UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Ingegneria
Corso di Laurea Magistrale in Ingegneria
Informatica

Improving WATSS web application with Computer Vision techniques

Visual and Multimedia Recognition

Lorenzo Cioni

Anno Accademico 2015/2016

Introduction

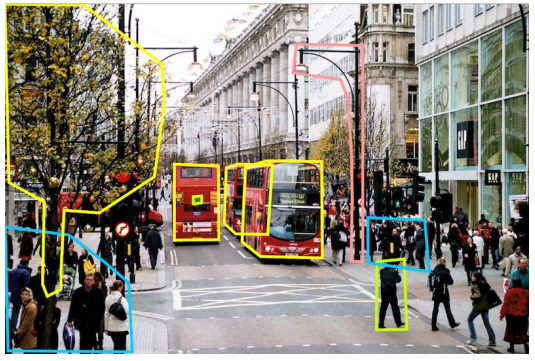
WATSS, **Web Annotation Tool for Surveillance Scenarios**, is a web-based annotation tool developed to annotate dataset in surveillance systems.

Main goal: improve WATSS with some **Computer Vision approaches**, in order to make easy for users to use this tool and make the annotation process more *automatic*

Comparative analysis of annotation tools

LabelMe

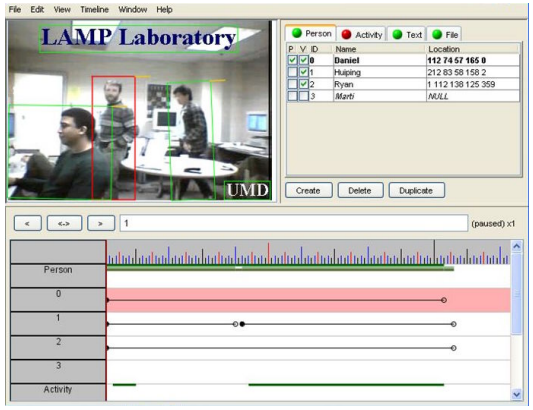
- **Web-based** tool, also for mobile applications
- Annotate scenes with **polygonal areas**
- **Nested** objects and **occlusion** annotation
- *Zoom in* and *out* of the scene



Comparative analysis of annotation tools

ViPER-GT

- **Java application** tool
- Annotate scenes with **geometrical shapes**
- **Timeline** and **annotation highlighting** on time change
- Linear **interpolation** between annotations
- *Zoom in* and *out* of the scene



Comparative analysis of annotation tools

VATIC

- **Online** tool
- Developed for **object detection**
- **Crowd-sourcing** to Amazon's *Mechanical Turk*
- Multiple **plugins**: *object tracking*, *sentence annotation*, etc.

