Video-surveillance methods and solutions

Francesca Odone
DISI, Dipartimento di Informatica e Scienze dell’Informazione
Universita` degli Studi di Genova
odone@disi.unige.it

http://slipguru.disi.unige.it/

VIT Vision for Innovative Transport

CAPACITIES

Outline

- Introductory concepts and definitions
- Motion analysis
- Video surveillance
- Building blocks:
  - Background construction
  - Change detection
  - Blob tracking
- Applications
- Discussion

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**:

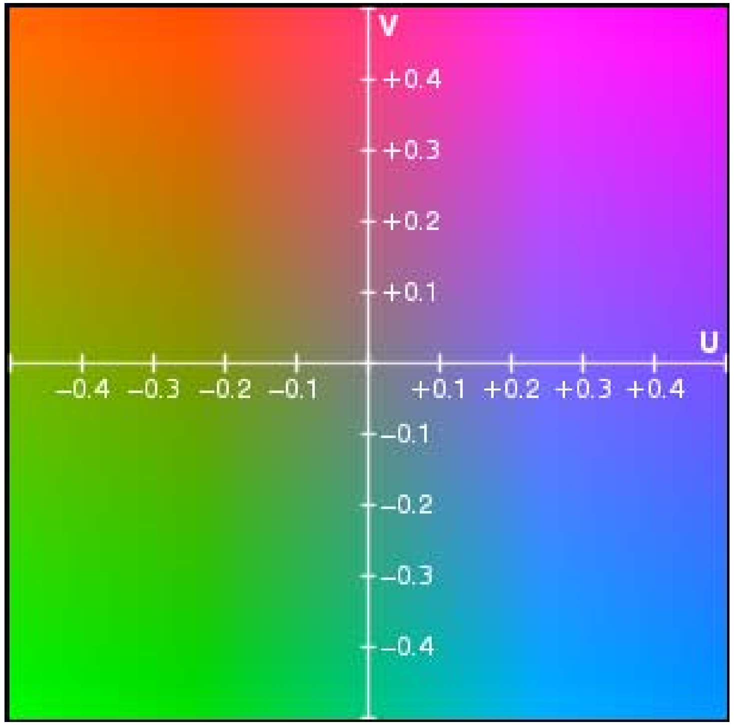
  it measures the acquisition speed. A typical rate is the *frame rate*, 24 fps (frames per second)

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:

