Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Video-surveillance methods with low constraints for the analysis of complex scenarios

by

Luca Zini

Theses Series **DIBRIS-TH-2013-06**

DIBRIS, Università di Genova

Via Opera Pia, 13 16145 Genova, Italy `http://www.dibris.unige.it/`

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Dottorato di Ricerca in Informatica

Ph.D. Thesis in Computer Science

Video-surveillance methods with low constraints for the analysis of complex scenarios

by

Luca Zini

February, 2013

Dottorato di Ricerca in Informatica
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
http://www.dibris.unige.it/

Ph.D. Thesis in Computer Science (S.S.D. INF/01)

Submitted by Luca Zini
DIBRIS, Univ. di Genova
luca.zini@unige.it

Date of submission: February 2013

Title: Video-surveillance methods with low constraints
for the analysis of complex scenarios

Advisor: Francesca Odone
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
odone@disi.unige.it

Ext. Reviewers:
Alessandro Bevilacqua
Dipartimento di Informatica Scienza e Ingegneria
Università di Bologna
alessandro.bevilacqua@unibo.it

Andrea Prati
Dipartimento di Progettazione e pianificazione in ambienti complessi
Università Iuav di Venezia
andrea.prati@iuav.it

Abstract

Multi-camera video surveillance is a research domain with several open challenges. In particular, there is a growing interest in studying and designing adaptive methods to process observations gathered by multi-camera systems and to extract high-level information such as behaviours of interest that are either defined a priori or learned on-line.

Modern video surveillance systems are composed of multiple cameras (possibly complemented by other sensor types) to observe wide areas with potentially multiple and redundant views of the objects in the scene. In this context, the coordination of the data gathered and information fusion is crucial to obtain actionable results and to increase the accuracy of the high-level algorithms.

In the literature the high-level analysis of the scene and the cameras coordination task are usually addressed independently. For example, geometrical methods are used to coordinate cameras, while signal processing methods extract features that are fed to machine learning algorithms to extract higher level information. In this thesis, instead, we rely on high-level information for both tasks. Indeed, we study, design, and implement methods for cameras coordination and object matching which exploit apparent behaviour to reduce assumptions on pose, motion, or structure of the scene. Moreover, we address the problem of extracting semantic elements of the observed scene, including the presence of objects of interest and of crowd phenomena.

Our work led to a set of independent modules all characterized by a low computational cost and a moderate number of a priori assumptions and required interaction with a human operator. These modules may also be seen as the initial building blocks of a system for detecting and analysing dynamic events from multiple views.

Table of Contents

Chapter 1

Introduction

1.1 Scope and motivation

Video analysis is a challenging field of computer vision whose final objective is to reproduce the ability of humans of perceiving dynamic events and understanding actions.

The fields of applications are many and range from security [CLK+00] and optimization of resources [GBTTO12] to human-machine interaction [WH99]. Typical tasks in security applications are the detection of potentially dangerous situations: abandoned luggages in a public space, abnormal and suspicious behaviours, or the presence of people in a prohibited area. As for logistics and resource optimization, possible tasks are the estimate of the crowding levels to rationalize public transports or goods displays in departments stores [CHBY06, GBTTO12].

Computer vision solutions are nowadays more appealing thanks to the computational power of modern processors and the availability of low-cost consumer's acquisition devices. More specifically, in the field of video-surveillance, computer vision algorithms brought to the marked smart digital video recorders, able to process video streams from multiple sensors, detect and store dynamic events in real-time and display the detected events on appropriately designed graphical user interfaces. These products are nowadays largely accepted and spread on the video-surveillance market and bring a huge support to human operators monitoring large environments although they have still little generalization ability and intelligence.

Therefore, one of the main goals of video-surveillance research is to widen the spectrum of scenarios that can be handled autonomously by computer vision algorithms, increasing the auto-configuration ability of camera systems and reaching a higher understanding of complex scenes.

Before we reach the objective of having computer vision algorithms able to extract robustly high level information from videos, there are still many open challenges in low and mid-level tasks that need to be solved. Some of those issues are raised by the environment: from the

intrinsic variability of the scene due to clutter or complex backgrounds, to appearance changes caused by point-of-view variations, to unstable lighting conditions. Other issues are due to the sensors themselves, or even to the number and configurations of multi-camera systems. Indeed, a multi-camera system provides more data, allows us to observe a wider area and to cope with scenarios that may be too ambiguous to be handled with a single camera. However it is also a source of complexity as to exploit the information in multiple data streams the cameras need to be coordinated and the information they convey has to be merged at some level. Finally, while traditional surveillance systems generally exploit static or Pan-Tilt-Zoom cameras positioned in a known and controlled setting, we may need to process data from unknown and uncalibrated systems as it may be the case of a remote camera system that cannot be accessed physically. Moreover we may have to work with cameras that move in a non controlled way (e.g, smartphones or mobile security agents) and that are not synchronized with the rest of the system.

Most of the algorithms available in the literature have some constraints w.r.t. the scenario or the acquisition system: common requirements are the presence of calibrated cameras, planar environments, static cameras or the possibility to access the area to configure and train the algorithm. Our objective is to study computer vision techniques to extract quantitative and qualitative information on the presence of objects (people in particular) in complex scenarios from single and multi-camera systems balancing the accuracy of the methods with the constraints imposed on the system, the computational resources needed to run in real time and the need of human interaction. To cope with the large variability of many problems machine learning techniques may be used profitably.

1.2 Synopsis

This thesis can be divided in three parts: in the first one we review low-level tools to model the geometry of single and multi-camera systems (Chapter 2). The second part is about the extraction of qualitative and quantitative mid-level information from the scene, considering motion (Chapter 3), single objects (Chapter 4), crowds (Chapter 5). The third part of the thesis considers the coordination of multiple cameras, specifically studying the synchronization of video streams (Chapter 6) and the association of people between views (Chapter 7). While the latter part of the thesis considers low level tasks, it uses the tools from the previous chapters and high level information on the appearance of the dynamics that are generally associated to high level algorithms for actions recognition and modelling. Figure 1.1 shows a visual scheme of the structure of the thesis.

We endeavoured to make this dissertation self-contained, hence Chapter 2 is about basic camera geometry and overviews both the geometrical models of single and multi-camera systems and the methods used to compute them starting from images.

In Chapter 3 we discuss the most known methods from the literature to detect and to model the

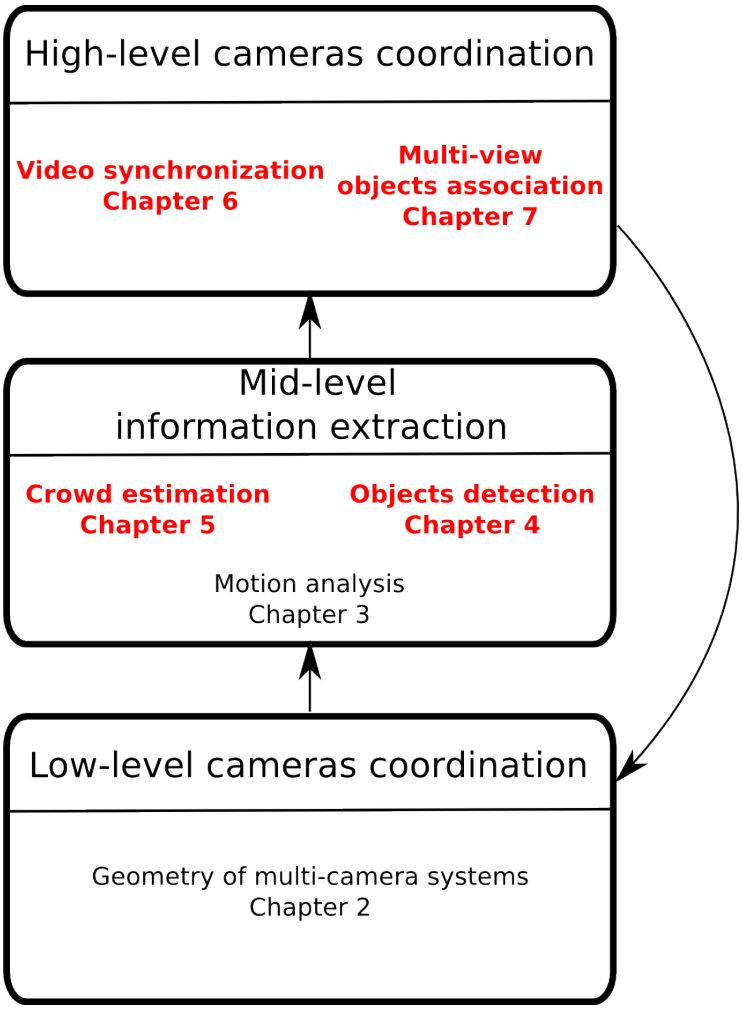


Figure 1.1: Scheme of the topics presented in this thesis. The chapters reported in bold red contain original contributions.

motion of both cameras and world objects. We first introduce the modelling of the motion of a camera from a geometrical viewpoint, then we consider the opposite case, where the camera is assumed to be static and the objects move. Finally we consider the general case where both the camera and the objects move.

Chapter 4 is devoted to object detection, with a particular focus on pedestrians. We present a framework based on regularized statistical learning methods for feature selection [ZO11] with the aim to learn automatically the representation needed to detect a specific class of objects balancing precision and computational requirements. To this end, the study is based on the analysis of the steps of the computation from the description of the image to the final detection of the objects, to add as much information as possible in the image description without slowing down the algorithm.

In Chapter 5 we consider a more complex setting where the number of people may grow and people in the scene may occlude considerably each other. In this case we try to estimate their number rather than detecting their position precisely. People counting can be used as a standalone algorithm in the case we are interested only in the number of people or as an alternative or as a support to people detection in challenging situations, where the crowding level makes too complex to localize each single person. Our contribution to the literature is a method [ZON13] that estimates the number of people framed by a static camera in a planar environment using very low computational resources.

Chapter 6 considers the problem of synchronizing a multi-camera system from video streams assuming this cannot be guaranteed by hardware solutions. This may be the case of remote cameras that suffer delays and connections losses or of videos acquired independently and processed offline. To work in this context we have proposed a constraint-free method [ZCO13]: the approach is based on the analysis of the actions of each actor in the scene and, thanks to such a high level approach, does not depend directly on low level assumptions as the motion of cameras or the planarity of the ground.

In Chapter 7 we present our study on the association of people between multiple views. Our contribution to the literature is an approach [ZOCed] based on the high level analysis of the actions of people that adopts a similar approach to the alignment method proposed in Chapter 6 and shares with it the same loose requirements.

Finally Appendix A overviews the datasets that have been used to evaluate the algorithms described and provides details on the data acquired specifically to test the methods presented across the thesis.

Chapter 2

Geometry of multi-camera systems

The objective of this chapter is to provide an overview of the geometrical models of single and multi-camera systems. Sec. 2.1 introduces the chapter, Sec. 2.2 describes the geometry of a single camera and Sec. 2.3 the relations between the pixels of a pair of cameras. Finally in Sec. 2.5 it is given an overview of the tools that can be employed to compute the models presented

2.1 Introduction

In this chapter we present the models of single and multi-view geometry.

Geometrical models relate the pixels of the images and the observed world objects. Starting from the model (e.g. positions and orientations of cameras, structure of the sensors) we can relate pixels and light rays in the world or, starting from the information on correspondences between pixels and world objects, we can extract the geometry of system.

The task of estimating the camera model is called *calibration* and involves the computation of the relation between the pixels on the image and the world points.

The applications of geometry in computer vision are many and range from the computation of correspondences between points, to the 3D reconstruction of objects from images, to the estimation of camera motion. In this chapter we will introduce the basic tools that are generally used in video surveillance application. A more detailed reference is [HZ00].

2.2 Single camera model

In this section we introduce a model of the process of the image formation with the objective of extracting the relations between the parameters of the camera, the objects in the scene and the pixels in the final image.

The parameters of the camera system can be classified in three different groups that depend on:

- Sensor.
- Lens.
- Position of the camera.

the first two are modelled by the *intrinsic parameters*, while the position and the orientation w.r.t. a scene reference frame are the *extrinsic parameters*.

The projection of a 3D homogeneous point $\boldsymbol{p_w} \in \mathbb{R}^{4\times1}$ in the world reference frame to its corresponding position in the image $\boldsymbol{p_i} \in \mathbb{R}^{3\times1}$ in the reference frame of the image pixels (see Figure 2.1) is modelled by the linear equation

$$\boldsymbol{p_i} = KM\boldsymbol{p_w} \tag{2.1}$$

where M is the matrix of the extrinsic parameters that encodes rotation and translation of the camera w.r.t. the world reference frame and K is the matrix of the intrinsic parameters that converts from millimetres to pixels.