Comparative analysis of annotation tools

WATSS

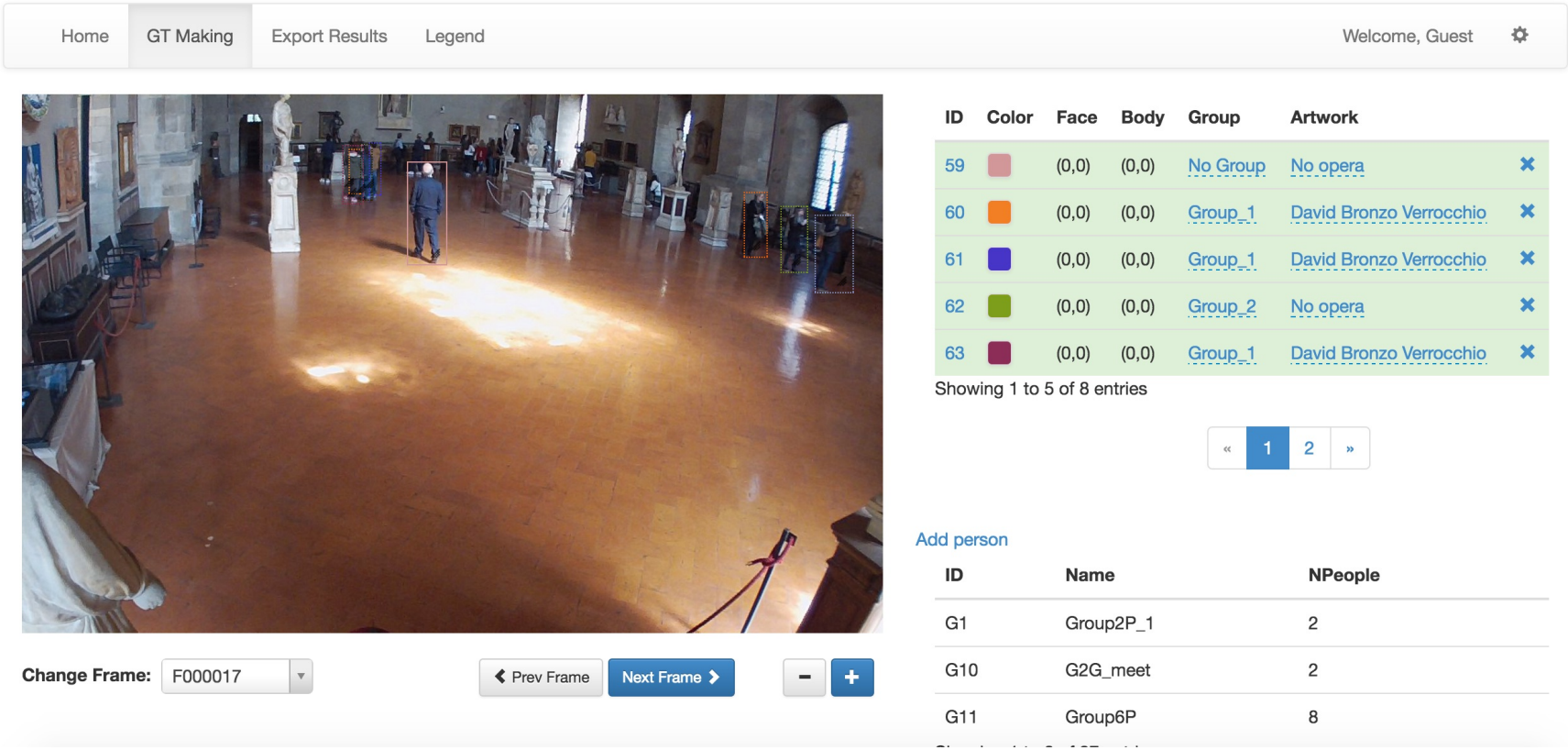
- **Web-based** tool
- Annotation with bounding box
- **Occlusion** area
- Coarse **gaze** estimation
- **Groups** and POI under observation
- **Multiple cameras** manager

Improvements

- **User interface** renovation
- Simpler annotation making and editing
- Video **timeline** for annotations
- Annotation automatic **proposals** generation
- **Scene geometry**-based enhancement
- Easy **setup** process

