriented Gradients




- tile window into 8 x 8 pixel cells
- each cell represented by HOG

Feature vector dimension = 16 x 8 (for tiling) x 8 (orientations) = 1024

## Face Detection (Buffy)






## Unsupervised Classification

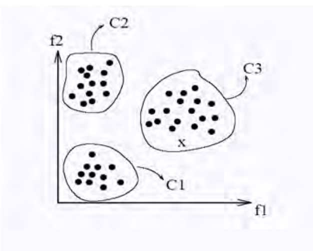
## Clustering

We do not have labeled Training Data



## Clustering

N Patterns in K-Clusters



## Clustering Tasks

**Familiarity** Learn what is typical
**Clustering** Which class a pattern belongs to
**Prototyping** Find prototypes
**Encoding** Data compression
**Feature mapping** Topographic map of the input
**Principal Component Analysis** Eigenvectors of Correlation Matrix



## What is Similarity?

Similarity is hard to define, but…
"*We know it when we see it*"

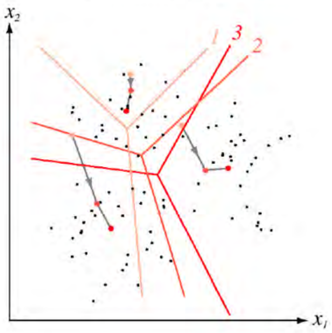
The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.



## K-Means Algorithm

Select initial Cluster centers c_i from data points p_j
Assign all data points to nearest cluster center
Update cluster center by calculating new mean value
Repeat until there are no more changes

## K-Means Algorithm






## Color-Segmentation

RGB Trippels are Datapoints.

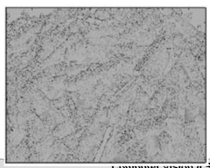
Clustering using:
- k-means
- fuzzy k-means



## Image Compression

Compressionrate: 1:50.

8x8-Blocks.



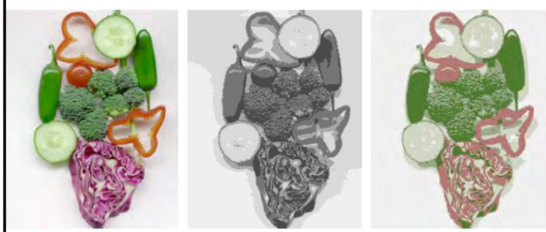
## Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together
- Agglomerative clustering
  - attach closest to cluster it is closest to
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - repeat

- Point-Cluster distance
  - single-link clustering
  - complete-link clustering
  - group-average clustering
- Dendrograms
  - yield a picture of output as clustering process continues



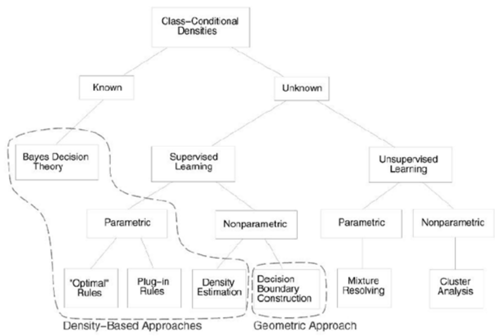
Image Clusters on intensity Clusters on color



K-means clustering using intensity alone and color alone



## OUTLOOK

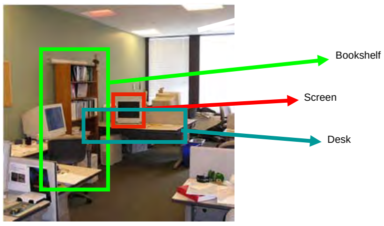


# Object Recognition



## A computer vision goal

Recognize many different objects under many viewing conditions in unconstrained settings.






## Object Recognition

(D. Lowe 2003)

Definition: Identify an object and determine its pose and model parameters

Commercial object recognition
- Currently a $4 billion/year industry for inspection and assembly
- Almost entirely based on template matching

Upcoming applications
- Mobile robots, toys, user interfaces
- Location recognition
- Digital camera panoramas, 3D scene modeling

## Where is the field of computer vision?

There are efficient solutions for detecting a single object category and view:

Viola & Jones 2001; Papageorgiou & Poggio 2000, …

detecting particular objects:

Lowe, 1999

Recognizing objects in isolation

Leibe & Schiele, 2003;



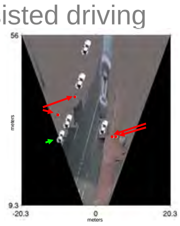
## Computational photography





## Assisted driving

Pedestrian and car detection




Lane detection

- Collision warning systems with adaptive cruise control,
- Lane departure warning systems,
- Rear object detection systems,

