Introduction to Video analysis

DISI Università di Genova

Francesca Odone

odone@disi.unige.it

Nicoletta Noceti

noceti@disi.unige.it

VIT Vision for Innovative Transport

CAPACITIES

Outline of the course

A *practical* introduction to video analysis organized in 3 modules

- Day 1
  - introductory concepts (1H)
  - introduction to the development framework (1H)
- Day 2
  - Motion segmentation in the lab (2H)
- Day 3
  - Object detection in the lab (2H)

VIT Vision for Innovative Transport

CAPACITIES

DAY 1

- Introductory concepts…
- … and then we move to the lab!

Definitions

- **Image sequences**:

  a series of N images *(frames)* acquired at discrete time instants $t_k = t_0 + kT$, where T is a fixed time interval and k = 1, …, N

- **Acquisition rate***:*

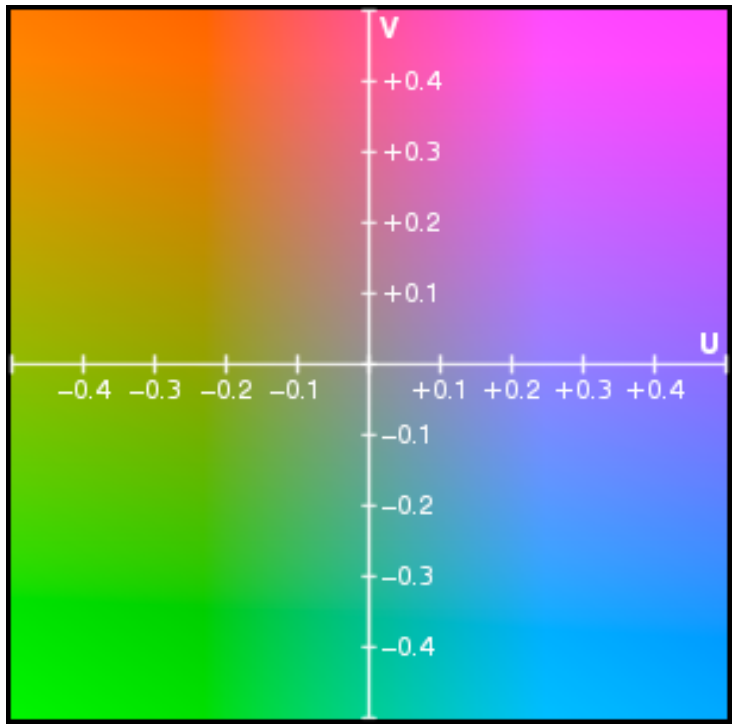
  it measures the acquisition speed. A typical rate is the *frame rate,* 24 fps (frames per second)

Definitions

- It is important that T is small enough so that the image sequence is a “good” approximation of a continuously evolving scene.

Video color space

- The color space more common in videos is **YUV**
  - Y *(luminance component)* carries information on the pixel brightness
  - U and V *(chrominance components)* carry color information



Video color space

Single Frame YUV420:

Position in byte stream:

VIT Vision for Innovative Transport

CAPACITIES

Video formats

- Typical digital video formats:
  - PAL: 720x576 pixel, 25 fps
  - NTSC: 720x480 pixel, 30 fps
- Video compression algorithms:
  - Mpeg2: standard in DVDs
  - Mpeg4: when a more compact representation is required (Internet, portable readers - DivX, Xvid, QuickTime, iPod Video…)

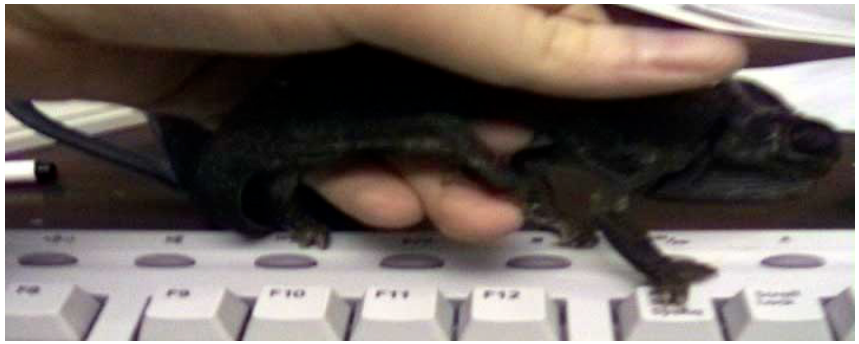
Video vs image sequence

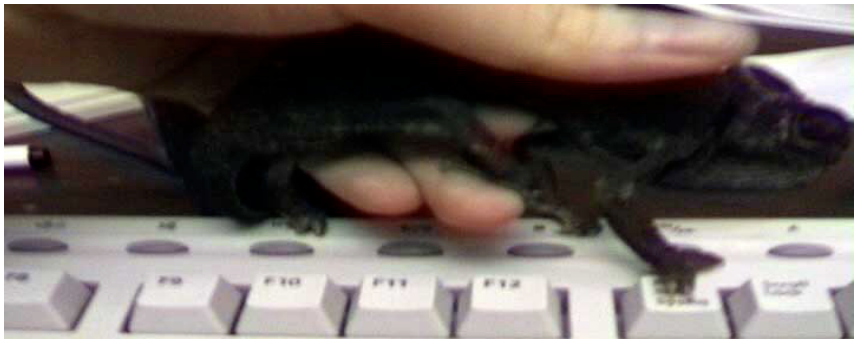
VIT Vision for Innovative Transport

CAPACITIES

Interlaced video

- An interlaced video is composed of *fields* with a smaller vertical resolution than the original video
  - The PAL format, for instance, is 720x288px