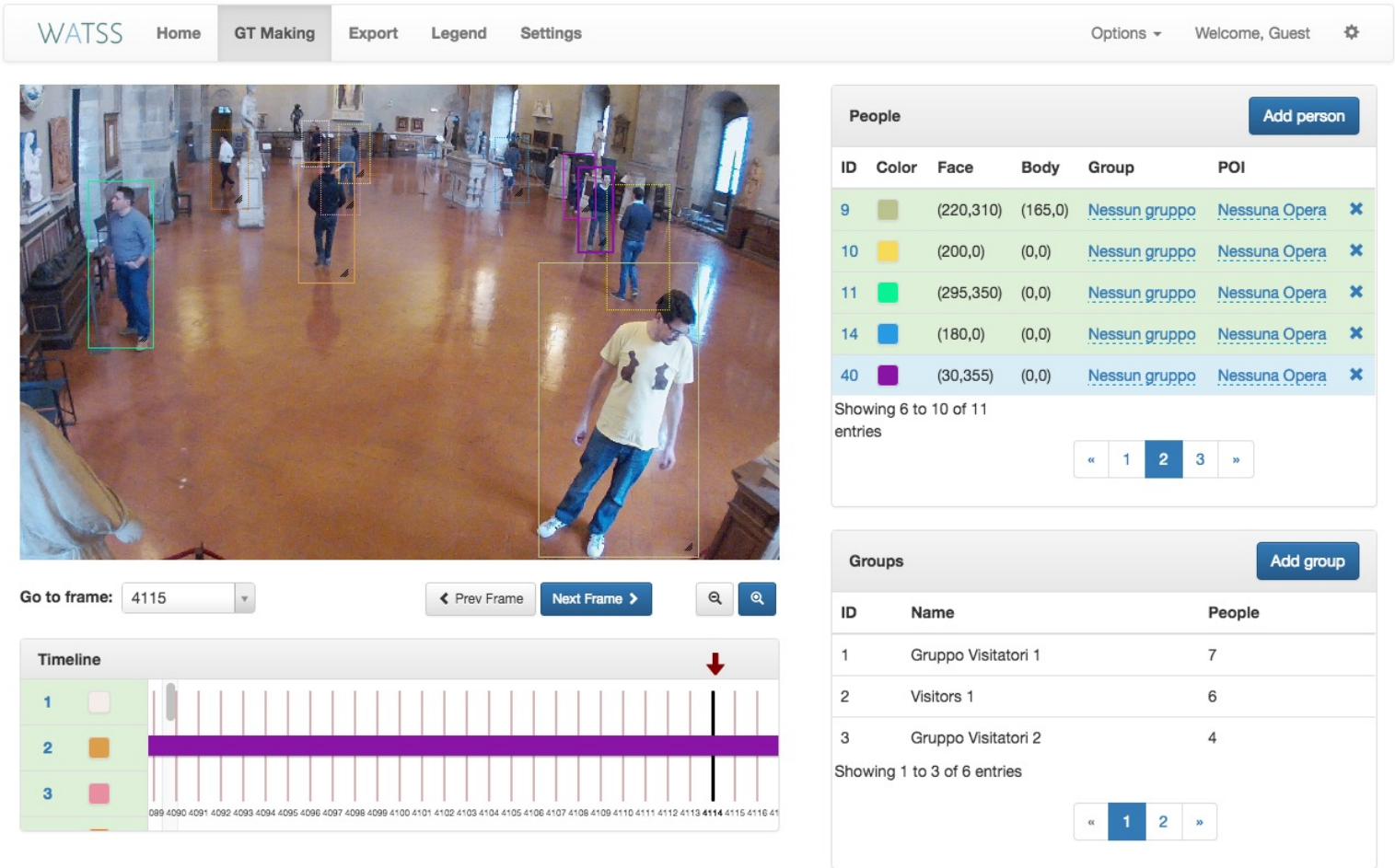
User interface renovation

The **old** WATSS user interface



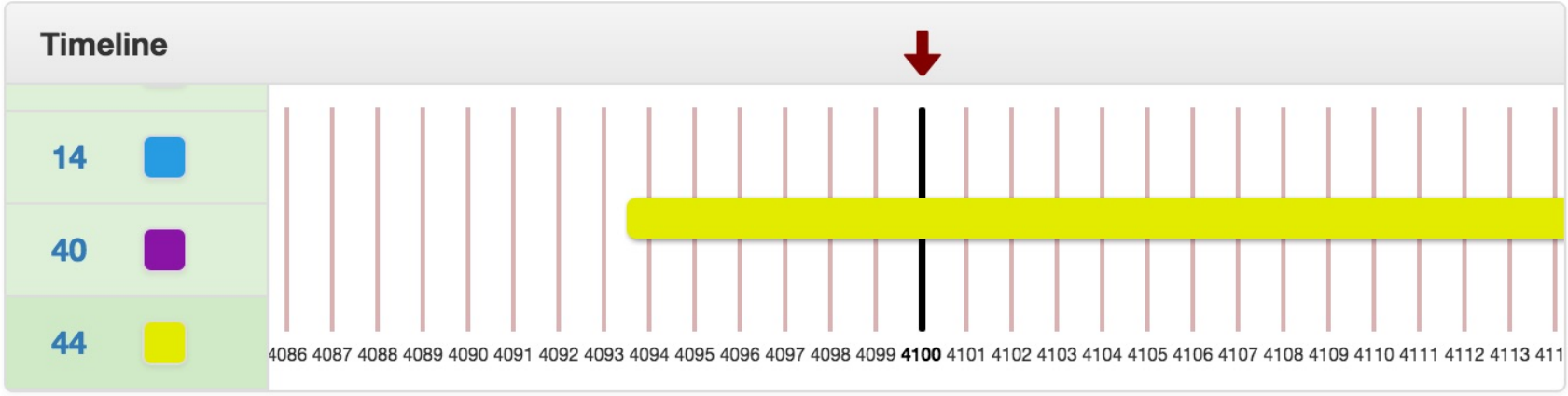
User interface renovation

The **new** WATSS user interface



Video timeline

In the video **timeline** all the video frames are shown, coloring the ones with at least one annotated person.



Selecting a person in the list, the timeline displays its **history** highlighting frames where it is present. It is possible to navigate video frames by clicking on it.

Proposals generation

It is possible to generate **proposals** for a person in some selected frames based on previous annotation of a it using timeline: just click and drag highlighted annotation.