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

Interlaced video

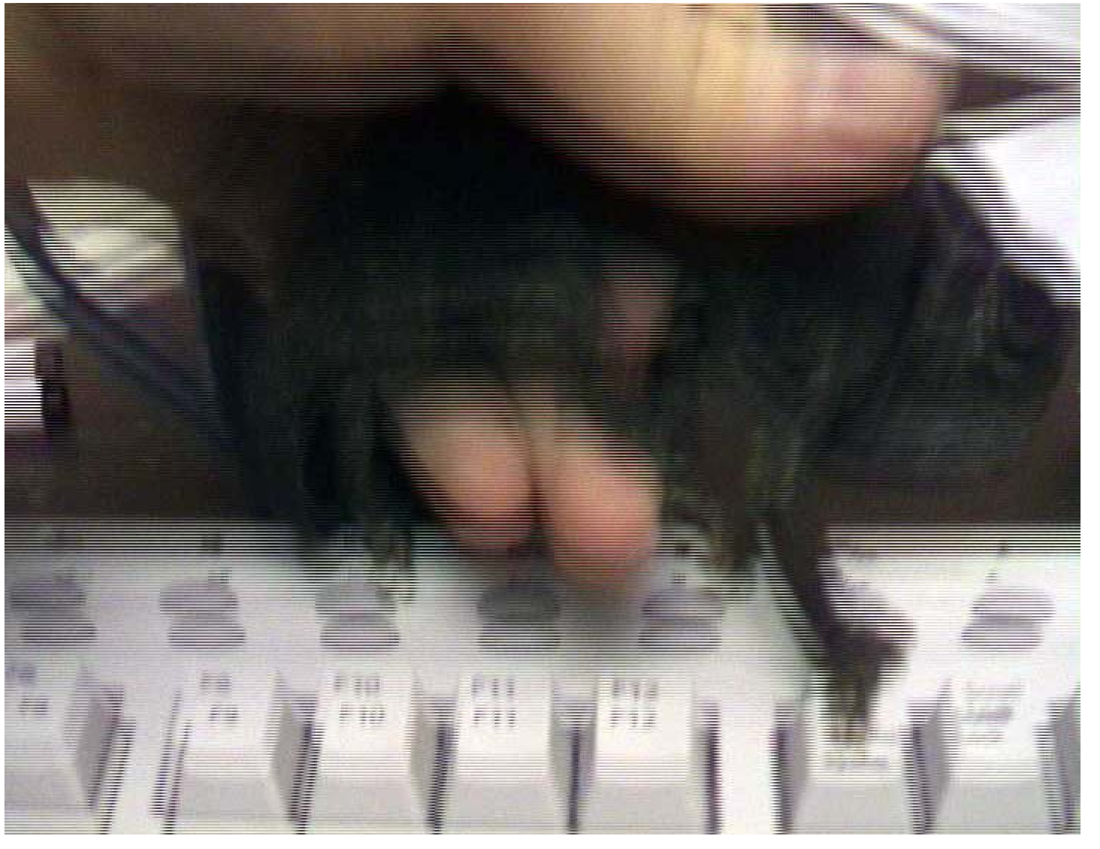
- An interlaced video is composed of *fields* with a smaller vertical resolution than the original video
  - The PAL format, for instance, is 720x288px

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
  - ....

Video-surveillance

- Video-surveillance refers to techniques and methods for inferring information on a scene under observation
  - Observation -> CCTV cameras
  - Information -> object localization and classification, object tracking, behaviour analysis...
- It is a rather general problem!

In the context of video-surveillance...

- More focused applications:
  - **Access control**
  - **Abandoned objects detection**
  - **Anomaly detection**
  - **Traffic monitoring**

Building blocks: change detection

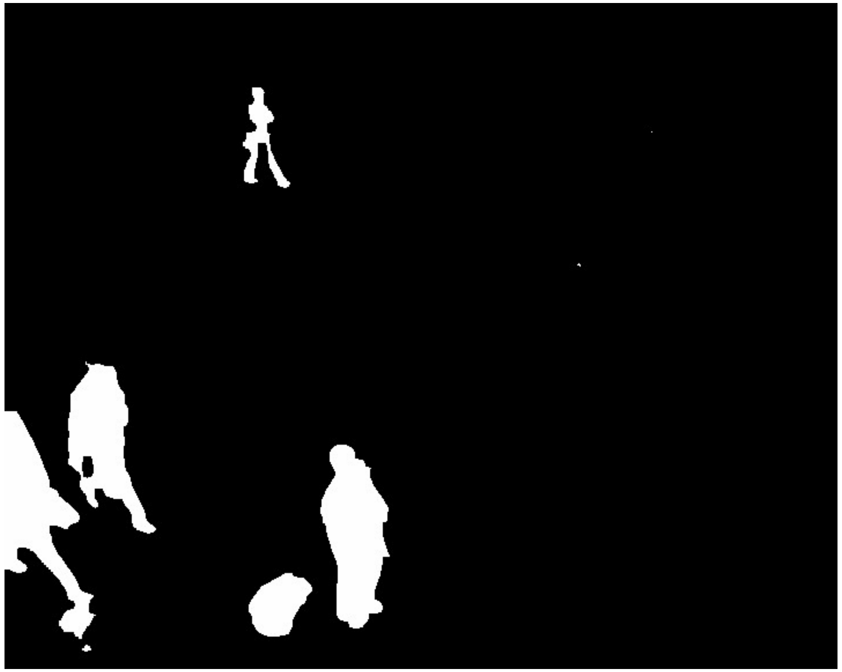
- **Motion segmentation:** localize image regions with a common motion pattern
- If the camera is still motion segmentation is usually referred to as **change detection**
- Change detection is commonly addressed comparing each frame of the sequence with a reference model of the empty scene (the so-called **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**