The matrix P that computes the projection of world points on a camera is the combination of intrinsic and extrinsic parameters and can be defined from Eq 2.1 as:

$$P = KM. \tag{2.2}$$

The most simple case of P is generated by a pin hole camera positioned in correspondence of the world reference frame with the *optical axis* parallel to the z axis (i.e. looking toward the z axis). In this case we have

$$M = [I|\mathbf{0}] \tag{2.3}$$

where I is a 3×3 identity matrix and $\mathbf{0}$ is a null column vector of size 3×1.

If we assume a pinhole camera model, the matrix of the intrinsic parameters is:

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

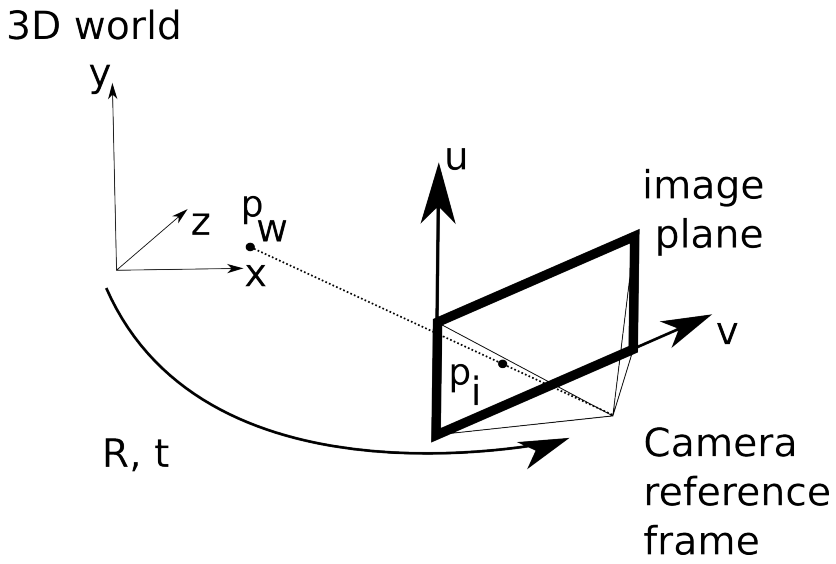


Figure 2.1: Sketch of a single camera system: the calibration parameters allow to map points in the world reference frame to image pixels.

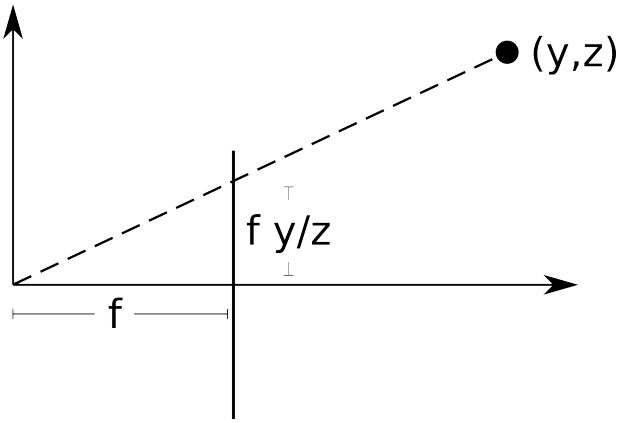


Figure 2.2: Scheme of the projection of a point (y, z) on the image plane using a pinhole camera model.

where f is the z coordinate of the image plane (focal plane, see Figure 2.2).

If we take into account the geometry of the image sensor, the matrix K needs to be modified as:

$$K = \begin{pmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

where f_x and f_y are the focal distance expressed using as unit respectively the vertical and horizontal sizes of pixels, α is the skew parameter (usually it is assumed to be 0) and (c_x, c_y) are the coordinates in pixels of the *principal point* (the intersection between the optical axis and the image plane).

To consider the general case with the camera in an arbitrary position and orientation in the world (see Figure 2.1) we define M as:

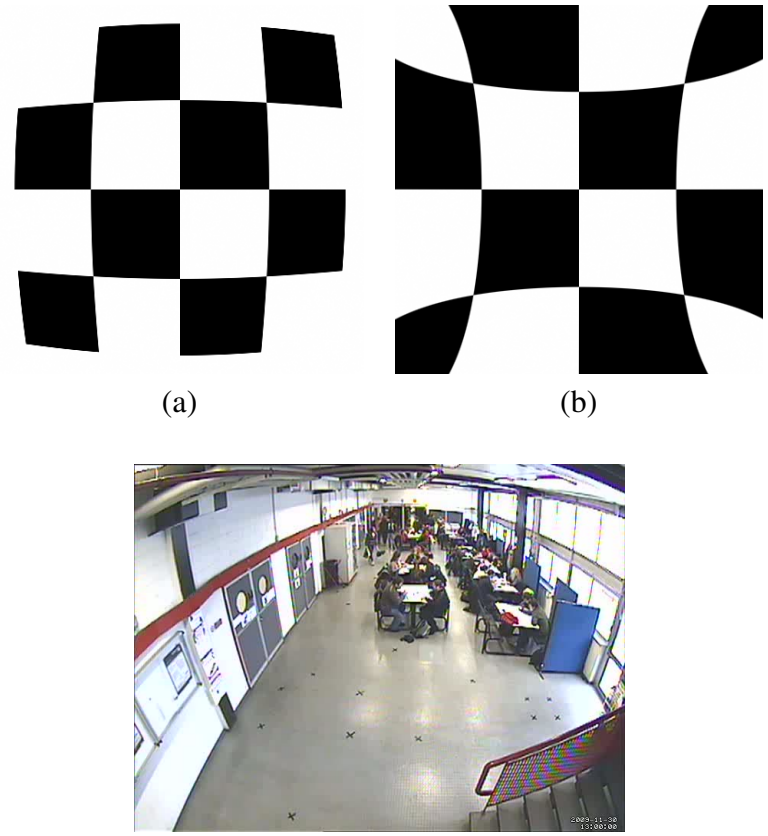
$$M_E = (R| - R\boldsymbol{t}) \tag{2.6}$$

where $R \in \mathbb{R}^{3\times3}$ is the orthogonal rotation matrix applied to the camera w.r.t. the world reference frame and $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ is the position of the camera in the world.

The linear model of Eq. 2.1 does not consider the distortion that is usually induced by the lens elements. The most common is *radial distortion* (see Figure 2.3) that can be modelled and corrected using a polynomial equation with only odd terms [Fai75].

$$\hat{x} = x + (x - c_x^I)\left(\sum_{i=1}^{N} d_i r^{2i}\right) \tag{2.7}$$

$$\hat{y} = y + (y - c_y^I)\left(\sum_{i=1}^{N} d_i r^{2i}\right) \tag{2.8}$$



(a) (b)

(c)

Figure 2.3: Radial distortion: (a) barrel distortion; (b) pincushion distortion; (c) an image distorted by a wide angle lens.

where (c_x^I, c_y^I) is the image center and r is the distance of the point (x, y) from it:

$$r = \sqrt{(c - c_x^I)^2 + (y - c_y^I)^2}. \tag{2.9}$$

Note that the values of the image center (c_x, c_y) in the matrix K and the image center needed to compensate for the radial distortion (c_x^I, c_y^I) in general differ.

Generally only terms up to the fifth degree ($N = 2$) are considered as higher terms does not add precision [WCH92] and make the estimation of the parameters and the solution of the polynomial equation too complex.

Depending on the values of the parameters d_i we may have *barrel distortion* where the angles of the image are pushed toward the center of the image or *pincushion distortion* where they are stretched. An example of barrel distortion from a wide angle lens of a video surveillance camera is shown in Figure 2.3.

More complex distortion models can be extended to take into account the *decentering distortion* of not aligned lenses [Bro71], however it is not commonly required.

Camera matrix decomposition

In case we have the projection matrix P associated with a camera, we may be interested to revert the process of Eq. 2.1 to retrieve information on the system.

The position of the camera can be easily computed as it is the null-space of the matrix P since, combining Eq. 2.1 with 2.6, we have

$$P\boldsymbol{t} = 0 \tag{2.10}$$

and the rank of matrix is 3.

Given the columns of P

$$P = [P_1 P_2 P_3 P_4] \tag{2.11}$$

the rotation matrix and the matrix of the intrinsic parameters can be computed by decomposing the sub-matrix

$$P_{1..3} = [P_1 P_2 P_3] \tag{2.12}$$

that in Eq. 2.6 is defined as:

$$P_{1..3} = KR. \tag{2.13}$$

The result can be achieved using a RQ decomposition that decomposes the matrix in an upper triangular matrix (R) (i.e. the matrix of the intrinsic parameters) and in an orthogonal matrix (Q) (i.e. the rotation matrix). The sign of the solution has to be chosen so that the entries of K related to the physical size of the sensor (i.e. the values of the diagonal) are positive.

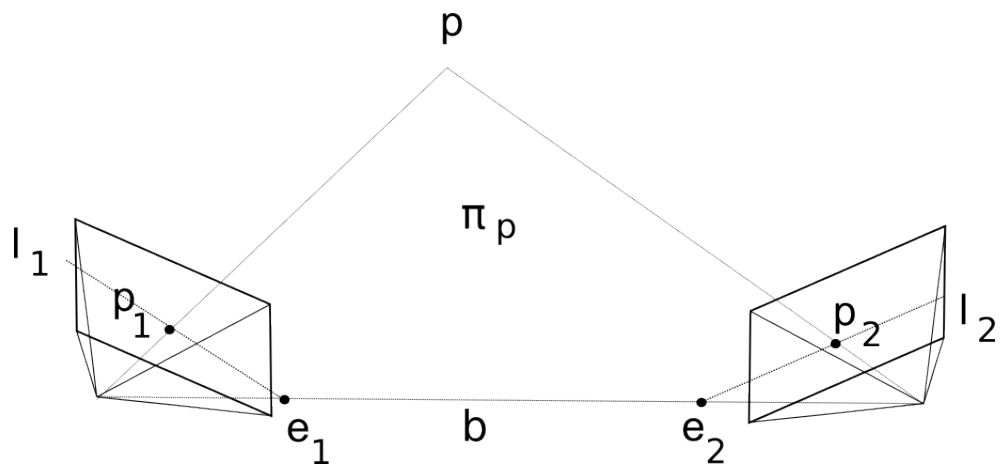


Figure 2.4: Scheme of the epiolar geometry of two cameras. The *baseline* is the line joining the two cameras and its intersections with the image planes are the epipoles $\boldsymbol{e_1}$, $\boldsymbol{e_2}$. Given a world point, the epipolar plane $\boldsymbol{\pi_p}$ is the one that includes the baseline and the point itself.

2.3 Two cameras model

If we consider two or more cameras, usually we are interested in extracting their relative position in the world and the relation between their pixels.

In this case we may not be interested in a fixed world reference frame, and with no loss of generality we can consider the reference frame of one of the two cameras as the world reference frame.

Both the relative pose of the cameras and the mapping between their pixels are encoded in the *epipolar geometry* of the cameras system.

In Fig 2.4 it is shown the general scheme of two cameras: the epipoles $\boldsymbol{e_1}$, $\boldsymbol{e_2}$ are the projections of a camera on the image plane of the other one and are joint by the baseline $\boldsymbol{b}$. Given a point $\boldsymbol{p}$ the epipolar plane π_p is the plane passing through the baseline and $\boldsymbol{p}$.

The pixels of two cameras are related by the *Fundamental matrix* (F). $F \in \mathbb{R}^{3\times3}$ is defined such that the projections $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ on the two cameras of the same world point respect the equation:

$$\boldsymbol{p_1}^T F \boldsymbol{p_2} = 0. \tag{2.14}$$

The previous equation encodes all the properties of the camera system and allows to compute the line on the image of one camera where we expect to find the correspondence of a point of the other camera:

$$\boldsymbol{l_1} = F\boldsymbol{p_2} \tag{2.15}$$

$$\boldsymbol{l_2} = F^T\boldsymbol{p_1}. \tag{2.16}$$

The position on the line depends on the depth of the point along the ray of its projection. The lines $\boldsymbol{l_1}$ and $\boldsymbol{l_2}$ can be geometrically defined as the intersections between the image planes and the epipolar plane associated with the considered point (see Figure 2.4).