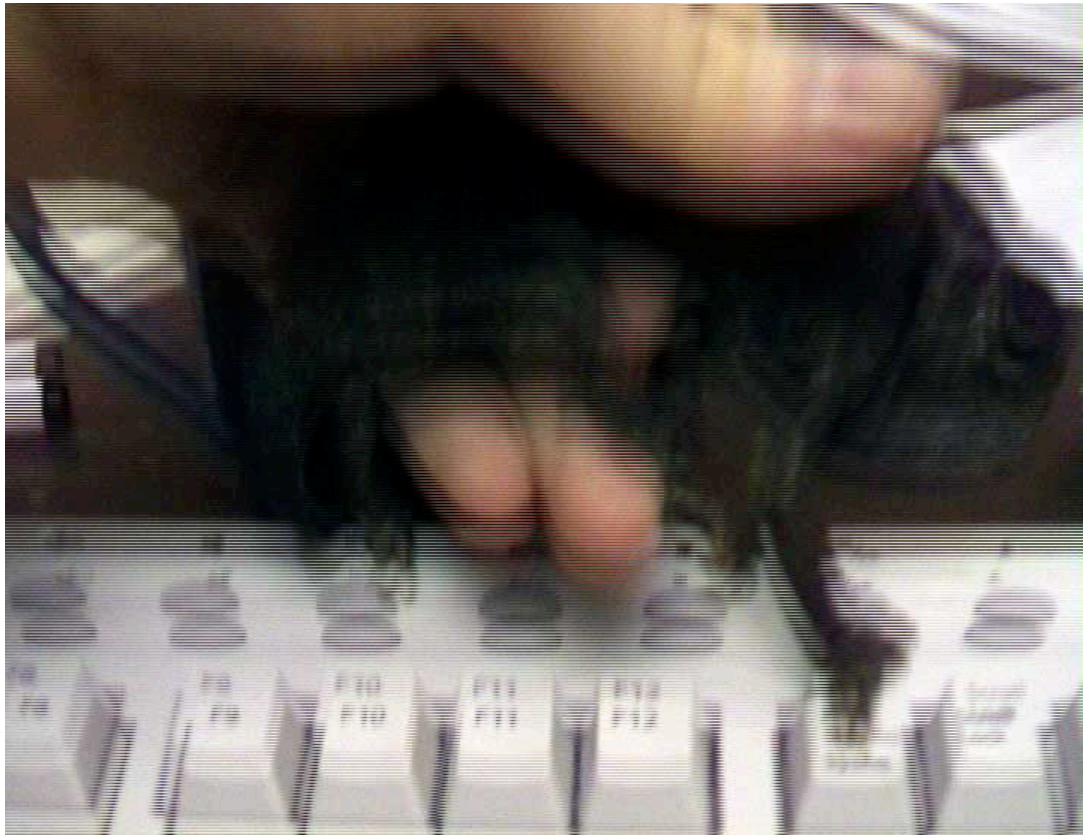


Interlaced videos

- The two fields are shown in a sequence and they exploit the image persistence on the display (and on the retina)
- At a reasonable speed the observer does not perceive that part of the observed visual information is static
- They allow to reache the same frame rate at a smaller transmission cost

Interlaced videos

- A single frame of an interlaced video



Deinterlacing

Methods

- Waving (temporal resolution loss)
- Line doubling (spatial resolution loss)
- Blending (spatio-temporal resolution loss)

- Motion compensation

Motion analysis is useful for..

- Inferring information
  - Changes in the scene
  - Objects evolving in time (matching along the temporal component)
  - Object tracking (and prediction)
- Applications
  - Video-surveillance and monitoring
  - Structure from motion
  - Video annotation
  - ....