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

Limits of simple algorithms

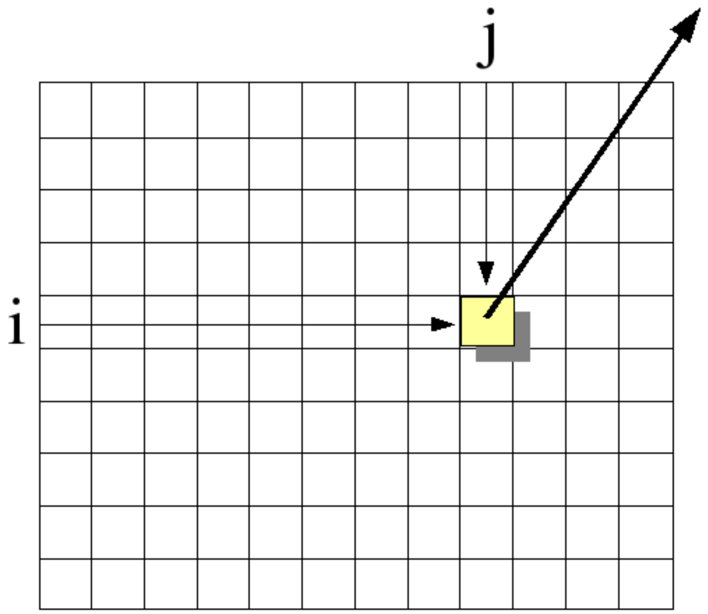
- In the previous methods each pixel of the background is described by a single value, therefore it is not possible to model evolving (e.g., periodic) patterns
- Examples: waving leaves, trembling objects..

Building blocks: background construction

METHOD 3: Codebook model (Kim *et al*, 2005)