## Improving online search




## Recognition problems

- What is it?
  - Object and scene recognition
- Who is it?
  - Identity recognition
- Where is it?
  - Object detection
- What are they doing?
  - Activities
- All of these are **classification** problems
  - Choose one class from a list of possible candidates



## What is recognition?

- A different taxonomy from [Csurka *et al.* 2006]:
- Recognition
  - Where is *this* particular object?
- Categorization
  - What *kind* of object(s) is(are) present?
- Content-based image retrieval
  - Find me something that looks similar
- Detection
  - Locate *all* instances of a given class

## Issues in Object Recognition

We want a machine to be able to identify thousands of different objects in their usual environment.

Representation
- Local vs. Global Appearance based
- Discriminative vs. Generative

Recognition
- Robustness

Learning
- Robustness
- Minimal samples



## Object Recognition

Hierarchy of problems
- classify image according to objects present
- localize objects (position, scale, pose, ...)
- segment objects

Research frontier: 100s or 1000s of categories



## Issues

Variability due to changes in
- location
- scale
- pose
- lighting
- non-rigid deformation
- intra-class variability

High dimensionality of input space

Occlusion

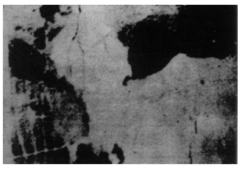
Large number of object classes

Multiple occurrences of objects

Some categories semantic rather than visual e.g. chairs

## Problems

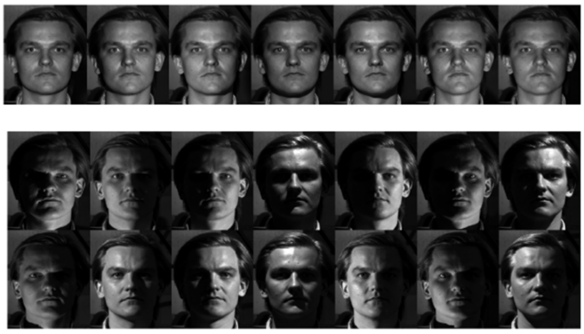
Segmentation:



Pose/Shape:



## Illumination





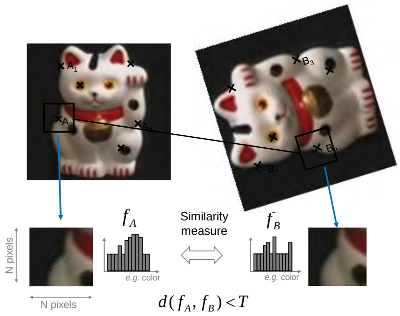
## What is a chair

Image from H. Bülthoff MPI Tübingen

Invariance?



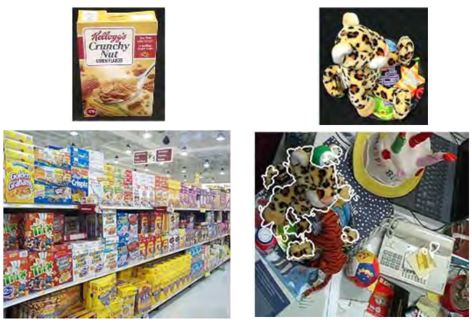
# Local Object Recognition




1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors



# Single object recognition



## Single object recognition


- Lowe, et al. 1999, 2003
- Mahamud and Herbert, 2000
- Ferrari, Tuytelaars, and Van Gool, 2004
- Rothganger, Lazebnik, and Ponce, 2004
- Moreels and Perona, 2005
- …



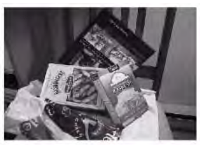
## Planar object recognition [Lowe]

- Use SIFT features
- Verify affine (or homography) geometric alignment



## Planar object recognition [Lowe]

- Use SIFT features
- Verify affine (or homography) geometric alignment




## 3D object recognition [Lowe]

- Extract object outlines with background subtraction



## 3D object recognition [Lowe]

- Use 3 matches to recognize
- Use additional matches for verification
- Tolerant to occlusions



## Feature-based recognition

- How can we scale to millions of objects?
- Comparison to *all* stored objects/features is infeasible.
- **Answer**:
- quantize features into *words* [Csurka *et al.* 04]
- use information retrieval (inverted index)
- use *metric tree* for faster quantization (NN) [Nister & Stewenius 05]