In the context of VIT project

- Video analysis is being used for
  - 2D train scanning
    - Buildiling the train profile via feature matching and mosaic construction
  - Identification of people moving in forbidden areas
    - Run-time background construction and update
    - Motion segmentation
    - Object tracking

DAY 2

- Motion analysis lab
  - Background construction and update: we will try different methods; we will discuss issues related to outdoor video analysis
  - Change detection
  - Object tracking: practical hints and a little theory will accompany the lab
  -

Building blocks: change detection

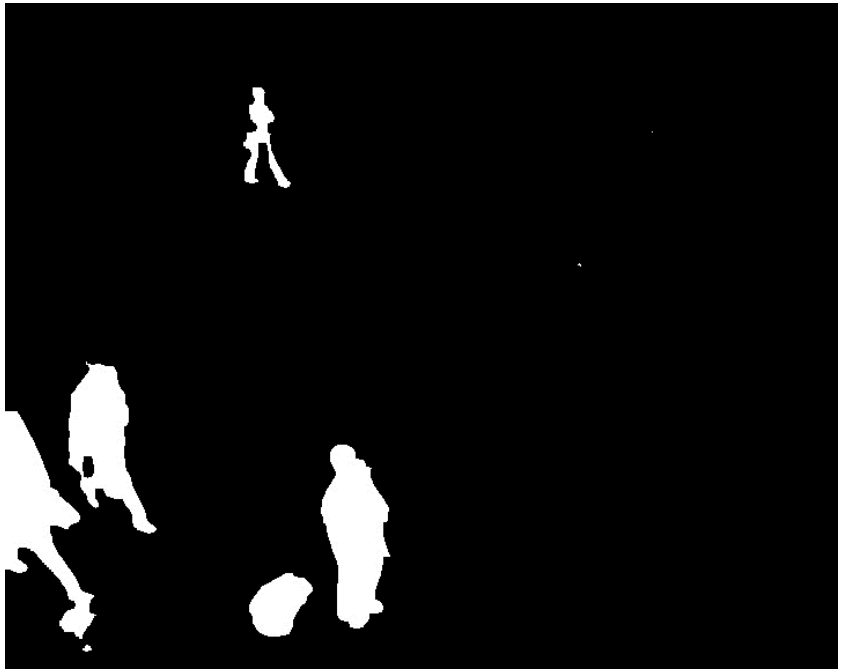
- **Motion segmentation:** localize image regions with a common motion pattern
- If the camera is still motion segmentation is usually referred to as **change detection**
- Change detection is commonly addressed comparing each frame of the sequence with a reference model of the empty scene (the so-called **background**)
- Changes with respect to the background are usually caused by moving objects (**foreground**)

Building blocks: change detection

- Assuming that we can rely on a reference image I_{REF}, change detection produces a binary map of the scene regions that changes w.r.t. the reference:

$$BM_t(x,y) = \begin{cases} 1 & \text{if } \left|I_t(x,y) - I_{REF}(x,y)\right| > s \\ 0 & \text{otherwise} \end{cases}$$

Building blocks: change detection



Building blocks: background construction

METHOD 1: frames average:

- The simplest model is an average of N video frames:

$$I_{REF} = B = \frac{1}{N}\sum_{t=1}^{N} I_t$$

- This is not always possible. WHY?

.....

N frames

Building blocks: background construction

METHOD 2: **Running average**

$$B_t(i,j) = \begin{cases} B_{t-1}(i,j) & if \left| I_t(i,j) - B_{t-1}(i,j) \right| \geq s \\ (1-\alpha)B_{t-1}(i,j) + \alpha I_t(i,j) & otherwise \end{cases}$$

VIT Vision for Innovative Transport

CAPACITIES

Limits of simple algorithms

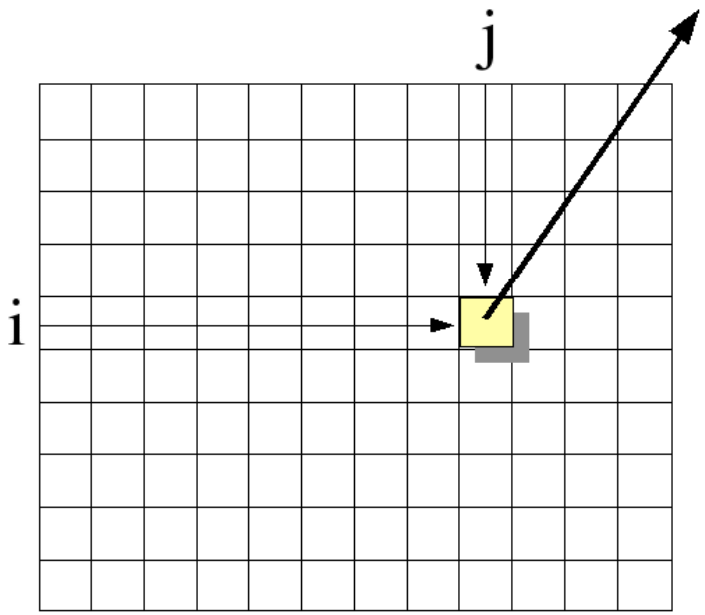
- In the previous methods each pixel of the background is described by a single value, therefore it is not possible to model evolving (e.g., periodic) patterns
- Examples: waving leaves, trembling objects..