- Codebooks allow for more complex pixel models

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

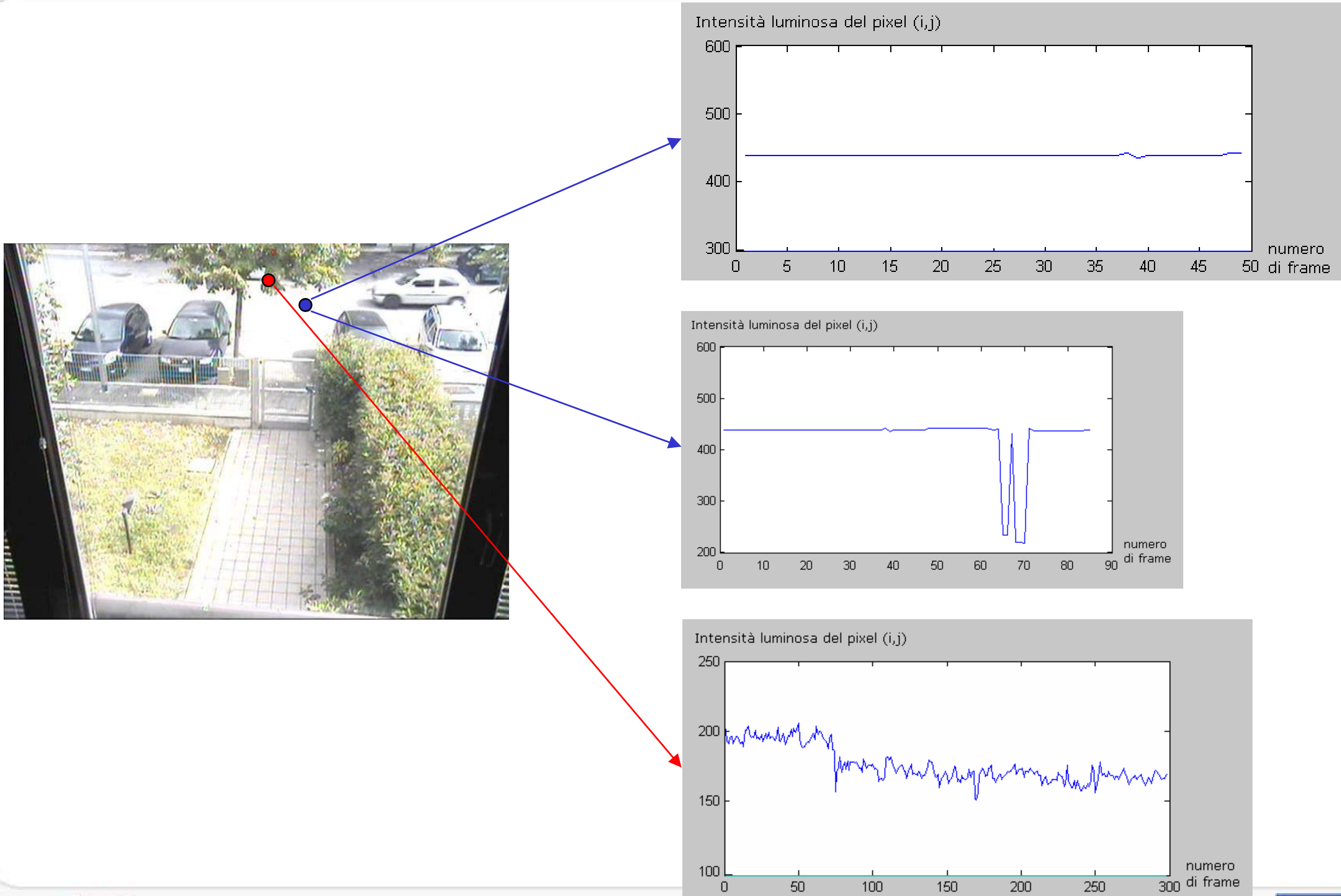


The codebook approach

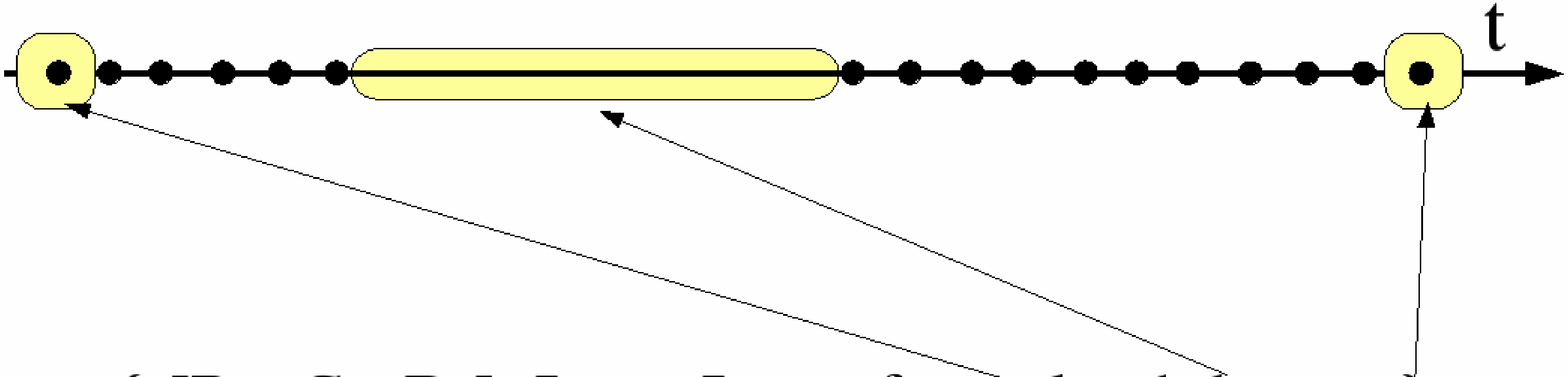
- Each pixel is associated a codebook
- Each codebook is formed by a set of codewords
- A codeword encodes information on the appearance and the dynamics of a pixel