If we fix the world reference frame on the first camera we can encode the two projection matrices P as:

$$P_1 = K_1[I|\mathbf{0}] \tag{2.17}$$
$$P_2 = K_2[R|\boldsymbol{t}] \tag{2.18}$$

where K_1 and K_2 are respectively the matrix of the intrinsic parameters of the first and the second camera and $R \in \mathbb{R}^{3\times3}$ and $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ encode the rotation and the translation between the two cameras.

In the previous setting the fundamental matrix can be computed in close form as

$$F = K_2^{-T}[\boldsymbol{t}]_X R K_1^{-1}. \tag{2.19}$$

The matrix F has rank 2 as the size of the nullspace is 1 ($F\boldsymbol{e} = 0$), hence, due to the constraint

$$det(F) = 0 \tag{2.20}$$

its 9 entries have 7 degrees of freedom (due to the constraint 2.20 and a scale ambiguity).

If the matrices of the intrinsic parameters are available it is possible to compute the *Essential matrix* (E)

$$E = K_2^T F K_1. \tag{2.21}$$

The essential matrix encodes the information on rotation and translation between the two cameras. Introducing Eq. 2.21 into Eq. 2.19 we have:

$$E = [\boldsymbol{t}]_X R. \tag{2.22}$$

The 9 entries of E have 5 degrees of freedom due to rotation and translation (6 minus a scale factor), the constraints are:

$$det(E) = 0 \tag{2.23}$$
$$2EE^TE - trace(EE^T)E = 0 \tag{2.24}$$

the first is shared with F and ensures that the matrix has rank 2, the second one is the *cubic trace constraint* and enforces that the two non null singular values are equal.

2.4 Planar configurations

In the case we consider two planes π_1 and π_2, it is possible to map each point of π_1 to the corresponding one of π_2 preserving their collinearity. Such transformation is an *homography* $H \in \mathbb{R}^{3\times3}$ such that:

$$\boldsymbol{p} \times H\boldsymbol{p'} = 0 \tag{2.25}$$

for all the points $p' \in \pi_1$ and $p \in \pi_2$.

The homography between two cameras framing a common plane can be defined starting from the position of the plane, the pose and the intrinsic parameters of the camera:

$$H \propto K_2(R - \boldsymbol{t}\boldsymbol{n}^T/d)K_1^-1 \tag{2.26}$$

where R and $\boldsymbol{t}$ encode the rotation and translation between the cameras, K_1 and K_2 are the matrices of intrinsic parameters of Eq. 2.17 and Eq. 2.18 and $\boldsymbol{n}$, d define the plane $\phi = (\boldsymbol{n}, d)$.

If the matrices K_1 and K_2 are known, the process from Eq. 2.26 can be reverted to extract two possible decompositions that are numerically and geometrically consistent. More details and an overview of the methods available in given in [MV07].

2.4.1 Epipolar geometry and planes

In case it is available only a plane, it is not possible to compute the fundamental matrix, as it is not possible to fix all its degrees of freedom. Any solution in the form

$$F = SH \tag{2.27}$$

where H is the homography between the two views and S a skew symmetric matrix, is a valid solution for the points on the plane.

This can be easily proved, as for any point $\boldsymbol{p}$ and skew symmetric matrix S holds the equation

$$\boldsymbol{p}^T S\boldsymbol{p} = 0 \tag{2.28}$$

that is equivalent to Eq. 2.14 for all the points on the plane:

$$\boldsymbol{p}^T F\boldsymbol{p'} = \boldsymbol{p}^T SH\boldsymbol{p} = \boldsymbol{p}^T S\boldsymbol{p} = 0. \tag{2.29}$$

2.4.2 Quasi-planar world

While observing a plane it is not possible to compute an unique fundamental matrix, if the scene contains a plane and a set of points out of the plane (see Figure 2.5), it is possible to exploit the structure of the data to reduce the number of points needed to compute it.

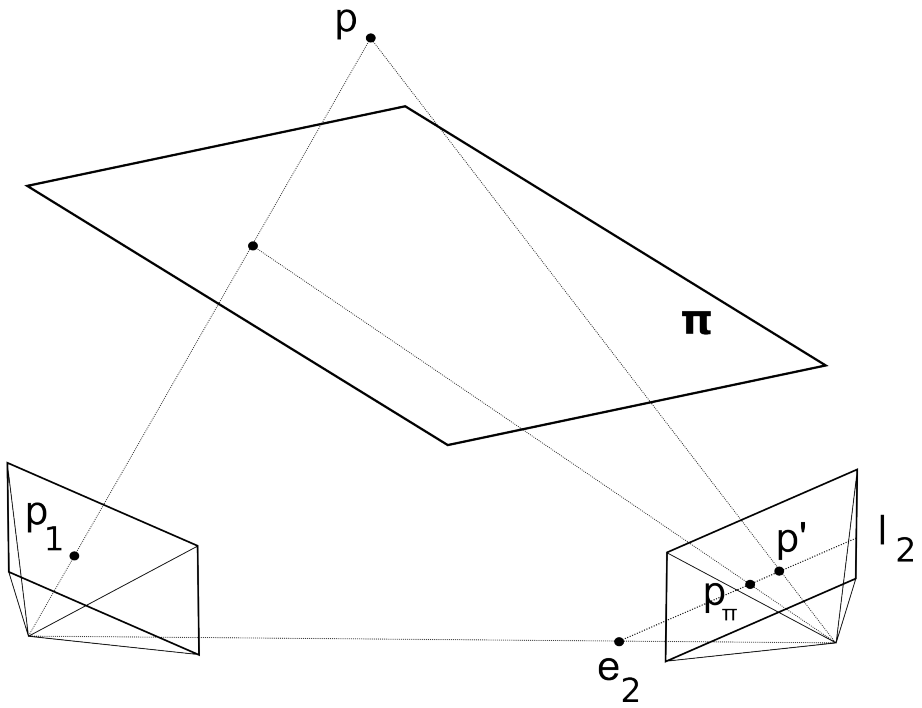


Figure 2.5: Scheme of the epipolar geometry of two cameras that observe a plane and a point $\boldsymbol{p}$ out of it. The virtual parallax induced by the plane can be exploited to estimate F with a lower number of points.

Eq. 2.27 can be completed computing the 3 values of S. It can be shown that the solution is $S = [e]_X$, it follows that the matrix F can be computed as:

$$F = [\boldsymbol{e}]_X H. \tag{2.30}$$

In this case the solution can be computed using only 6 points rather than the 7 needed to compute a general fundamental matrix: 4 planar points are required to compute the homography and 2 out of the plane to compute the epipole.

2.5 Models estimation

In this section we provide an overview of the tools to compute the relations presented in the previous sections. A more detailed overview can be found in [HZ00].

2.5.1 Single camera calibration

The simplest method to compute the intrinsic and the extrinsic parameters of a camera, known as Direct Linear Transformation (DLT [AAK71]), is to solve the associated linear system using a set of points with known coordinates in the world by minimizing the algebraic error. The implicit assumption in this method is that the radial distortion is null and that each entry of the relation is independent from the others.

For each correspondence between a point $(X_w, Y_w, Z_w, 1)$ in the world and the point $(x, y, 1)$ in the image we can write the equations:

$$x = \frac{p_{11}X_w + p_{12}Y_W + p_{13}Z_w + p_{14}}{p_{31}X_w + p_{32}Y_W + p_{33}Z_w + p_{34}} \tag{2.31}$$

$$y = \frac{p_{21}X_w + p_{22}Y_W + p_{23}Z_w + p_{24}}{p_{31}X_w + p_{32}Y_W + p_{33}Z_w + p_{34}} \tag{2.32}$$

that can be encoded in a linear system of the form

$$A\boldsymbol{p} = 0 \tag{2.33}$$

where $\boldsymbol{p}$ is the vector with all the entries of the matrix P stacked in a column and A the matrix containing in each couple of lines the equations derived from Eq. 2.31 and Eq. 2.32:

$$A = \begin{pmatrix} . & . & . & . & . & . & . & . & . & . & . & . \\ X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -xX & -xY & -xZ & -x \\ . & . & . & . & . & . & . & . & . & . & . & . \end{pmatrix}. \tag{2.34}$$

To obtain high precision it is usually needed to collect as many correspondences between pixels and coordinates in the world as possible. The presence of more equations than unknowns usually leads to an overdetermined system due to the effect of noise on the measurements. The most common procedure is to assume that the noise distribution of the measurements is Gaussian: in this case the optimal solution can be obtained with the least squares, whose result is the right null vector of the SVD of A.

The main limitations of this method are three:

- It does not consider the distortion.
- The structure of P is not exploited.
- It may fail in presence of non Gaussian noise.

The most common solution for the first issue is to combine the method with a non linear refinement to consider lens distortion. The second point is due to the procedure where each entry of the matrix P is computed not considering the dependence between the values (e.g. the matrix R has 9 entries but can be determined using only three angles), obtaining a system with more unknowns than degrees of freedom. The third point is not considered in the DLT and its solution is the combination of the algorithm with methods from *robust statistics* (see Sec. 2.5.4).

Tsai [Tsa87] proposed a two steps approach that considers both the structure of the solution and the radial distortion of the lenses. In the first step the algorithm computes the solution for the rotation and the translation (except for the z component) by exploiting the *radial alignment constraint*. The second step computes the solution for the remaining parameters using a set of non linear equations.

If there is not a fixed world reference frame and we require only the intrinsic parameters of the camera, it is possible to exploit a known planar pattern that moves freely in front of the camera. More specifically it is possible to calibrate the intrinsic parameters by estimating both the matrix K and the distortion parameters using two views of a moving known planar pattern without requiring any knowledge on the motion of the object [Zha00]. The latter result has been further extended to consider the use of multiple views of a known rigid object with three points on a line of which two moving an the other one fixed [Zha04].

2.5.2 Distortion estimation

A possible alternative to the previous algorithm is to consider separately the distortion parameters and the matrix P as the coefficients of Eq. 2.8 can be computed independently from K. The distortion can be found both using points (e.g. [Ste97]) or enforcing the straightness of lines that in an image appear curved (e.g.[PM97]). An advantage of line based approaches is that in

most of the cases in an urban environment are available plenty of straight lines that can be easily detected by a human with no knowledge of the specific scenario.

Given a set of lines that in the world are straight, the approach can be based on a non linear optimization method to compute the distortion parameters that minimize the sum of the squared distances between the lines curved by the radial distortion and the straight lines fitted on them [PM97]. In Figure 2.6 it is shown an example of the application of an implementation of such approach on a real world camera.

The frequency domain of the image can be analysed with heuristics to detect the amount of distortion with no knowledge of the acquisition device [FP01], however in this case the aim of obtaining a completely automatic algorithm is different from the general setting studied where we assume to have the possibility of some user interaction during the calibration and we require a precise solution.

2.5.3 Multi-camera geometry estimation

The same procedure applied to the single camera calibration with the DLT can be applied to compute the fundamental matrix, the essential matrix and a homography starting from appropriate points correspondences and deriving a system from Eq. 2.14, 2.21 or 2.25. However, with the fundamental matrix and the essential matrix, we have to take into account the constraints of Eq. 2.20, 2.23, 2.24 that cannot be guaranteed by the solution obtained with the DLT.

In the *eight point algorithm* [LH87] the constraints are enforced by decomposing with the SVD the matrix F or E and modifying the results.

Given the decomposition

$$U \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} V^T = F, E \tag{2.35}$$

the matrices are updated with

$$F = U \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T \tag{2.36}$$

$$\tag{2.37}$$

or in the case of the essential matrix with

$$E = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T. \tag{2.38}$$

(a)

(b)

(c)

Figure 2.6: (a) Radial distortion of a wide angle lens; (b) the green lines in the world are straight as the red dotted lines, but appear curved due to the barrel distortion; (c) picture (a) corrected enforcing the straightness of lines.

The result is the closest matrix w.r.t. the Frobenius norm to the one obtained solving the linear system.

To improve the numerical stability the points correspondences need to be normalized such that the sets of points on the two images have mean 0 and Root Mean Square distance from 0 equal to $\sqrt{2}$ [Har97]. Given the matrices N_1 and N_2 that normalize the points of the two cameras and the fundamental matrix F_N obtained with the DLT on the normalized sets, it is possible to compute the solution on the non normalized coordinates as:

$$F = N_1^T F_N N_2 \tag{2.39}$$

since

$$(N_1 p)^T F_N (N_2 p_1) = p^T N_1^T F_N N_2 p_2 = 0. \tag{2.40}$$

An alternative to compute the solution is to apply a constrained optimization method to enforce the constrains while minimizing the error. However, the precision of the results is obtained at the cost of a higher complexity that makes the algorithm slower. In this case the DLT is generally used to initialize the optimization algorithm.