Building blocks: background construction

METHOD 3: Codebook model

- Codebooks allow for more complex pixel models

$$C_{ij} = \{c_{ij}^{0}, c_{ij}^{1}, ..., c_{ij}^{N}\}$$

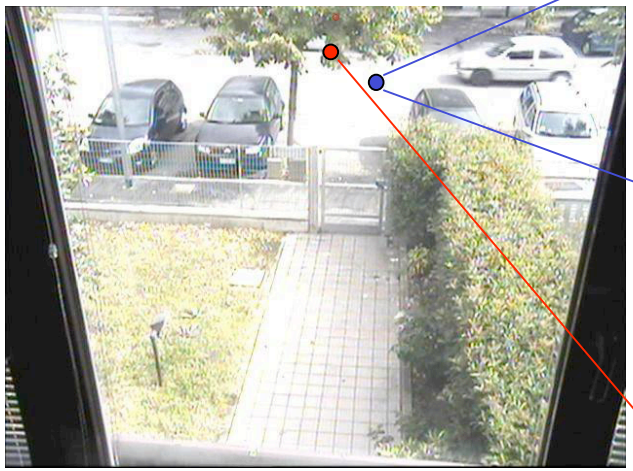


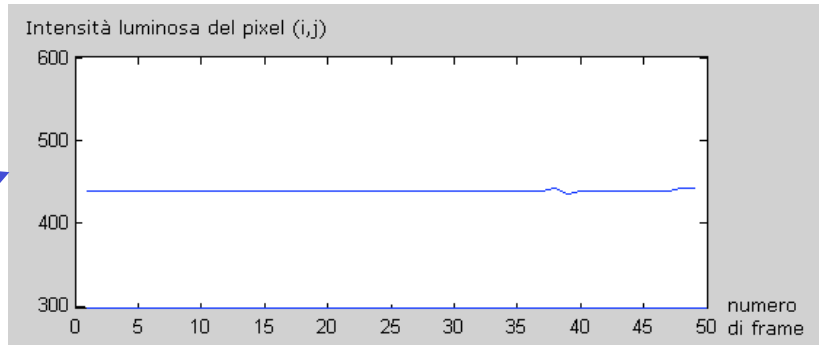
The codebook approach

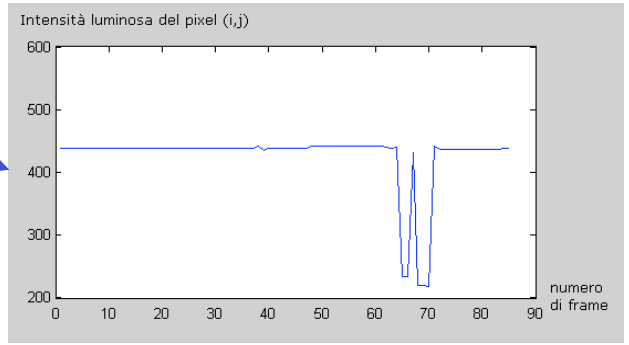
- Each pixel is associated a codebook
- Each codebook is formed by a set of codewords
- A codeword encodes information on the appearance and the dynamics of a pixel

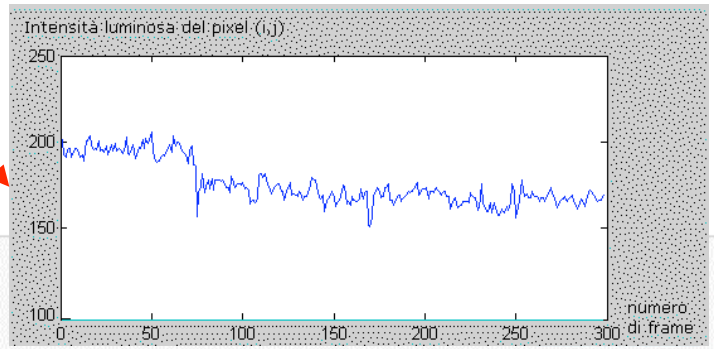
- A **training phase** allow us to build a model of the scene