- Abbiamo bisogno di una fase di allenamento **(training)** perché il sistema costruisca il codebook che modella la scena

Codebook model



Codebook model: codewords



$$c_k = \{\ [\underline{R}_k, \underline{G}_k, \underline{B}_k], I_{k,\,min}, I_{k,\,max}, f_k, p_k, \mathrm{lambda}_k, q_k, \}$$

- **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

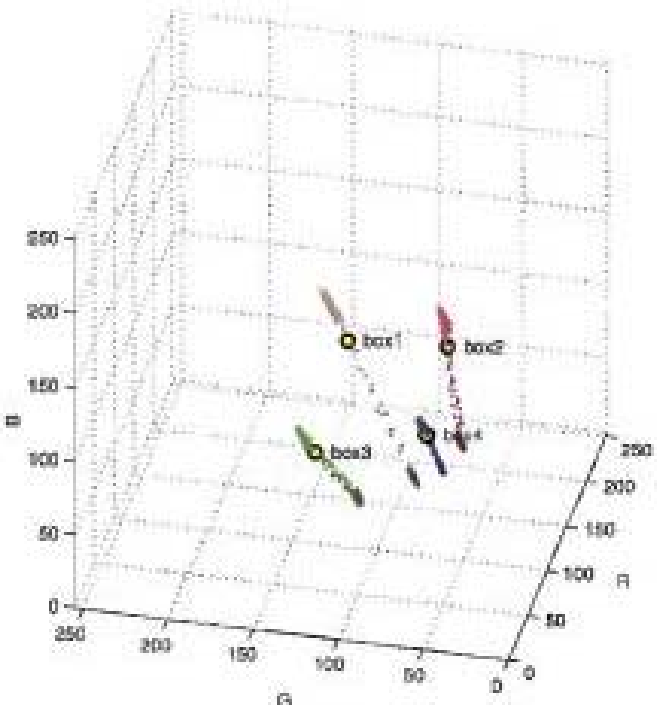
- A training set of N frames is used to initialize the model
- We associate a codebook to each position p=(i,j)
- For each p=(i,j) we have N observations:

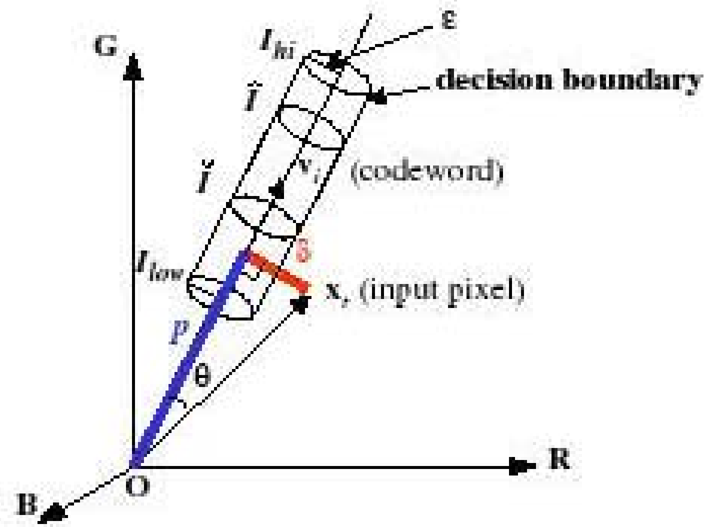
$$\{x_{p,t}=(R_{p,t}\,,G_{p,t}\,,B_{p,t})\}_{t=1,\ldots,N}$$