2.5.4 Robust statistics methods

In the case the points are extracted automatically, the computation of the desired relation (e.g. F, H, P) cannot be reliably computed using the least squares as we can expect the presence of errors (outliers) that do not respect the Gaussian noise assumption that is built-in in the least square solution. The risk is to move away the solution from the correct one due to the effect of a few wrong correspondences.

The better known solution to fit a relation to a set of data containing outliers is the Random Sample Consensus algorithm (RANSAC [FB81]). The assumption of the algorithm is that the data can be partitioned in two parts: a set of random outliers that are not related to the solution and a set of noisy data samples that are sampled from the correct relation. The idea is to compute multiple solutions from different subsets of data until the probability of having selected a subset of data with no outliers is sufficiently high. The correct solution is hence selected by evaluating the size of the consensus on the whole set of points (i.e. the subset of data for which the relation holds).

The scheme of the algorithm is shown in the Algorithm 1.

The number of iteration N usually is set as

$$N = \frac{log(1-p)}{log(1-w^n)} \tag{2.41}$$

Algorithm 1 RANSAC

Require: X, N, n
  $y \leftarrow 1$
  while $i < N$ **do**
    $X_s = \text{sample}(X, n)$
    $R_i = \text{estimate}(X_s)$
    $err = \text{error}(R_i, X)$
    if $err < min_{err}$ **then**
      $R = R_i$
    end if
  end while
  return R

where p is the required probability of selecting a set of all inliers, w is the proportion of inliers in the data and n is the number of points required to fit the solution. Eq. 2.41 ensures that, with probability p, a set of n inliers only will be extracted at least once. To estimate the percentage of outliers in the data it is usually computed the cardinality of the consensus set of the best relation computed, that can be selected by thresholding the errors to distinguish between inliers and outliers.

In the original formulation of RANSAC the error function evaluates the loss defined as

$$l(e) = \begin{cases} C, & \text{if } |e| > T \\ 0, & \text{otherwise} \end{cases} \tag{2.42}$$

where T is a user defined threshold. This simple thresholding does not allow to distinguish between two solutions with the same number of inliers but with different precision. To overcome this limitation has been proposed M-estimator SAC (MSAC) [TZ00]. The difference with RANSAC is in the loss function, that is:

$$l(e) = \begin{cases} T^2, & \text{if } |e| > T \\ e^2, & \text{otherwise} \end{cases} \tag{2.43}$$

Further evolutions of RANSAC are for example MLESAC and MAPSAC [Tor02] (see [CKY09] for a comparison), different approaches that aim to solve similar problems include voting schemes or the optimization of errors with robust non convex loss functions (e.g. Lorentzian instead of the L2 norm).

In case most of the data come from a degenerate configuration and do not have enough information to compute a full solution, RANSAC may select as outliers the few points that are useful to complete the computation. To solve this issue it has been proposed QDEGSAC [FP06] that is a

two steps algorithm: in the first it looks for the number of degrees of freedom that can be fixed using most of the data and in a subsequent step looks for the data needed to complete the computation. An example of possible application is the quasi planar scenario presented in Sec 2.4.2, but differently from the algebraic solution, the method should be able to achieve the same result with no a priori information on the type and on the presence of the data degeneration, estimating a first solution using 4 points and looking separately for the remaining 2 points out of the plane.

2.6 Discussion

In this chapter we have shown a model of the image formation that takes into account position, sensor geometry and lens distortion and is useful to understand the relation between the pixel of the image and the light ray read by the sensor. Moreover we have presented the tools (fundamental and essential matrix) to model the relations between the pixels of different cameras.

Finally we have given an overview of the methods to extract the relations presented starting from manual annotations of images and robust statistics algorithms that are useful to manage automatically extracted data that may contain wrong information.

We have employed the tools presented in this chapter in an empirical study on computation of the fundamental matrix in a quasi degenerate configurations, the results are discussed in [ZOV+10, LNM+12].

Chapter 3

Motion Analysis

This chapter presents an overview of the methods available in the literature to extract information on the relative motion between camera and world and the motion of objects in the scene. Sec. 3.1 introduces the chapter, in Sec. 3.2 it is analysed the camera motion from a geometrical viewpoint without considering the motion of world objects. In Sec. 3.3 it is studied the opposite case, where we are interested in the motion of the objects rather than the motion of the camera. Finally in Sec. 3.4 we present some methods to extract higher level information from moving objects.

3.1 Introduction

In this chapter we analyse tools useful to model motion and to extract information from it.

Apparent motion may be caused by objects in the field of view of the camera, by translation/rotation of the camera or both. Computer vision tools that can be applied to extract information from motion differ for the requirements and for the semantic information that are able to extract.

A common assumption is that only the camera or only the objects move in the scene, whereas more general approaches are more complex and computationally intensive (e.g. object detection and tracking) or less semantically meaningful (as for example sparse point tracking).

3.2 Camera motion estimation

To recover the motion of a camera, we can consider the virtual multi-camera system whose views are images from the same camera but acquired in different instants and poses. The required input

for a calibrated estimation are the fundamental matrix F and the matrix of the intrinsic parameters K, that, since the camera is the same in the two frames, is the same for both the views.

The essential matrix E can be obtained from Eq. 2.21 using the same K for both the virtual cameras:

$$E = K^t F K. \tag{3.1}$$

From Eq. 2.22 we know that E encodes up to a scale all the information on the motion of the camera. Our objective is to decompose the essential matrix in rotation and translation (the latter up to a scale factor).

We can assume with no loss of generality that one of the viewpoints corresponds to the world reference frame, hence the rotation and the translation that we want to compute are the one that move the origin of the world in the position and orientation of the second acquisition point.

Given the SVD decomposition of E:

$$E \propto U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T \tag{3.2}$$

and the matrices

$$W = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.3}$$

$$Z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

we can compute the three matrices

$$[e]_X = UWU^t \tag{3.5}$$

$$R_1 = UZV^T \tag{3.6}$$

$$R_2 = UZ^TV^T \tag{3.7}$$

It is easy to show that the decompositions $[e]_X\ R_1$ and $[e]_X\ R_2$ are valid. Eq. 2.22 can be written as:

$$[e]_X R_1 = UWU^tUZV^T = UWZV^T = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T \propto E; \tag{3.8}$$

the same holds for R_2. These two decompositions are unique (for a proof see [HZ00] Result 9.18).

Since there are two possibilities for the rotation matrix and the sign of the translation cannot be determined, we have four possible solutions given by their combinations.

To disambiguate between them we can use one corresponding point on the two images, it is possible to show from the relation between the four solutions that there is only a possible combination of rotation and translation for which $\boldsymbol{Q}$ appears in front of both cameras, that is the correct decomposition of E.

If we know a priori that the motion does not include rotation, it is possible to extract directly the direction of motion from E: in this case we can write Eq. 2.22 as

$$E = [\boldsymbol{e}]_{\times} \tag{3.9}$$

where $\boldsymbol{e}$ is the direction of motion. This case is particularly interesting as, with no rotation, it is possible to compute the fundamental matrix using only points on a plane (that is not possible in the general case) as the ambiguity shown in Chapter 2 is removed by implicitly setting a priori all the angles to 0.

3.3 Observing moving objects in the scene

In the case we are interested in the objects that move in the scene rather than in the motion of the camera, we have to rely on objects tracking algorithms. In this case our objective is to detect (potentially) moving objects and possibly track them over time.

At higher level the two main steps of the procedure are:

- **Object detection**: an algorithm to locate/identify the objects we are interested in.
- **Object tracking**: a method to associate the same object between consecutive observations.

In the following subsections we will overview some well known methods to extract interesting objects and, finally, to track them.

3.3.1 Interesting objects detection

Our objective is to identify the objects that we are interested to follow with a tracking algorithm. In the literature it is possible to find three main possible classes of approaches:

- To detect the differences between a background model and the current frame (motion-based detection).

- To extract a set of local points that are easy to re-identify in the following observations (feature-based detection).
- To detect a semantically significant class of objects (appearance-based detection).

The first class of solutions is generally applied with static cameras and it allows us to segment moving objects in the scene. More in general with moving cameras we refer to *motion segmentation* (e.g. [IRP94]).

In the second class of methods motion is detected by tracking local features over time. In this case it is not required to have a static camera, however it is not straightforward to recover the semantic meaning of the extracted points as a single object (e.g. a person) can generate a variable number of points.

The third class of algorithms usually exploits a statistical learning algorithm for *classification* that scans the image, possibly at different scales, looking for the a predefined class of objects.

3.3.2 Change detection

If we assume that the camera is fixed, it is possible to build a model of the background of the scene and to find the differences with a new frame.

To extract meaningful and reliable information on the objects in the scene, an algorithm based on background modelling needs to:

- Be robust w.r.t. to illumination changes.
- Be able to update the background model with stable changes.
- Learn periodic motion in the scene (e.g. the leaves of a tree in the background moved by the wind).
- Compute a precise segmentation of the objects of interest.

The simplest approach [Kar90] is to compute a background image B as a running average of the observed frames. Given the image I_t received at the instant t we can update the background image with:

$$B_{t+1} = \alpha B_t + I_t(1 - \alpha) \tag{3.10}$$

where the value of $\alpha \in (0, 1)$ rules the speed of the update of the background w.r.t. newer frames. Small values of α will incorporate foreground objects that move slowly, higher values may be too slow to react to permanent changes in the scene and to variations in the illumination. The background update speed α can be differentiated to slow down the update of the pixels

(a)

(b)

(c)

Figure 3.1: (a) Frame of a video; (b) the associated background computed using a low α is affected by slowly moving people; (c) the same background computed with a higher α.

detected as foreground (and hence the absorption of static objects) using a different value α' for all the pixels that are detected as foreground [Kar90]. However this approach slows down also the absorption of artefacts that appear in the foreground map due for example to changes in the illumination.

To check if the frame I_t contains new objects or not it is possible to check the differences with the background B_t with a pixel by pixel difference in absolute value. The *foreground map* is a binary matrix of the same size of the images that is computed as

$$M(i,j) = \begin{cases} 1 & \text{if } |B_t(i,j) - F_t(i,j)| \geq T \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where T is a value fixed by the user that corresponds to the minimum deviation from the background model that is required to consider a pixel as foreground.

The method can be formally justified considering each background pixel of each frame as the realization of a Gaussian distribution with unknown mean $B_t(i,j)$ (which may change over time) and fixed variance σ^2 (that is the same for all the pixels in all the instants). Eq. 3.10 estimates the mean of the Gaussian of each pixel, while the threshold T can be expressed in this setting as

$$T = k\sigma \tag{3.12}$$

with k the constant provided by the user and σ an estimate of the standard deviation of the distribution.

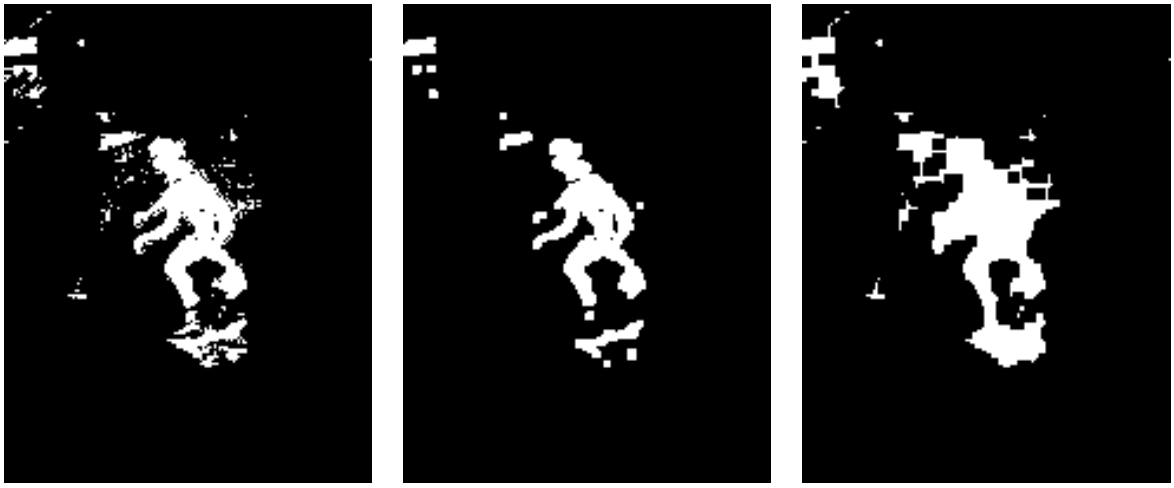
Even if the method has been proposed for gray scale intensity values it can be easily generalized to different colour spaces (e.g. RGB, YUV and HSV) to the price of additional computational cost.