Proposals generation is based on the combination of three different techniques:

- **Motion detection** using a *background substractor*
- **Pedestrian detection** using *HOG descriptors*
- **Kalman filter** for the *motion estimation*

Motion detection

Motion detection is based on a **background substraction** method: moving objects are detected performing a subtraction between the current frame and a background model of the current scene, obtaining a **foreground mask**.

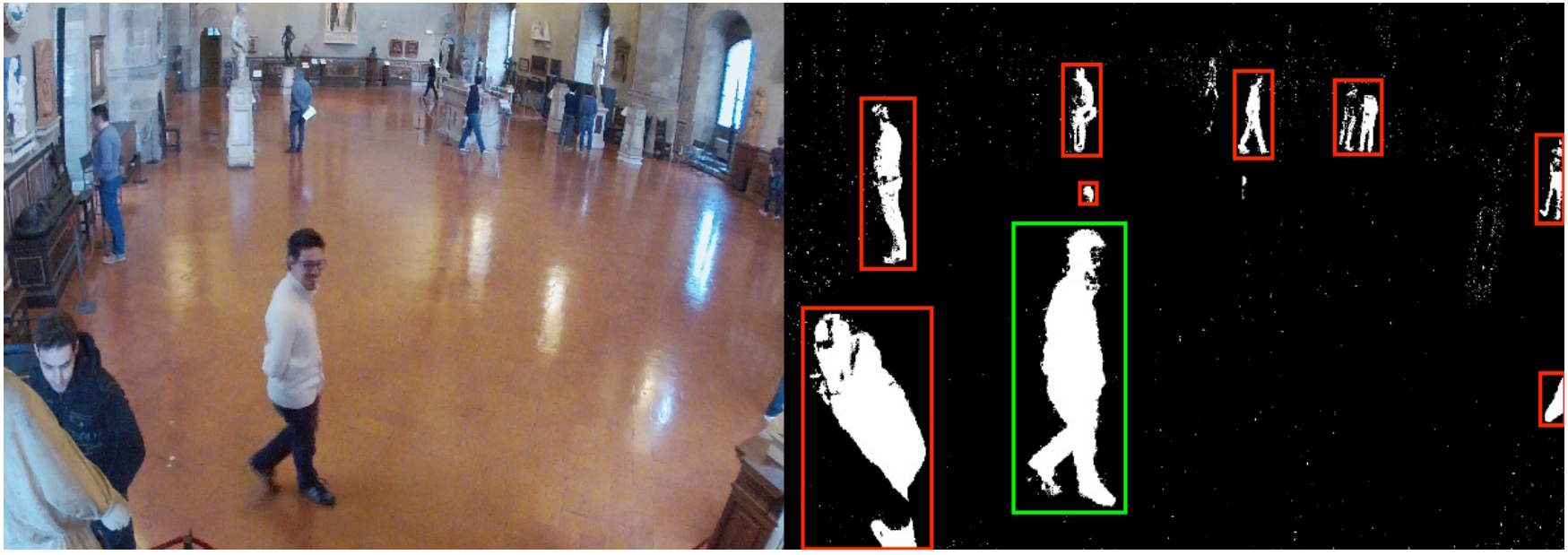
Each pixel of a frame is modeled as a **Mixture of Gaussians** and those which correspond to background colors are selected according to variance and persistence.
Pixel values that do not fit the background distributions are considered part of the foreground.