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

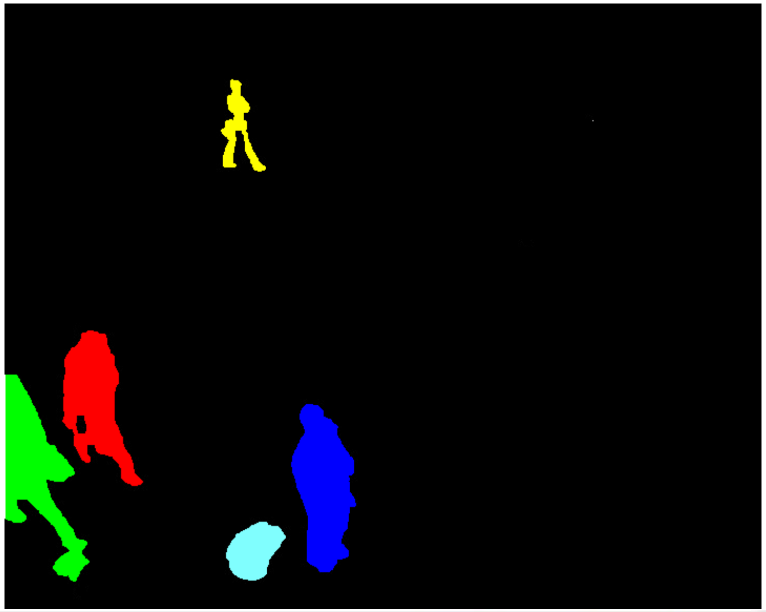




Codebook model: example video

Building blocks: blob tracking

- The binary map obtained from change detection can be described as a set of *connected components* **(blobs)**
- Each blob corresponds to an object moving in the scene