Codebook model





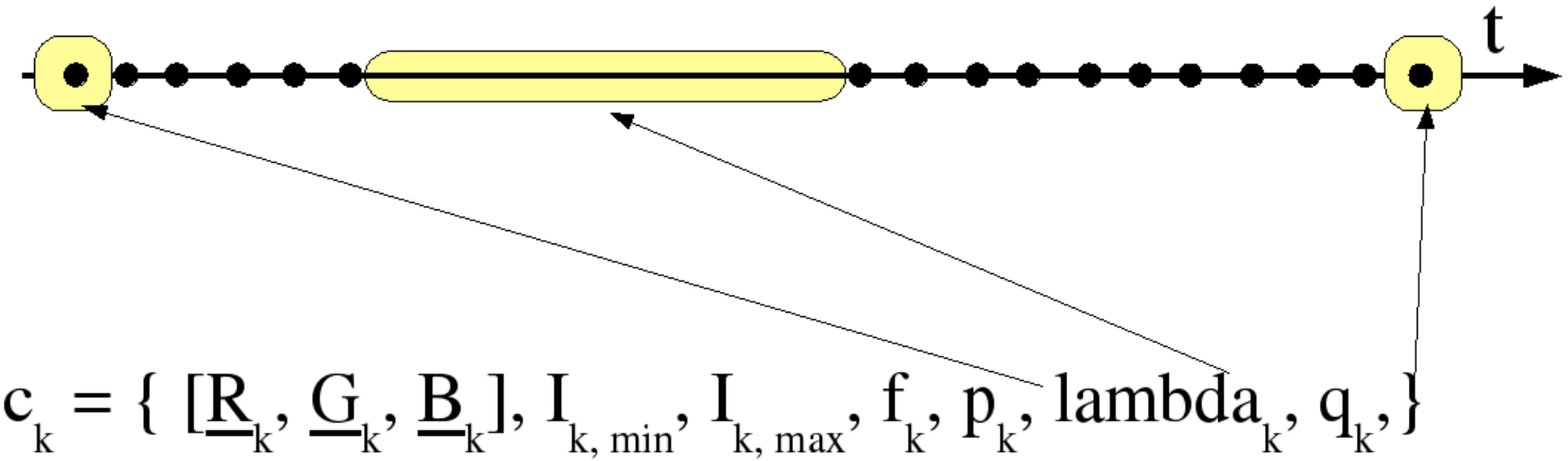




VIT Vision for Innovative Transport

CAPACITIES

Codebook model: codewords



- **Appearance** is described by information on color and intensity values
- **Dynamics** is described by information on the codeword *life* (when it was first observed, how frequent is it observed, etc)

Codebook construction

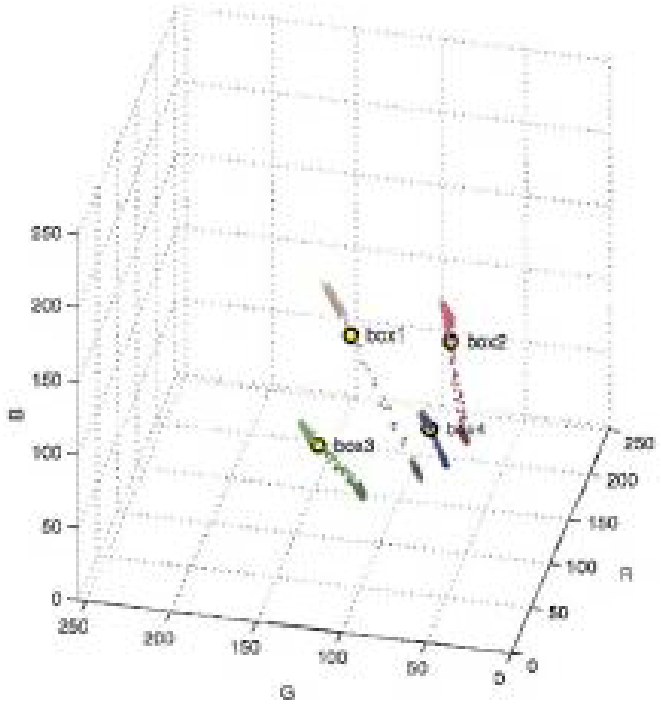
- A training set of N frames is used to initialize the model
- We associate a codebook to each position p=(i,j)
- For each p=(i,j) we have N observations:

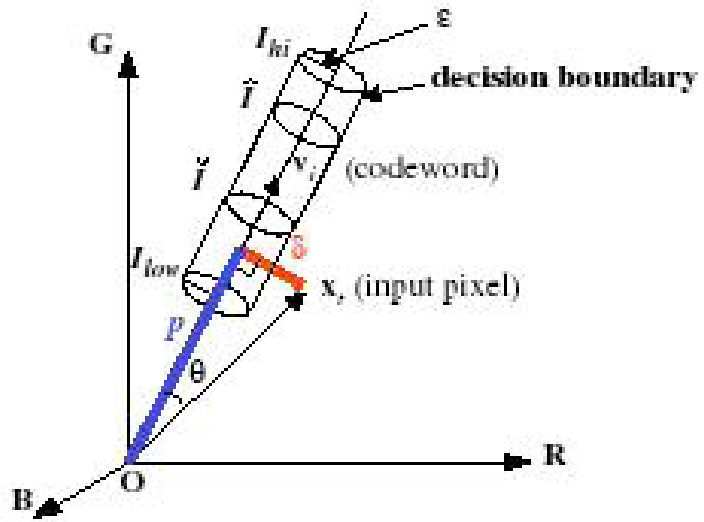
$$\{x_{p,t}=(R_{p,t}, G_{p,t}, B_{p,t})\}_{t=1,\ldots,N}$$

Codebook construction

- For each $x_{p,t}=(R_{p,t}, G_{p,t}, B_{p,t})$ in O (t=1,…N):
  - Look for a codeword c_i *appropriate* for $x_{p,t}$
  - If it does not exist create a new codeword
    $\rightarrow$ [I, I, 1, t, t-1, t]
    $\rightarrow [R_{p,t}, G_{p,t}, B_{p,t}]$
  - If it does exist update its parameters
    $\rightarrow [\min(I, I_i), \max(I, I_i), f_i+1, p_i, \max(\lambda_i, t-q_i), t]$
    $\rightarrow \left[\frac{f_i R_i + R_{p,t}}{f_i+1}, \frac{f_i G_i + G_{p,t}}{f_i+1}, \frac{f_i B_i + B_{p,t}}{f_i+1}\right]$

Codebook model: similarity





Change detection with codebooks

- For each $x_{p,t}=(R_{p,t},G_{p,t},B_{p,t})$ search a codeword c_i *close to* $x_{p,t}$
  - If it exists the pixel is classified as a background pixel
  - If it does not exist the pixel is classified as a foreground pixel

What you should obtain

VIT Vision for Innovative Transport

CAPACITIES

DAY 3

- Object detection
  - A brief discussion on learning from examples
  - Applications to video analysis: pedestrian detection

Introduction

- Modeling an object / class...
  - ....one image is not enough!