Background modeling consists of two main steps:

- **Background initialization**: background model evaluation.
- **Background update**: background model is adapted to possible changes in the scene.

Motion detection

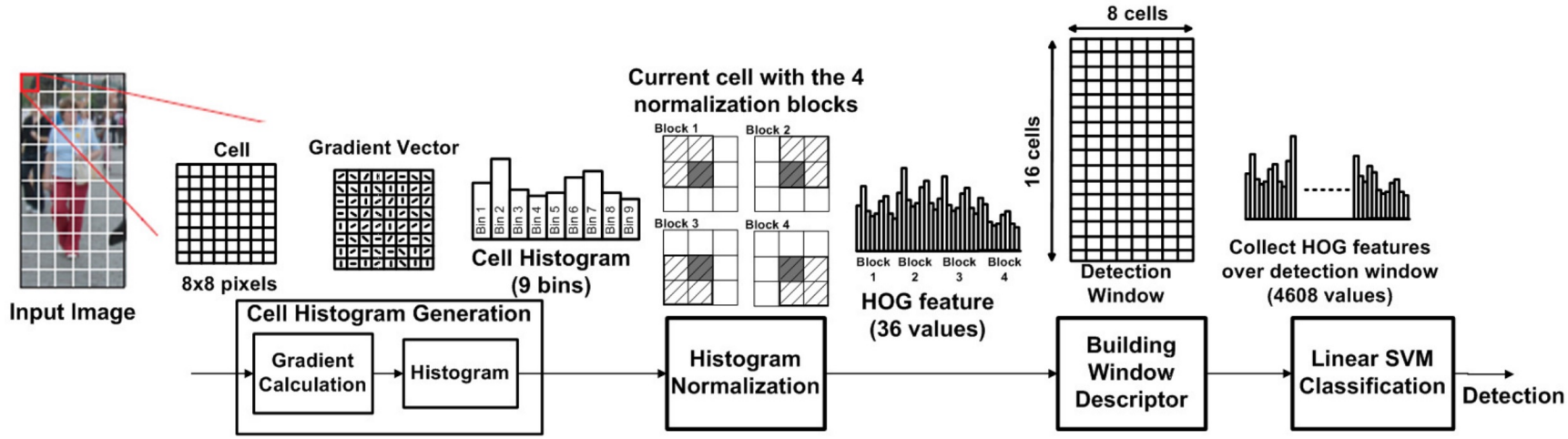
Using the foreground mask, a set of **detections** are extracted based on contours.



Given the previous frame person bounding box, those that do not *overlap* or are *inconsistent* with its dimensions are discarded.

Pedestrian detector

The used *pedestrian detection* technique is based on **Histogram of Oriented Gradients** and SVM classifier.