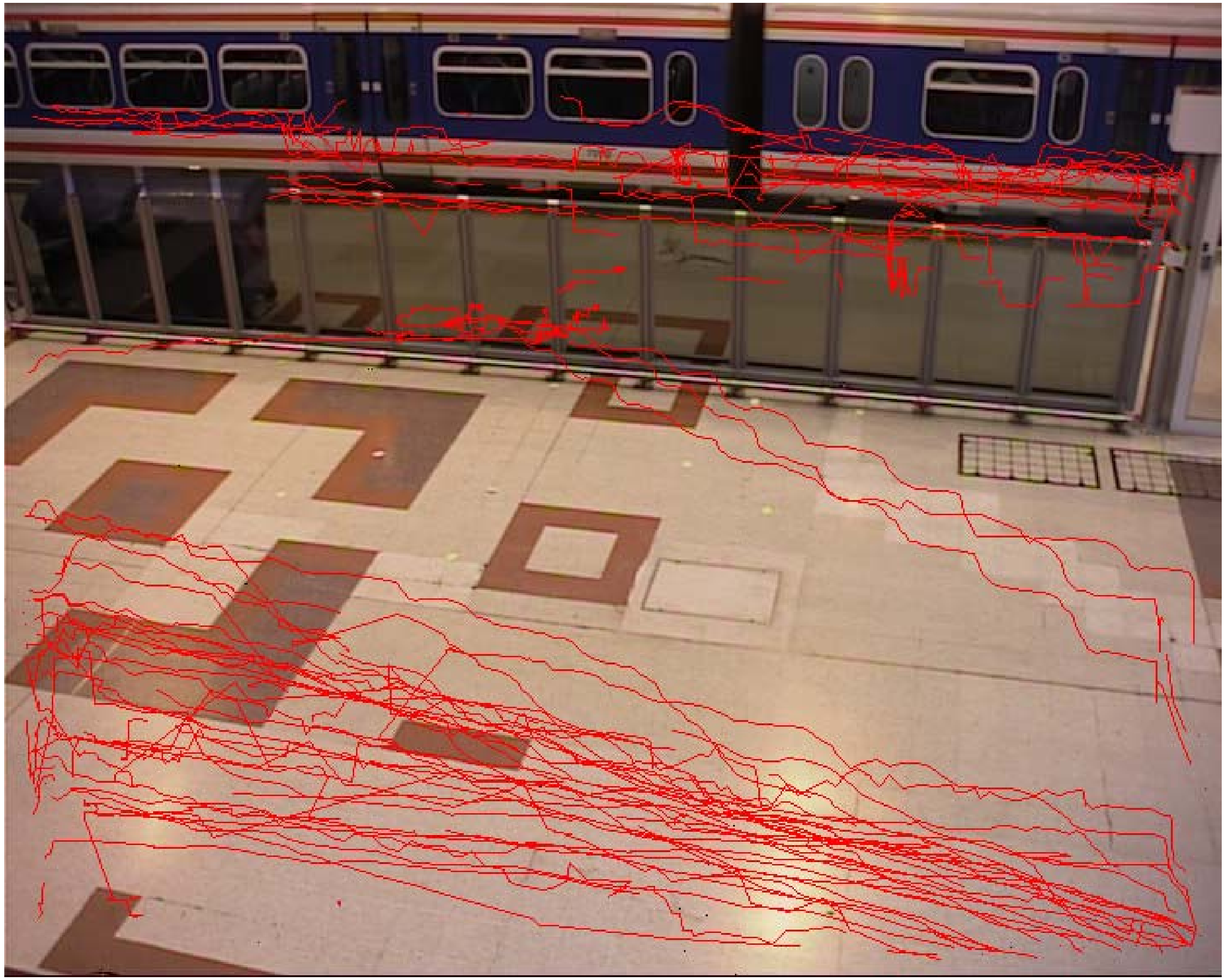


Building blocks: blob tracking

- In order to analyse the dynamics of the moving object it is useful to match blob elements at time t+1 with blob elements observed at time t
- The sequence of matches over the video sequence is usually referred to as **blob tracking**
- Dynamic filters (e.g., Kalman filter) may be applied to smooth the observations and perform predictions