Learning from examples

DEFINITION (TO LEARN)

- *Gain or acquire knowledge of or skill in (something) by study, experience, or being tought.*
- *Become aware of (something) by information or from observation*

(The New Oxford Dictionary of English)

- The meaning of learning very much depends on the context (education, sociology, artificial intelligence) ...
- In AI the learning paradigm loosely refers to instructing a machine by feeding it with appropriate examples, instead than lines of commands (learning from examples).

Learning from examples

We say that a program for performing a task has been acquired by learning if it has been acquired by any means ofther than explicit programming

(Valiant, 1984)

learning from examples, refers to systems that are trained instead of programmed with a set of examples, that is, a set of input/output pairs.

(Poggio & Smale, 2003)

What is it useful for?

The learning paradigm is useful whenever the underlying process is

- partially unknown,
- too complex, or
- too noisy

to be modeled as a sequence of instructions.

Learning in everyday life

- Security and video-surveillance (including event classification, pedestrian detection, face detection, etc)
- OCR systems
- robot control
- biometrics
- speech recognition
- early diagnosis from medical data
- knowledge discovery in big dataset of heterogeneous data (included the Internet)
- microarray analysis and classification
- stock market predition
- regression applications in computer graphics (view syntesis, etc)

Supervised vs unsupervised

SUPERVISED LEARNING

- The set of examples available contains input/output pairs
- The learning process consists of inferring the input/output relationship from the training data in a generalizing way

UNSUPERVISED LEARNING

- No output data are available
- The goal is to learn some understanding on the process that generated the data

We will mainly focus on supervised learning

Datasets

In supervised learning we are given a set of *input-output* pairs

$$(x_1, y_1), \ldots, (x_n, y_n)$$

that we call a training set

- **Classification:** A learning problem with output values taken from a finite unordered set $C = \{C_1, \ldots, C_k\}$.
  A special case is *binary classification* where $y_i \in \{-1, 1\}$.
- **Regression:** A learning problem whose output values are real $y_i \in IR$

Binary classification

- Input data: $x \in X$, $X \subseteq R^n$
- Classes: $y \in Y$, $Y=\{-1, 1\}$
- Training set: $D_l \equiv \{(x_i, y_i) \in X \times Y\}$, $i=1,\ldots,l$

Drawn from $X \times Y$ according to $P(X,Y)$ fixed but unknown.

Target: $f : X \rightarrow Y$

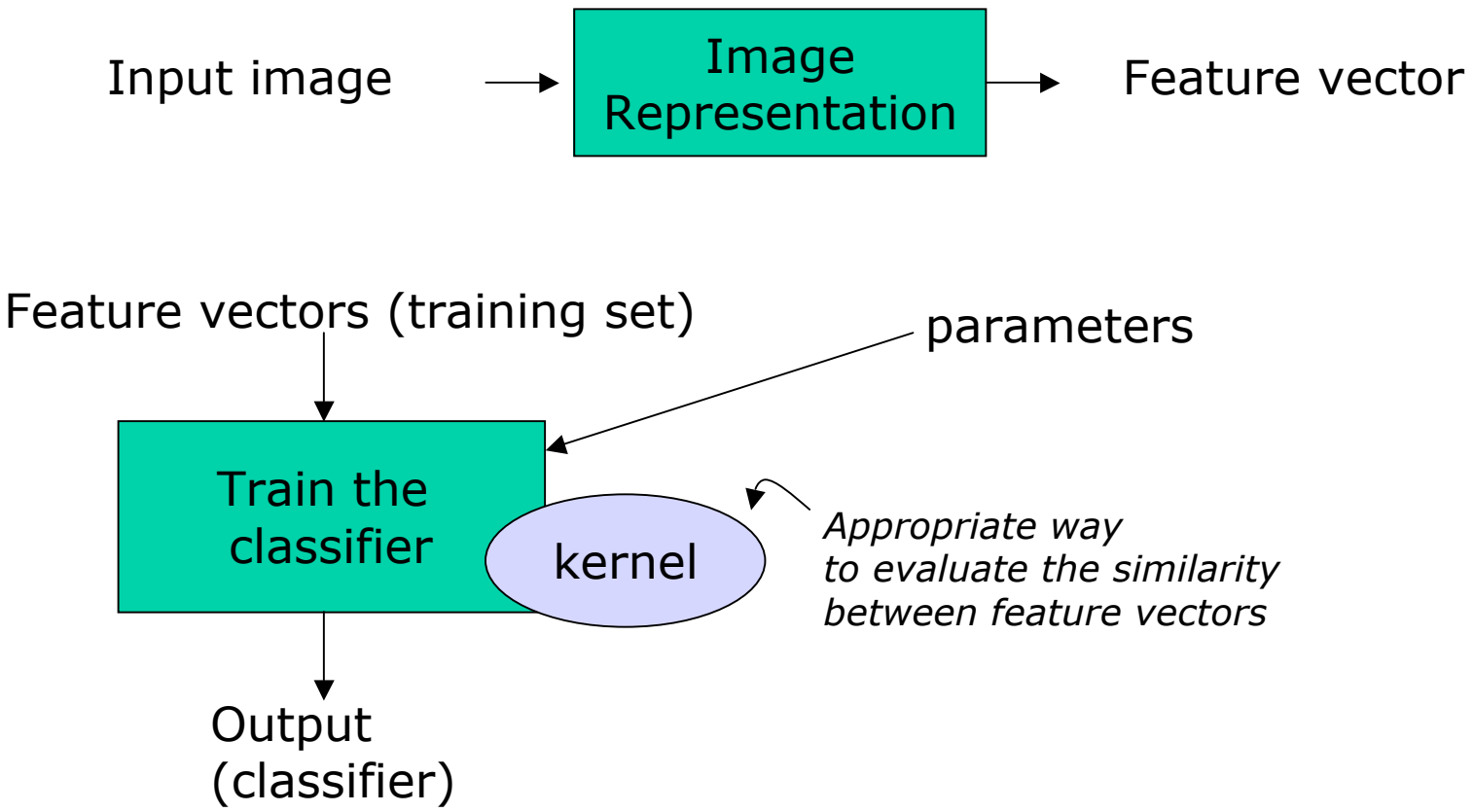
So that given a data x' it is possible to determine to which class it belongs