More complex approaches remove the assumption that a pixel is extracted by a single Gaussian distribution, to model backgrounds that have different configurations. The most known method in this class is the mixture of gaussians model [SG99], where a pixel is modelled as the realization of a mixture of Gaussian distributions, each of which represents a different configuration. Another algorithm that learns for each pixel a more complex pattern than a single value (e.g. to model periodic motion) is the codebook model [KCHD05], which compresses the observed pixel in time in a set of *codewords*.

Another assumption that is made comparing each pixel of the background with each pixel of the image is that each point is independent from the others. It is instead reasonably to assume that the information of a pixel can help to understand if its neighbours are foreground or not. A possible way to exploit such correlation is to consider blocks of pixel (e.g. [SWFS03, LLC09]).

One of the main challenges for background modelling methods that has to be addressed to effectively monitor a scene, is to distinguish between real objects in the scene and sudden changes in the pixel values due to illumination variations (e.g. lights switching, clouds).

A detailed evaluation of the state of the art algorithms is available in Brutzer et al. [BHH11].



(a) (b) (c)

Figure 3.2: Examples of morphological operations applied to a foreground map. (a) the input foreground image; (b) an *open* removes small elements from the map; (c) a *close* fills the holes of the map and merge close elements.

Shadow removal

All the methods listed previously look for changes between a background model and a new frame evaluating the intensity value (of one or more channels) of each pixel.

However, while our objective is to detect foreground objects, also shadows modify the values of pixels and will appear in the foreground map. Shadow removal methods aim to discriminate between variations in the pixels values that are due to moving objects and the one that are caused by their shadows.

If we consider a colour background it is possible to detect changes that affect differently the luminance and the hue of the pixels (e.g. [CGP$^+$01, HHD99, KB01]).

In the case of a gray scale background can be considered models based on the analysis of photometric properties of foreground and shadows (e.g. [Bev06, RE95]). A more detailed survey is available in [PMTC03].

Foreground map post-processing

Once the foreground map is extracted, it is usually enhanced with morphological operations [Ser82] on the binary map.

Common operations are *open* and *close*, that are respectively the sequential combinations of

erode and *dilate* and of dilate and erode.

In the erosion a kernel (a mask that selects the pixels affected by the operation) is applied with the center on the pixels of the perimeter of the foreground elements and all the values affected by the kernel are set to 0. The dilate is the result of the same procedure, but the elements in the kernel are set to 1.

The effect of the open is to remove small elements from the foreground map; the close connects elements that are distant less than the diameter of the kernel for a side bigger than the diameter of the kernel (see Figure 3.2).

Connected components labelling

The most common way to extract information from the foreground map is to extract its connected components assuming that each of them has a semantic meaning (e.g. one person or a group of people).

The process of the extraction of the connected components (labelling) has been studied to obtain optimized algorithms. Depending on the specific hardware there are available a broad range of methods: some of them studied to be optimized w.r.t. sequential accesses (e.g. [SHS03, HCSW09]), others (e.g. [CCL04]) to extract at the same time multiple information during the labelling (e.g. the contour of the blob and the number of holes) or to be optimized for FPGAs on embedded systems (e.g. [JB08]).

3.3.3 Stable feature detection

If the camera is not assumed static, techniques based on background modelling cannot be easily applied in real-time.

In this case the most common approaches to detect motion are two: in the first one a set of points easy to be tracked are selected and are searched in the next frame, in the second one (similarly to image retrieval) the extraction is done on both the frames and the points are matched.

A well known example of features that can be used with the first approach are *corners* [HS88] (see Figure 3.3 (a)), that are points that show a strong spatial gradient in the image in two directions. This requirement makes possible a precise localization and ensures a good degree of stability that simplifies the search in the next frame.

Corners are a robust feature to track if the frame rate is sufficiently high w.r.t. the motion [ST94], however they are extracted from a single user defined scale and hence are not stable w.r.t. different distances of the same object, making this approach not suited to work with fast moving cameras and objects.

A first attempt to take into account the scale has been proposed in Lindeberg [Lin98] where it is studied a method to achieve a higher degree of invariance w.r.t. scale variations by looking for extrema in the Laplacian of Gaussian.

To apply the second approach and deal with fast camera/objects motion it is possible to employ the *Scale-Invariant Feature Transform* (SIFT Figure 3.3 (b) [Low99]), whose original task was retrieval. In this case both the orientation and the scale are explicitly handled by the descriptor, whose key-points are searched in the scale space of the image. The invariances achieved make easier to recognize the same point from a different position, orientation and distance. Moreover a description based on the gradients in the neighbourhood is associated to each key-point and can be used to compute the matches.

Another feature that has been inspired by SIFT is the *Speeded Up Robust Feature* (SURF [BTVG06] Figure 3.3 (c)), that maintains the rotation and the scale invariance detecting stable uniform areas in the image.

3.3.4 Object detection

Another possibility to identify interest elements of the image is to look for instances of a specific class of objects (e.g. cars, people) within a frame.

Theses approaches are usually based on statistical learning algorithms performing image classification. The *image classification problem* is a binary classification formulated with a function s.t.

$$C(\phi(I)) = \begin{cases} 1 & \text{if } I \text{ contains an instance of the class} \\ -1 & \text{otherwise} \end{cases} \tag{3.13}$$

where I is an image (or a window over an image) and $\phi(\cdot)$ an appropriate *descriptor* that maps an image to a vector of numbers. Usually it is meant that $C(\phi(I)) = 1$ iff the object has a size comparable to the one of I.

A common way of performing *object detection* is to apply the image classification function $C(\cdot)$ over a window w sliding over the image, possibly considering windows of different size to detect variable size objects (examples of more complex and effective approaches are [GPC12, LBH08].

The output of the detection algorithm is the set of sub-windows W such that

$$w \in W \iff C(w) = 1. \tag{3.14}$$

The main ingredients of an object detection algorithm are:

- A set of examples (*training set* images).

Figure 3.3: Key-points on two frames acquired at roughly one second of distance using: (a) Harris corner detector; (b) SIFT; (c) SURF.

- An *image descriptor* ϕ.
- A learning algorithm for classification.

The training set is the collection of positive (images depicting the object of interest) and negative examples (usually with many more negatives as they are simpler to collect).

The image descriptor, or more in general *feature extractor*, is a function $\phi(\cdot)$ that, given an image in the same format of the ones in the training set, extracts a vector of numbers that represents the sample.

The statistical learning algorithm processes the descriptions of the positive and negative data in the training set (*training step*) to learn a general rule to distinguish between the two classes (e.g. (face, non-face), (pedestrian, non pedestrian)).

We refer to the training set both as the example images and the descriptions extracted from them depending on whether we are referring to the detector (whose training set is the collection of sample images) or the classifier (whose input are the description vectors).

Usually a single object gives multiple results in similar positions and scales, hence we need a further filtering step to remove redundant matches. Common approaches are the averaging of overlapping results [VJ01] or peak detection using for example the Mean-Shift algorithm [CM02].

Since the scan of an image in all the couples (positions, size) is a time consuming task, in the literature have been studied different approximations to reduce its computational cost [DF12] or to improve its accuracy (e.g. [BLK12]).

In the following part of this section we will focus on some examples from the state of the art of well known features and learning algorithms. Further details are available in Chapter 4.

Image descriptions

The requirements of the image descriptions usually are: (i) a low computational cost (ii) the ability of separating the classes (iii) compactness.

The simplest image description possible is the vector containing the stacked values of the image, however more complex functions are usually considered as it is needed to cope with small shifts in pose and scale, different illuminations and often a high intra-class variability (e.g. the different clothes of people).

In this section are provided details for two examples of well known image descriptions, namely the Haar features [VJ01] and the Histograms of Oriented Gradients (HOG [DT05]). Other common examples are the Local Binary Patterns (LBP [OPH94]), Covariance Image Features [TPM08a], Gray Level Difference Method (GLDM [KP99]), Gabor filters [JF91].

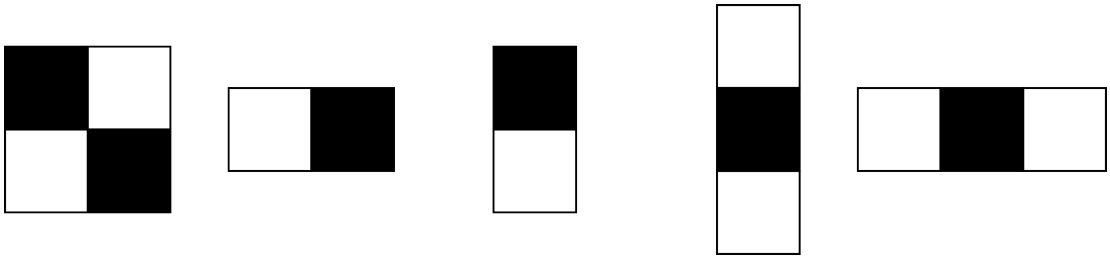


Figure 3.4: Examples of the features extracted from the Haar wavelet basis [VJ01]. Each feature is related to a mask of a specified size and position and is computed as the difference between the sum of the image pixels in correspondence with the white part of the mask and the sum of the image pixels in the black part.

One of the most known methods for object detection in real time [VJ01] is based on a feature set derived by the **Haar wavelet basis**. The dictionary of features is composed by an over-complete set of simple descriptors, each of which is the result of the difference between the pixels in two masks of different shape position and size. In Figure 3.4 are shown the shapes of the masks: the result of the application of a mask on a portion of the image is the subtraction between the sum of all the pixels in the black area and the sum of the pixels under the white part.

The strength point of these features is the possibility of computing each of them with only a few memory accesses and with a cost that is independent on the size of the mask. To this aim it is possible to compute the *integral image* (II), that is a matrix that contains in each position the sum of the pixels of the image (I) in the upper left part w.r.t. it:

$$II(x,y) = \sum_{x^I<x, y^I<y} I(x^I, y^I) \tag{3.15}$$

Using this representation it is possible to compute the sum of an arbitrary area with four operations as it is shown in Figure 3.5, and hence each mask can be computed in constant time and independently from its position and size.

Even if the expressive power of each single feature is small, the combination of the responses of multiple masks has proved to be effective for face detection. A drawback is that all the masks in all the combinations of shape position and size, create a vector of responses that grows rapidly and it is needed to select a meaningful subset of them to exploit effectively this description.

The **Histograms of Oriented Gradients** (HOG [DT05]) are a description proposed to detect pedestrians. The computation of the HOG proceeds in the following steps: in the first step the image is divided in a regular grid of non overlapping *cells* of fixed size and for each cell it is computed a histogram of the orientations of the gradients weighted by the magnitude of the gradient (in the case of colour images the gradient magnitude of each pixel is the highest of its three channels).

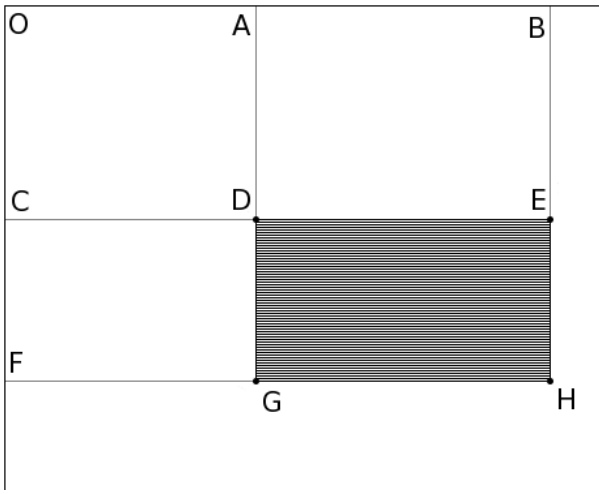


Figure 3.5: The sum of the pixels in the square DEGH can be obtained by subtracting from the value in H of the integral image the values in E and G, and adding D. With this procedure it is possible to compute the sum of an arbitrary area with only four accesses to the integral image.