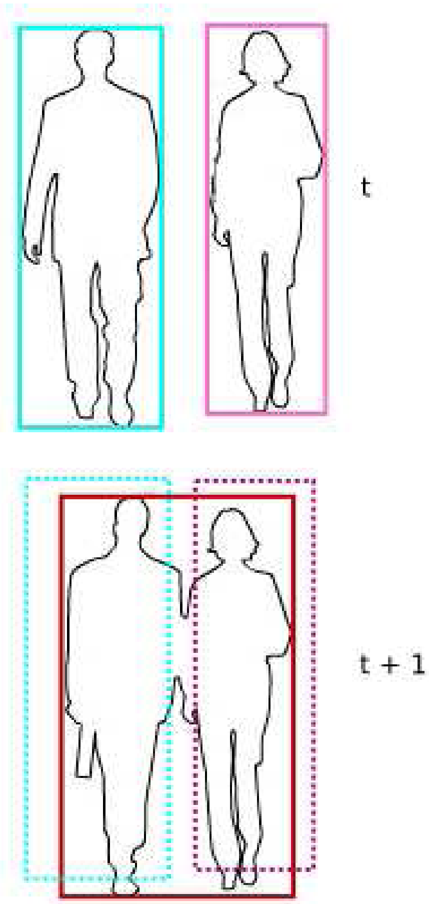
Blob tracking: example video

Blob tracking results

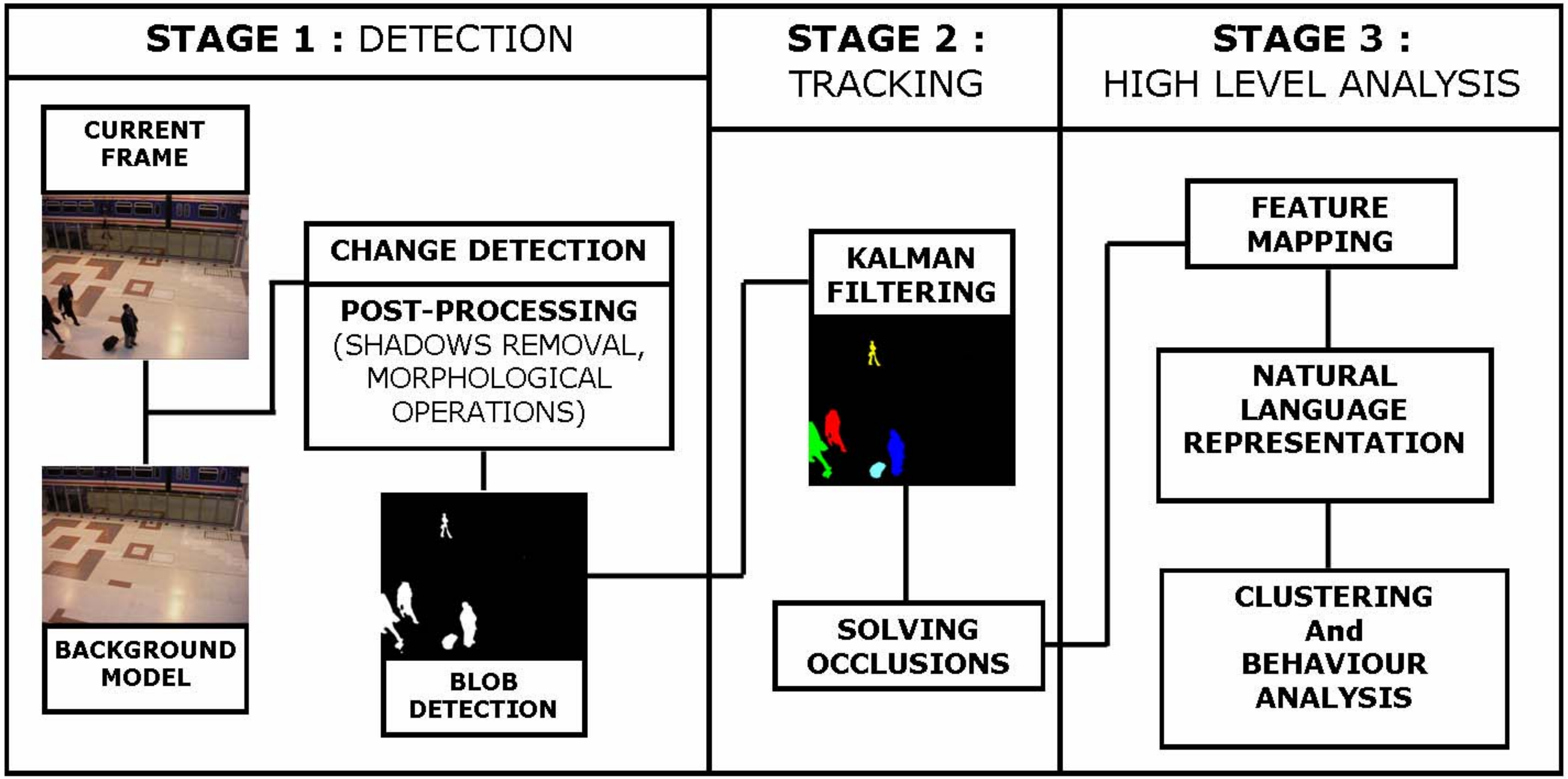


Dealing with occlusions

- If the scene is moderately crowded blob elements may merge
- How to deal with such a problem (data association)?



Current research: unsupervised behaviour analysis



VIT Vision for Innovative Transport

CAPACITIES

Current research: video-based face detection e recognition

A straightforward application: people counting

- Time for questions!