Example-based detection

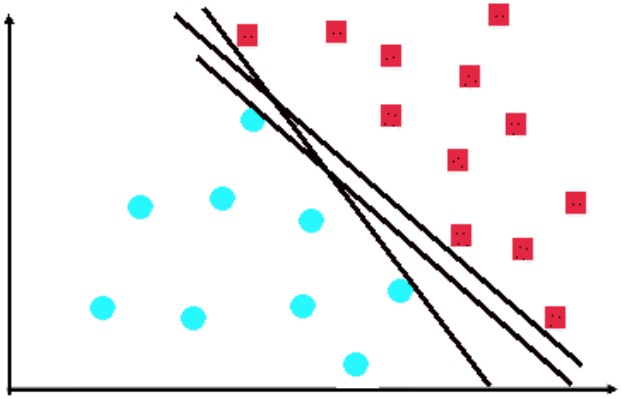
- For each object of interest we build a classifier learning from a dataset of positive images (eg, *people* images) and negative images (eg, *not-people* images)

- The choice of an appropriate dataset is crucial

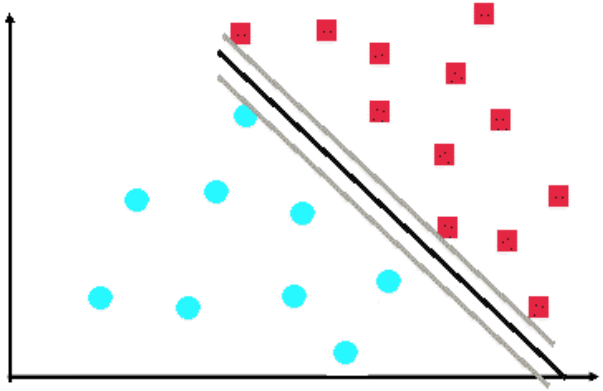
The pipeline



Support Vector Machines SVM



Not unique solution



Optimal separating hyperplane

Problem CL2

$$\min_{\mathbf{w},b} \quad \frac{1}{2} < \mathbf{w} \cdot \mathbf{w} > + C \sum_{i=1}^{n} \xi_i$$

$$\text{subj to} \quad y_i(< \mathbf{w} \cdot \mathbf{x}_i > + b) \geq 1 - \xi_i \quad i = 1, \ldots, n$$
$$i = 1, 2, \ldots, n$$
$$\xi_i \geq 0 \; i = 1, 2, \ldots, n .$$

The parameter C controls the number of misclassified points.