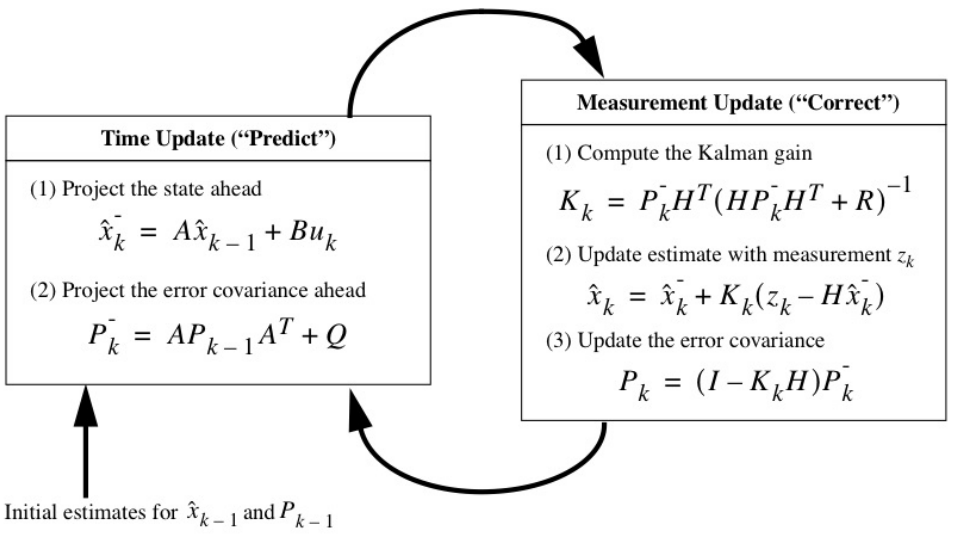


As in the previous case, detections are filtered according to person history in scene.

Figure from Suleiman, A., Sze, V. J Sign Process Syst (2016)

Kalman filter

A **Kalman filter** is an optimal estimator used for following state estimation based on a set of previous observations.



As system state is considered the **coordinates** (x, y) **of the person** in the current scene, using motion and pedestrian detection for updating the measure.

Proposals generation

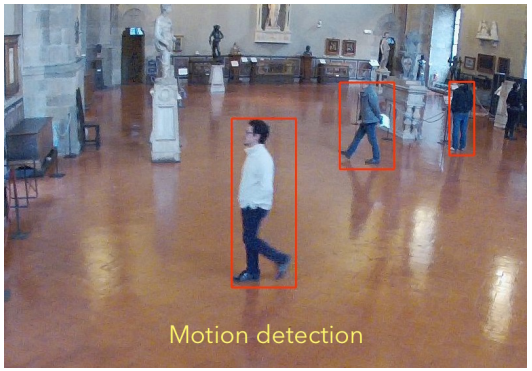
A proposal for a generic frame is the result of the combination of the above descripted methods. each resulted bounding box is compared with the previous frame annotation for evaluating a **score**:

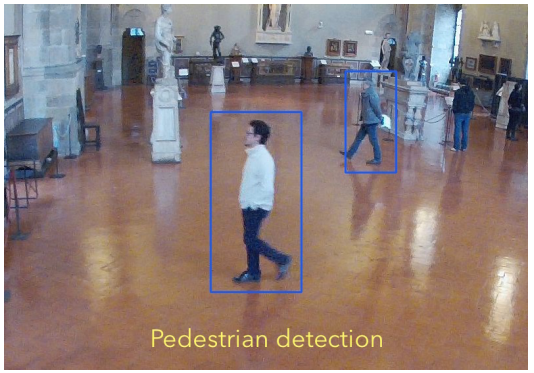
$$score(r) = \frac{intersection(r, p)}{union(r, p)}$$

where r is a resulting bounding box (*i.e. the output of the motion detector*) and p is the bounding box of the previous frame.

If motion or pedestrian detector fails, then the *Kalman filter prediction* is used as proposal.

Proposals generation example





Scene geometry

In the annotation insertion step, it is possible to use **scene geometry** knowledge in order to generate proposal based on the pointer position.