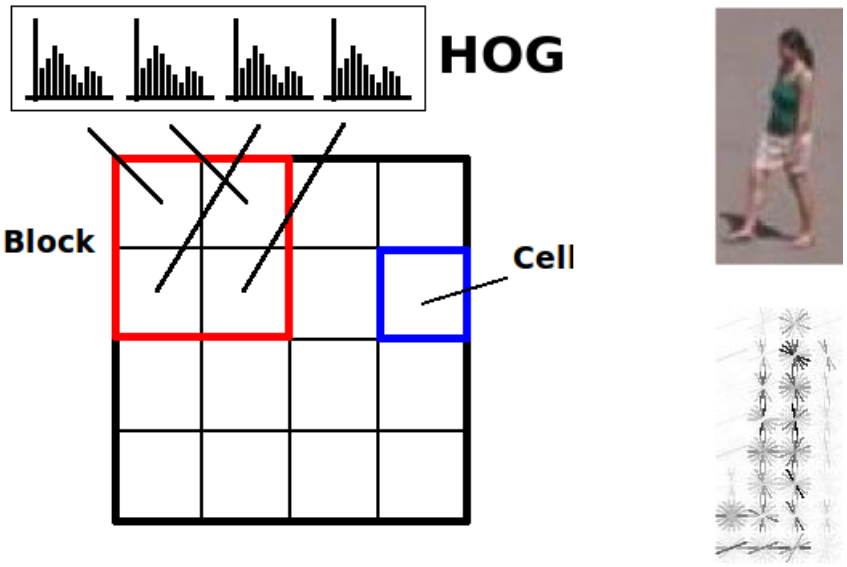


Figure 3.6: Scheme of the computation of the HOG. To each cell (black grid) is associated a weighted histogram of the orientations of the gradient. The cells are grouped in blocks, that are the concatenation of the histograms and each block is normalized. On the right we show a visual representation of the cells of the HOG computed on a sample image.

In the second step the histograms of the cells are grouped in a set of overlapping *blocks* of fixed size each of which is normalized w.r.t. the L_2 norm. The values higher than a fixed threshold are cut and each block is normalized again. The output of the description is the concatenation of all the normalized histograms of all the blocks.

Learning algorithms

The objective of learning algorithms is to extract meaningful information from a set of available data (*training set*) for prediction on new data.

The class of **supervised algorithms** contains all algorithms (e.g. Support Vector Machines (SVM) [CV95], Regularized Least Squares [HTFF01]) that compute their solution starting from a training set of examples in the form $(\boldsymbol{x}, y)$, where $\boldsymbol{x}$ is a vector of data (input) and y is an output assigned to the vector.

The problem can take the form of *binary classification* if the possible outputs (or labels) are two (generally 1 for the positive class and -1 for the negative class), *multi-class* classification if the possible labels are a finite set of values or *regression* if the labels are real values.

The examples are supposed to be generated by a fixed unknown probability distribution $P(\boldsymbol{x}, y)$, and the objective is to estimate the function f that minimizes the expected risk

$$E = \int_{X \times Y} L(f(\boldsymbol{x}), y) p(\boldsymbol{x}, y) dx dy \tag{3.16}$$

where L, the loss function, measures the error when approximating the correct output with the estimate.

Since the training set is finite, it is only possible to approximate Eq. 3.16 with the empirical risk

$$E = \sum_{i}^{N} L(f(\boldsymbol{x_i}), y_i) \tag{3.17}$$

where N is the number of examples in the training set.

The minimization of the empirical risk is an ill-posed problem and has not a single solution. Moreover, we are interested in computing a solution that is able to compute the correct output for samples extracted from the same probability distribution that, however, are not in the training set.

A common solution is to penalize complex functions with the objective to balance the ability of the algorithm to minimize the empirical risk and the stability of the solution w.r.t. small changes in the training set:

$$\arg\min_{f} ERM + \lambda PEN \tag{3.18}$$

where ERM is the empirical risk minimization, PEN a penalty for the complexity of the solution and λ a user provided parameter that balances the two terms.

In the regularization framework [EPP00], to obtain generalization ability from learning algorithms the space of the possible solutions is limited by a regularization parameter. The general procedure can be defined as

$$f = \arg\min_{f \in H} \frac{1}{N} \sum_{i}^{N} L(f(\boldsymbol{x_i}), y_i) + \lambda ||f||_H \tag{3.19}$$

where H is an appropriate hypothesis space where we look for the solution f and λ is a regularization parameter that controls the complexity of the possible solutions that is defined by the norm $||f||_H$.

Using different loss functions in Eq. 3.19 we can obtain different algorithms (e.g. SVM with hinge loss and RLS with squared loss function).

3.3.5 Tracking

Tracking algorithms aim to find the association between subsequent observations of the same object observed in different instants.

The inputs of the tracker are a set of features (e.g. interest points, blobs, detected objects) extracted from the last frame I_t and the history of the observations. The desired output is a set of updated sequences where, in each of them, appears the same world object observed during time.

This result can be achieved extracting a description from each observation (e.g. position, size and appearance) and looking for similar ones in the following frame.

In simple cases can be applied very simple algorithms that compare all the objects that are similar enough and link together the closest objects. However the presence of occlusions, noise and of high number of objects in the observation make the problem more complex.

A first attempt to track objects from noisy and incomplete information (due to occlusions) has been the application of the dynamic filters: one of the most known example is the Kalman filter [WB95]. For each observed object the Kalman filter computes the coefficients (the state of the object) of a user provided linear model that represents how the state evolves over time (e.g. the state can be the velocity and the position so that a linear model can compute the next position based on its internal state). Other than the linearity of the model, the assumption is that the noise is Gaussian and has mean 0.

At each instant the filter computes a prediction on the future state of the object exploiting the internal state and the model and, once new information is received, it corrects the internal state balancing the noise model of the input data and the uncertainty in the model.

The advantage of the Kalman filter is twofold: it works as a filter of noisy measurements and, thanks to the predictive power, it helps to recover the tracking after interruptions in the observations (e.g. due to occlusions).

Extensions of the Kalman filter have been studied (e.g. Extended Kalman Filter [EW99], Unscented Kalman Filter [WVDM00]) to remove the assumptions on the linearity of the model. However, while the linear Kalman filter is statistically optimal, this is not true in the case of the non linear approaches.

More recently has been proposed particle filters tracking [RAG04] that generalizes the hypothesis of the UKF to work with non linear models and non Gaussian noise.

Another well known tracking method is the Multiple Hypothesis Tracking (MHT [Bla86]) that computes the associations based on the history of the previous frames. In this algorithm at each instant are maintained multiple possible associations, even if generally only the most probable sequence is shown to the user. The key point of this algorithm is that the final decision on a track is delayed with the objective of collecting more information. Hence, given a new observation, it may change both the most probable association and its history.

A different approach based on the appearance of the tracked objects is the mean-shift tracking [CRM00], where each object is tracked independently from the others by using an iterative method to look for the most similar part of the image to the target.

3.4 Description of the dynamic

Once tracking information is available, it is possible to find a spatio-temporal description of the object dynamics useful for high level analysis.

A possible instance of methods that analyse the dynamic of one or more objects is *action recognition*, another common application is the modelling of the normal dynamics for *anomaly detection*. In both cases, similarly to the features extraction in object detection, we are interested in the computation of a numerical representation of the evolution of the appearance in time that should be generic enough to make possible to group intra class elements and specific enough to separate different classes.

The description of the dynamic of objects and *action recognition* algorithms can involve tracking (e.g. [YB98, Bre97]), or the analysis of space-time volumes [Lap05a].

An example of the latter class of descriptions are the *Space Time Shapes* [GBS$^+$07], where the evolution of an action is encoded as the three dimensional shape that on each slice along the z axis has the segmented object of interest acquired in a specific instant. Those three dimensional shapes can be converted in a compact numerical description by computing local and global features (e.g. local orientations and global moments).

Another example of description of the evolution of the appearance in time are the Self Similarity Matrices (SSM) [JDLP11]. This description encodes the evolution of the appearance of an object in time as a function of the differences between observations from different instants. The idea is to compute a matrix that in each entry contains the similarity between the configuration at the instant associated to the row and the one observed in the instant associated to the column:

$$S(y, x) = 1 - ||x - y||_2. \tag{3.20}$$

The key feature is that, by encoding the repetitions of the configurations, it is possible to have a description of the action in time that is independent from the static appearance of the object (e.g. the dresses of a person) and robust w.r.t. viewpoint changes.

A more detailed survey on actions modelling from images is available in [Pop10].

3.5 Discussion

In this chapter we have presented an overview of computer vision techniques that can be adopted to model motion and to extract information from it.

In Sec. 3.2 we have analysed the information that can be extracted from a geometrical viewpoint if the camera moves, differently, in Sec. 3.3 we have summarized techniques to model motion of objects in the scene.

All the different techniques presented in Sec. 3.3 allow to extract information on the moving objects in the scene. The difference between them resides on the objects detected. In the case of change detection the object of interest is everything different from the background independently on its semantic meaning; in the case of feature tracking the object of interest is the motion of sparse points and, together with the lack of semantic meaning, there is no explicit grouping of the elements in the scene; with object detection we are only indirectly interested in motion (generally to objects that potentially move), however it is a tool to select a subset of objects of interest that belong to a specific semantic class (e.g. in video surveillance we have cars and people) and to model their motion.

Together, the analysis of the motion of the camera and of objects allow to extract information on the real motion in the scene w.r.t. a world reference frame, moreover, being able to detect coherent changes due to camera motion may be useful for example to switch between techniques that require static cameras (e.g. change detection) and others able to work independently on camera motion (e.g. corner tracking).

Chapter 4

Object detection

In this chapter we present our study on image descriptions for object detection, with a specific focus on pedestrians. The aim is to learn meaningful and compact representation from data taking into account the structure of the computation. This study has the dual objective of minimizing the computational cost and maximizing the accuracy of the results. In Sec. 4.2 we overview the best known people detection algorithms from the literature, in Sec. 4.3 we introduce regularized algorithms for feature selection, in Sec. 4.4 we show a framework based on structured sparsity to build a representative description for object detection.

4.1 Introduction

The general objective of object detection algorithms is to find the occurrences of a class of interest in an arbitrary image.

The most common approach to detection (see Sec. 3.3.4) finds the objects by scanning all the possible image sub-windows (that varies for position and size) and using a machine learning algorithm to decide, on the basis of a numerical description, if they contains or not an instance of the requested object.

While the machine learning algorithms applied in computer vision are mostly standardized (the SVM [HTFF01] is the best known example), the crucial step in this process is the computation of a meaningful description to find a robust solution to discriminate between the required class of objects and the background.

In the literature have been proposed different methods for object detection that start from an *over-complete* dictionary of possible features and use a *feature selection* algorithm to find in the dictionary the elements that are suited for the specific task [OF97, VJ01]. The idea is that the

full dictionary is a generic image representation, that can be specialized for a specific task by removing irrelevant elements.

In this chapter we present our study on a framework based on sparsity enforcing regularization techniques for feature selection to compute a description starting from an over-complete dictionary. Our aim is to exploit the computational flow of the image description and detection to achieve a good balance between accuracy and computational requirements.

More specifically we focus on the specific problem of pedestrian detection as it is the typical object of interest in video surveillance. Also, detecting people is a challenging task due to the strong intra class variability induced by the different poses of a non rigid object and hence is a good benchmark for object detection algorithms. However all the techniques presented can be generalized to different tasks.

4.2 State of the art

Pedestrian detection has long been studied in the computer vision literature, particularly with applications to smart vehicles, video-surveillance, and video mining. To represent data, motion and shape information have been adopted in different combinations.

The approaches in the literature can be classified in groups based on the description extracted. Usually state of the art methods exploit:

- Holistic descriptions of the appearance of the global pedestrian.
- Descriptions of single parts of the person (e.g. head, feet).
- Motion.

In the remainder of this section we give an overview of the state of the art of pedestrian detection, following the proposed classification.

Holistic descriptions

Compared to other object classes of interest, such as faces and cars, pedestrians are challenging for their wide variability. Thus usually they are represented by means of local descriptions [PP00, DT05] which are more tolerant to appearance changes [MG06]. Combinations of local and global cues have also been considered as for instance integrating iteratively local and global cues using a top down segmentation approach [LSS05].