SVM: regularized formulation

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(\mathbf{x}_i)) + \lambda ||f||_K^2$$

- and apply the Hinge loss function

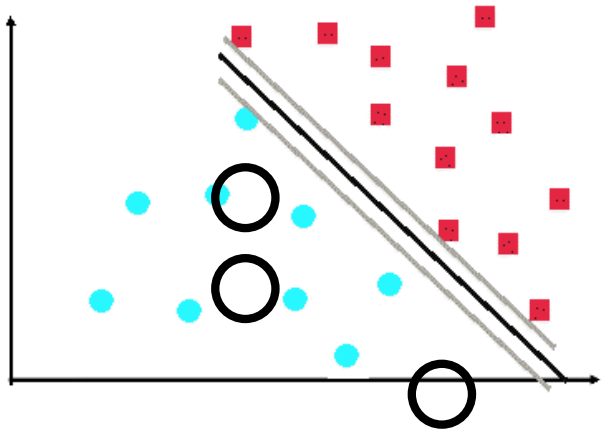
$$V(y, f(\mathbf{x})) = (1 - yf(\mathbf{x}))_+ = \begin{cases} 0 & \text{if } yf(\mathbf{x}) \geq 1 \\ 1 - yf(\mathbf{x}) & \text{otherwise} \end{cases}$$

Solution of SVM

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$$

xi training data
x test data

αi weights
K kernel



Optimal separating hyperplane

The solution in general will be **sparse:**
xi so that αi ≠0: *support vectors*

In the classification case
we consider the sign of the solution

Kernels

- **Linear kernel**

$$K(\mathbf{x}_i, \mathbf{x}_j) = < \mathbf{x}_i \cdot \mathbf{x}_j >$$

- **Polynomial**

$$K(\mathbf{x}_i, \mathbf{x}_j) = (< \mathbf{x}_i \cdot \mathbf{x}_j > +1)^d \text{ con } d \in IN, \quad d \neq 0$$

- **Gaussian**

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j||^2 / 2\sigma^2)$$

- Whenever possible one could choose appropriate similarity functions for the problem at hand
- Some mathematical properties should be met
  - Symmetry and positive definiteness

Data-driven representations

- Overcomplete, general purpose sets of features are effective for modeling visual information
- Many object classes have a peculiar intrinsic structure that can be better appreciated if one looks for symmetries or local geometry

- Examples of features: wavelets, curvelets, ranklets, chirplets, rectangle features ...
- Example of problems: face detection (Heisele *et al,* Viola & Jones, ...), pedestrian detection (Oren *et al., ..*), car detection (Papageorgiou & Poggio)

Data-driven representations

- This approach is inspired by biological systems
  - See, for instance, *B.A. Olshauser and D. J. Field "Sparse coding with an over-complete basis set: a strategy employed by V1?" 1997.*
- Usually this approach is coupled with learning from examples
  - The prior knowledge is embedded in the choice of an appropriate training set
- Problem: usually these sets are very big

  => *Feature selection!*

Feature selection

- Extracting features *relevant* for a given problem
  - What is relevant?
- Often related to dimensionality reduction
  - But the two problems are different

- Theory is lacking
- Many recent works (see the review by Guyon & Elisseeff, JMLR 2003)

The basic algorithm

- We assume a linear dependence between input and output

$$g = Af$$

- $A=\{A_{ij}\}$ is the data matrix
  - $i=1,\ldots,n$ examples
  - $j=1,\ldots,p$ features
- $\mathbf{f}=(f_1,\ldots,f_p)^T$ vector of unknown weights to be estimated
- $\mathbf{g}=(g_1,\ldots,g_n)^T$ output values

The basic algorithm

- Usually p >> n
  - Existence is ensured but not unique
- Measurements are noisy
- Features are correlated

  => ill-posed

- A possible way address the problem is to resort to regularization methods

Regularized least squares

$$\arg\min_{f \in R^p} \left\{ \left| g - Af \right|_2^2 + \lambda \left\| f \right\|^2 \right\}$$

- L2 penalty
- RLS also known as Ridge Regression or Tikhonov
- In general the solution *f* obtained is so that all f_i are different from zero.

L1 penalty

- Recently sparsity-enforcing penalties have been proposed
- They *automatically* enforce the presence of many zeros in *f*
- The L1 norm is convex therefore providing feasible algorithms

$$\arg\min_{f \in R^p} \left\{ \left| g - Af \right|_2^2 + 2\tau \left| f \right|_1 \right\}$$

PROB L1

$$\left| f \right|_1 = \sum_{j=1}^{p} | f_j |$$

VIT Vision for Innovative Transport

CAPACITIES

L1 penalty

- The regularization parameter τ regulates the balance between misfit of the data and penalty
- Also it allows us to vary the degree of sparsity

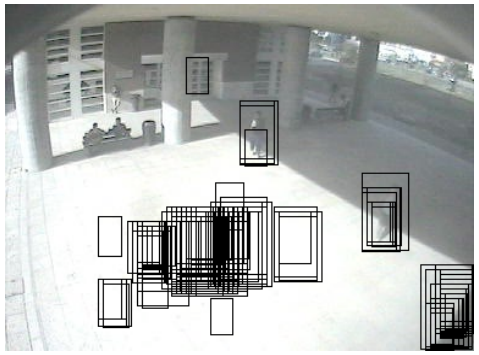
- (PROB L1) is the Lagrangian formulation of the so-called LASSO Problem

Image representations

- We consider the problem of pedestrian detection
- We analyse different local descriptions
  - Rectangle features
  - HoG features
  - Covariance features (later excluded from the analysis for convergenge problems)

Pedestrian detection

- Results on a complex scenario (VISOR repository)



Results

- ROC curve comparing Rectangle features and HoG on the VISOR dataset