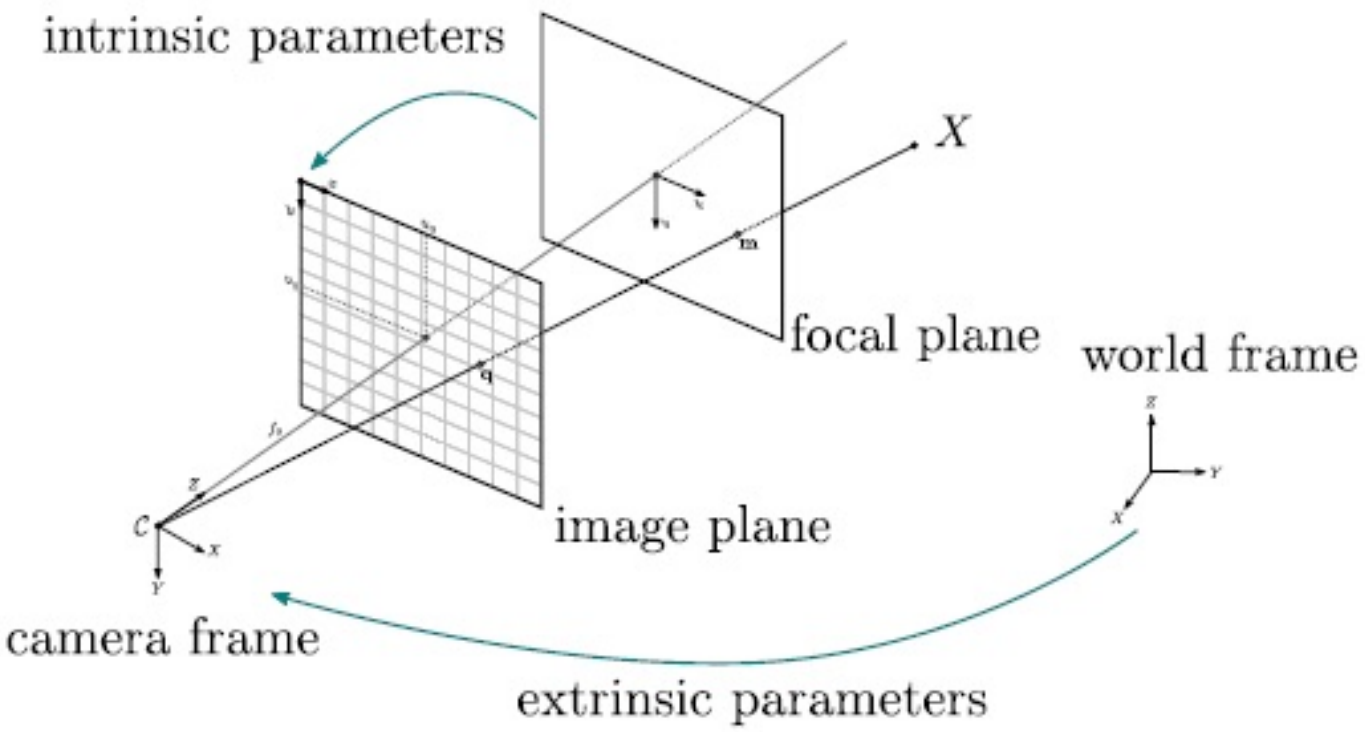
Requirements:

- **Static** cameras: they must be fixed in their positions
- Camera **calibration parameters**
  - *Intrinsic* parameters, **K**
  - *Estrinsic* parameters, rotation **R** and translation **t**
  - A *cross-ratio* μ, being projective invariant

Camera calibration

Given $\mathbf{X} = (X, Y, Z, 1)$, coordinates in **world**, and $\mathbf{x} = (x, y, 1)$, coordinates in **scene**:

$$\mathbf{x} = K[R|t]X$$



Human height estimation

From camera parameters, it is possible to evaluate **vanishing line l** and **vanishing point v**

$$l = P * [0, 0, 1]'$$

$$v = ((K')^{-1} * K^{-1}) * l$$

$$W = I + (\frac{1}{(1-\mu)} - 1) * \frac{v * l'}{v' * l}$$

Giveng head position $head = (x, y, 1)$ and **W**:

$$feet = W^{-1} * head$$

$$height = |head_y - feet_y|$$