The Viola and Jones algorithm [VJ01], one of the best known algorithms for object detection in real time, proposed rectangle features derived by the Haar basis to describe objects (see Sec. 3.3.4). Other than the description, one of its main contribution has been the design of the *cascade of classifiers* as a method to achieve a rapid and precise detection. The idea is to have a set of simple sequential classifiers each of which decides only if the input image might be a positive example: if the answer is positive the example is given to the next classifier, otherwise it is discarded. The advantage of this approach is that *weak classifiers* do not need to have high performances as, for example, with five independent classifiers able to reach a true positive rate of 0.99 and a false positive rate of 0.5, the global solution achieves a true positive rate of 0.95 and a false positive rate of 0.03.

Among the local features proposed for people detection, Histogram of Oriented Gradients (HOG [DT05]) are among the most popular techniques for representing pedestrian data presented in recent years; according to many authors the main limitation of HOG is their computational cost, and indeed more recent methods have been proposed to improve their performances. A possible solution to speed up the detection is the combination of HOG features with a boosted cascade of classifiers and a computation based on integral images [ZAYC06], moreover it is possible to combine the HOG with simpler feature in a cascade architecture (e.g. Haar wavelets [WS08]).

Other features proposed recently that have been applied to pedestrian detection are the covariance features [TPM08b]. The description is composed by an over-complete set of covariance matrices computed on different sub-windows considering a fixed set of variables (e.g. position, gradient magnitude and orientation). They can be extracted efficiently in constant time independently from the size of the window with a generalization of the integral images. Similarly to the description based on the Haar wavelets, such features produce a very high dimensional feature vector, therefore they also have been coupled with feature selection to build classifiers cascades [TPM08b, PSZ08].

A very effective method can be based on HOG-like descriptors and a SVM classifier equipped with a histogram intersection kernel [MBM09]. An approximation of the classifier allows the authors to achieve comparable results to the exact classifier with a limited computational cost.

A recent trend in object detection is to combine together different features to improve the accuracy. For example HOG have been combined with other features (to include, for instance color or texture information): Schwartz et al. [SKHD09] proposed a description based on HOG, co-occurrence features and color frequency features that is extracted in combination with a dimensionality reduction scheme based on partial least squares; such approach controls the size of the feature vector but does not decrease the amount of computation needed, thus the authors propose a two layers classifier that allows them to reduce the considerable computational cost of the method.

Another example is the combination of Local Binary Patterns (LBP [Mäe03]) and HOG for human detection [WHY09]: the main novelty of this approach is the explicit handling of occlusions,

that are detected analysing the local response of each HOG block and their influence is avoided triggering whenever it is needed a body part detector.

Wojek et al. [WS08] proposed a systematic study of the combination of different features (Shapelets, Haar-like features, HOG, Shape Context), that has been extended to consider the computation of self similarity features (CSS) together with HOG and Histogram of Oriented Flow (HOF [DTS06]) [WMSS10].

A further work [DTPB09] considered a set of computationally efficient features selected to be easily computed with integral images (gradient histograms, different colour spaces, and gradient magnitude). While the system performs well, it is not clear the effect of using colour information as in principle both the class of pedestrian and background have no colour constraints.

Part-based approaches

A different class of approaches aims to detect a pedestrian through its sub parts using a part-based model. Felzenszwalb et al. [FGMR10] proposed to combine results from a low resolution root filter (that looks for a global human body) and from a set of part filters computed on images with twice the spatial resolution of the root. Such procedure employs the Latent-SVM (LAT-SVM) to train a part-based detector. A possible variation of the method considers the training of a cascade architecture to reach similar accuracy with fewer computational requirements [FGM10].

The main disadvantage of a part-based model is that the part filters require a high image resolution to achieve good results and in general, especially in a video surveillance setting, we may be interested in locating people distant from the camera and/or from low resolution images. This has been confirmed by an empirical evaluation on the Caltech pedestrian dataset (see Appendix A.1) [DWSP09a] that shows how, w.r.t. other algorithms in the state of the art, the method proposed by [FGMR10] obtains good performances on near scale objects, however its accuracy drop faster than other methods as the size of objects decreases.

A possible solution is to generalize the part-based approach to take explicitly into account the scale of the object in a multi-resolution model as in [PRF10]. In this work the idea is to use a rigid model together with a deformable part-based model while considering images with a sufficiently high resolution and discarding the latter if the resolution is not sufficient. Moreover the method takes into account the geometry of the scene and penalizes incoherent detections. To this aim the camera is assumed parallel to the ground, the environment to be planar and each person is assumed to have a fixed eight. The general idea of considering pose and scale information to improve detection results was proposed earlier in Hoiem et al. [HEH06]. If the pose of the camera is not known, contextually to the detection it is possible to estimate the ground plane using stereo cameras and depth maps [ELVG07] to constraint the possible detections to results that are consistent with the geometry.

Motion descriptions

If we consider videos, it is possible to exploit temporal information to extract motion descriptors. In the literature there are only a few approaches, as generally the input of detection methods are images instead of videos. However their results suggest that motion is a valuable feature for human detection.

A first approach in this direction proposed to extend the Haar features to to consider a spatio temporal volume instead of the single image [VJS05].

In the case where both the cameras and the objects possibly move it is possible to consider the combination of HOG and Histogram of Oriented Flow to improve the detection accuracy [DTS06].

4.3 Feature selection

Feature selection [GE03] is a process that, given a set of features, allows us to choose the most relevant subset for a specific task out of a large and possibly redundant dictionary.

The task of reducing the size of the feature vector has different meanings in different contexts: it can be a standalone procedure to understand what are the variables that are relevant for a specific task or a pre-processing step to reduce the dimensionality of the feature vector. The first task is typical of biological application (e.g. to understand what are the genes related to a pathology or a biological process) while in computer vision feature selection is often adopted together with *over-complete* features (e.g. [VJ01, ZAYC06, TPM08b]), where an image generates a very high dimensional redundant description.

By reducing the size of the feature set we achieve two objectives:

- At runtime the computational cost of the extraction of the description is reduced.
- The dimensionality of the feature space is reduced making easier the related classification problem.

The latter point is due to two factors: from one side if the dimensionality of the feature space is high, machine learning algorithms struggle to compute a robust solution (see *curse of dimensionality* [HTFF01]), on the other side a smaller description occupies less memory and, with the same computational resources, with smaller descriptions it is possible to increase the number of training points.

The optimal selection of the best subset of features needed for a specific task is computationally intractable as it requires an exhaustive search of the possible solutions. To obtain tractable al-

gorithms in the literature have been proposed different algorithms that approximate the optimal solution.

The simplest approach (the so-called stage-wise algorithm) is to start from a set of features and to compute a greedy solution iteratively adding/removing the most/less relevant one. Probably the most known approach is Adaboost [HTFF01], that looks for a solution composed by a linear combination of *weak classifiers* (classifiers trained on a single feature) that are iteratively added starting from an empty set.

At each step the Adaboost adds a new weak classifier and computes its final weight in the linear combination, leading to a greedy solution. Further methods propose to slow the learning process by shrinking the weights of the added classifier [HTFF01], however such approach is rarely applied in computer vision as the choice of the shrinking parameter is not trivial and slows down the learning process.

In this thesis we are interested in studying a regularized approach to feature selection, that, thanks to its theoretical properties, is an appealing tool (however it is rarely used in computer vision) for feature selection from big sets of data. For the interested reader we refer to [HTFF01] (chapter 16) which proposes an interestingly overview of the link between the regularized approaches and the stage-wise approaches.

4.3.1 Regularized feature selection

Given a training set of i.i.d. data $(x_i, y_i) \in X \times Y, i = 1, \ldots, n$, $x_i \in \mathbb{R}^D$, $y_i \in \mathbb{R}$ where (x_i, y_i) are samples couples of input x_i and output y_i, we consider a dictionary $(\phi_j)_{j=1}^D : \mathbb{R}^L \to \mathbb{R}$ of features that extract a real value starting from an input of size L.

We formulate our problem as a generalized linear model

$$y = \sum_{j=1}^{D} \phi_j(x)\beta_j + \beta_0 \tag{4.1}$$

where β_j weights each atom of the chosen dictionary and β_0 is the bias of the solution.

The goal is to find a sparse β so that only the atoms associated to a non zero β_j are meaningful for the solution and hence have to be computed.

The *Least Absolute Shrinkage and Selection Operator* (LASSO) solution [Tib96] to Eq. 4.1 is

$$\beta^* = \arg\min_{\beta} \sum_{i=1}^{N} (\beta_0 + \sum_{j=1}^{D} \phi_j(x_i)\beta_j - y_i)^2 \tag{4.2}$$

subject to:

$$\sum_{j=1}^{D} |\beta_j| \leq t \tag{4.3}$$

where t is a user defined parameter.

Eq. 4.3 can be written in matrix form as

$$\beta^* = \arg\min_{\beta} ||\beta_0 + \Phi\boldsymbol{\beta} - \boldsymbol{y}||^2 \tag{4.4}$$

where $\Phi \in \mathbb{R}^{\mathbb{N}\times\mathbb{D}}$ is a matrix containing in each row i the vector $\phi(x_i)$, and $\boldsymbol{y}$ is the column vector of the stacked y_i.

The solution of LASSO is obtained computing the minimizer of the squared error inside the D-dimensional L_1 sphere of radius t. If t is high enough the solution corresponds to the least square estimate, as the space of the solution shrinks generally a lower number of features have non null coefficient β_j and hence less features are selected.

In Figure 4.1 we report a bi-dimensional example where the axis correspond to the two elements β_1 and β_2 of the weight vector, and the green square is the feasible set defined by Eq. 4.3. The LASSO solution is the intersection between the lowest contour line of the squared error of Eq. 4.2 and the green square associated to the chosen t. The example shows how with a higher t both the features are non zero, while with a lower t only one feature is selected.

The functional is not strictly convex, thus the minimum of 4.2 is unique, however it may have multiple minimizers (e.g. one of two identical variables can be selected randomly with no effect on the value of the function).

Eq. 4.3 can be inserted in a regularized framework writing the Lagrangian form of the constrained optimization problem:

$$\beta^* = \arg\min_{\beta} \sum_{i=1}^{N} (\beta_0 + \sum_{j=1}^{D} \phi_j(x_i)\beta_j - y_i)^2 + \tau \sum_{j} |\beta_j| \tag{4.5}$$

where τ is the Lagrangian multiplier, whose tuning has the effect of controlling the sparsity of the solution.

Group LASSO is an extension of LASSO that considers *structured sparsity* [YL06]. In this case we assume that we have partitioned a priori the set of features $\phi_j(\cdot)$ in a set of M groups and we are interested to select groups instead of single values.

If we define G_k to be the set of indexes of β associated to the k^{th} group, the functional of group LASSO can be written as:

$$\beta^* = \arg\min_{\beta} \sum_{i=1}^{N} (\beta_0 + \sum_{j=1}^{D} \phi_j(x_i)\beta_j - y_i)^2 + \tau \sum_{k=1}^{M} \sqrt{\sum_{j\in G_k} \beta_j^2} \tag{4.6}$$

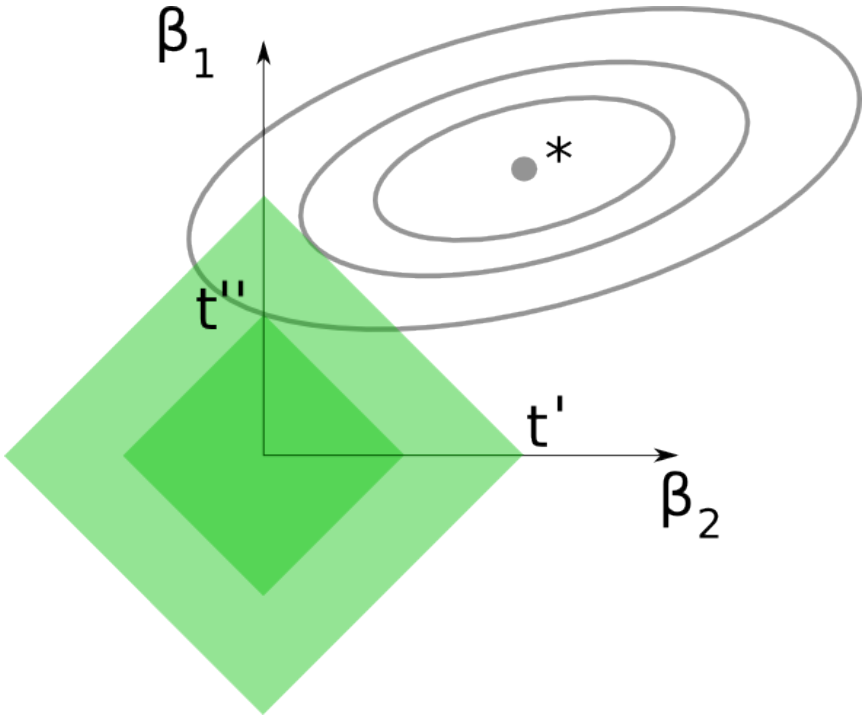


Figure 4.1: The LASSO solution is obtained by intersecting the contour lines of the least square loss (gray) with the L_1 sphere of radius t (green). If the intersection corresponds to a vertex of the green square, only a variable has a non null coefficient. In the example with a $t = t'$ both the variables are selected, with the smaller $t = t''$ only β_1 is selected.

It can be observed that this is equivalent to the LASSO problem, but instead of penalizing the L_1 norm of $\boldsymbol{\beta}$ we penalize the L_1 norm of a vector composed by the L_2 norms of the portions of $\boldsymbol{\beta}$ of each block (so that if the value is 0 all the group elements must have 0 coefficients).

A further variation that can be inserted in the same setting is *Elastic Net* [ZH05] that combines the L_1 penalty of LASSO with a L_2 penalty with the objective of having a single possible solution and controlling the amount of correlated variables to be selected.

4.3.2 Minimization algorithms

Algorithm 2 LASSO - Group LASSO algorithm

1: **require:** $\tau, \sigma > 0$
2: **initialize:** $\boldsymbol{\beta}^0 = 0$
3: **while** `convergence not reached` **do**
p=p+1
$\boldsymbol{\beta}^p = S_{\frac{\tau}{\sigma}}(\boldsymbol{\beta}^{p-1} + \frac{1}{\sigma}\Phi^\top(\boldsymbol{y} - \Phi\beta^{p-1}))$
endwhile
4: **return** $\boldsymbol{\beta}^p$

Since we deal with very high dimensional matrices, to solve the optimization problem of Eq. 4.5 or Eq. 4.6 we adopt proximal methods (see [MRS$^+$10] and references therein). Such methods are accurate, robust (their performance does not depend crucially on the fine tuning on the parameters involved) and computationally efficient. In particular, we adopt an iterative scheme that, with an appropriate choice of parameter σ, converges to the solution of functional [MRS$^+$10].

In the setting of regularized approaches proximal methods optimize the functional decoupling the contributions of the empirical error and of the penalty on the solution considering them in two separate steps that are applied iteratively: first it is applied a step toward the minimization of the empirical error and hence the result is projected using a soft-thresholding operator that is defined from the penalty only.

The procedure is summarized in Algorithm 2, where S is the soft-thresholding operator which is applied component-wise to a vector. In the case of LASSO:

$$(S_{\frac{\tau}{\sigma}}(\beta))_j = (|\hat{\beta}_j| - \frac{\tau}{\sigma})_+\hat{\beta}_j \tag{4.7}$$

in case of group LASSO:

$$(S_{\frac{\tau}{\sigma}}(\beta))_j = (||\hat{\beta}_{G_r}|| - \frac{\tau}{\sigma})_+\hat{\beta}_j. \tag{4.8}$$

Instead, σ represents the step size of the iteration and it affects convergence rate. Following [MRS$^+$10] we set it as proportional to the highest eigenvalue of matrix $X^\top X$ for optimal convergence.

The Algorithm 2 can be easily parallelized to exploit modern multi-core processors. Further optimizations are available [MRS+10] and include the computation of multiple solutions for different values of τ ordering the minimization inversely w.r.t. the regularization parameter. The previous solution can be used as initialization of the following optimization: this can be advantageous as sparser solution can be computed with a reduced number of iterations and, if the space of τ is sampled densely, the two subsequent solutions can expected to be close.

An interesting property for further optimizing the minimization of LASSO (an extension to group LASSO is possible) is that, given a training set and a regularization parameter, it is possible to remove a priori a subset of features that are guaranteed to be null in the solution [EGVR11], resulting a smaller data matrix and a more efficient minimization. The latter result can be extended with a heuristic to discard a higher number of features [TBF+12], however the correctness in not guaranteed and it may be needed to repeat the optimization.

4.4 Pedestrian detection with Group LASSO

Sparsity enforcing regularization techniques for feature selection are seldom used in computer vision, although they are very popular in other applications as computational biology and bioinformatics (see, for instance, [MMTV09]). Destrero et al. [DMOA09] adopted LASSO to extract face features for face detection, starting from an over-complete set of rectangle features. Only recently group LASSO has been chosen to select groups of features for a very simple image understanding problem [WYZ10] and to select groups of data for object recognition from depth information [LBRF10].

Our objective is to test a framework based on regularized structured sparsity in the context of object detection. Since most of the successfully descriptions for pedestrian detection (e.g. [DT05, TPM08b]) have a structure in the description that is induced both from logic and computational reasons, we are interested in studying a feature selection framework able to select groups rather than single values. If the groups are induced by the structure of the computation (i.e. all the values of the group have to be computed together) selecting them rather then single values makes possible to choose between multiple correlated features to minimize the number of required groups.

To test the algorithm we have first adopted variable size HOG, however the same setting can be combined with other features. Two further examples of descriptions are given in Sec. 4.4.3. Our work represents a further proof that HOG features are a valuable representation for the class of objects under analysis, in particular if variable size blocks are computed; indeed, the approach we propose is closely related to original formulation of the HOG but, similarly to the extensions based on cascades of boosted classifiers [ZAYC06], the choice of the support regions for the HOG is driven from data instead than being chosen a priori.

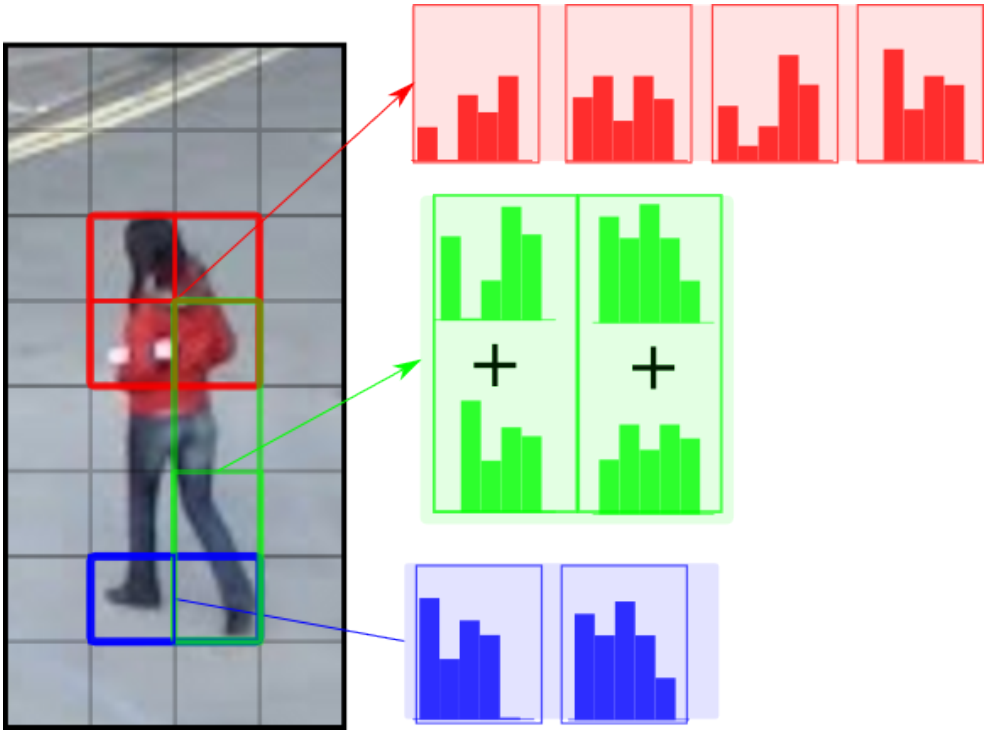


Figure 4.2: Each block of the variable size HOG is composed by the combinations of (1×2) (blue), (2×1) (green), or (2×2) (red) cells. The cells can be composed by multiple cells (2 in the green example), and can be easily computed summing together the associated histograms.

4.4.1 Variable-size HOG

The method we propose is based on learning an appropriate representation for pedestrians from data, starting off from a high dimensional dictionary of atoms obtained from variable size HOG (see Figure 4.2) descriptors and selecting the most meaningful representation with group LASSO.

HOG blocks selection

We now discuss how we describe the image content. We adopt a variable size version of HOG features [ZAYC06]: we consider a detection patch of 64×128 pixels, rectangular cells whose sides range from 8 to 128 pixels and rectangular blocks formed by (1×2), (2×1), or (2×2) cells. The shift between multiple blocks is equal to the minimum size of a cell, so that it is possible to extract efficiently the description from multiple windows.

Within each block we refer to the original HOG definition, where a block is represented as a concatenation of 9-bin HOG, each of which is associated to a cell, and then normalized (see Sec.

3.3.4). From the original formulation we only discard the Gaussian weighting that would prevent the computational optimization described below.

Note that in the case of non normalized histograms bigger cells can be obtained from linear combinations of smaller ones (this means that adding bigger blocks does not add information). However in the HOG formulation each block is normalized (with a non linear transformation) and hence different sub-parts of a block acquire different information that cannot be retrieved with a linear combination of smaller blocks.

In our setting the generalized linear problem described in Eq. 4.1 can be specialized to select meaningful HOG blocks as follows: Φ is the HOG matrix computed on the training set, each row of matrix Φ represents a training datum described by all possible blocks within the chosen size range. Each column Φ_j of the matrix is associated to a feature, that is a bin of a specific HOG. $\boldsymbol{\beta}$ is the vector of unknown weights to be estimated while, since we are considering a binary classification problem (pedestrian vs non pedestrian), $\boldsymbol{y}$ takes values in $\{-1, 1\}$.

As a final remark we observe that, in spite of the spatial overlap of the computed groups, because of the normalization step applied within each block, there is no numerical overlap among the different groups. Otherwise a variant of the group LASSO taking into consideration possible overlaps would have been needed [LOV09].

Pedestrian detection and efficient HOG computation

At run time a new image patch w is represented with respect to the selected groups of features only and classified with a linear SVM, tuned by K-fold cross validation. Notice that it would have been possible to classify directly via group LASSO, but the performances obtained with SVMs are slightly higher because of the well known shrinkage effect of the weights of LASSO methods [CT07].

Even if we can expect to achieve better results with a non linear SVM (similarly to the HOG [DT05]) we chosen a linear kernel due to its reduced computational cost: to classify a patch a non linear SVM has to compute multiple scalar products (one for each *support vector* whose number is bounded by the cardinality of the training points), in the linear case only a scalar product is sufficient (see [HTFF01] for details).

To detect pedestrians it is usually needed to compute the feature vector of all the possible image windows changing their position in the image. Moreover to detect pedestrian of different size the image is resized and the scanning is repeated.

To compute efficiently the variable size HOG representation, we first compute an intermediate representation of the image obtained by pre-computing the histograms over 8×8 image patches. This allows us to compute efficiently cells bigger than 8×8 simply by summing up smaller cells (temporary results can be stored to avoid redundant computations), and blocks by concatenating

their cells and normalizing the obtained histogram. At run time the intermediate representation provides a remarkable computational speed up, in particular if pedestrian detection is performed with a search step of 8 pixels. Moreover it is possible to obtain finer steps by computing shifted grids of cells.

This choice is alternative to the implementation based on multiple integral images (one for each histogram bin) [ZAYC06]: each integral images is computed over an image that contains the magnitude of the gradient only for pixels associated to the same bin. It is hence possible to compute the sum of the magnitude of the pixels with a given orientation by accessing four times the associated integral image.

The latter method can be advantageous with big and non overlapped blocks, however it needs computing and storing 9 integral images (one per bin) and accessing each of them four times to compute a single cell (144 accesses to compute each block), being sub optimal for smaller blocks.

4.4.2 Experiments

In this section we carry out a set of experiments to evaluate the ability of group LASSO to extract a meaningful and compact variable-size HOG representation. Our focus is the validation of the feature selection framework, hence the experiments aim to compare the solutions based on group LASSO with the ones obtained with more common feature selection algorithms (e.g. Adaboost, LASSO) and with descriptions fixed a priori (e.g. the standard HOG grid).

Most of the experiments have been carried out on the INRIA dataset (see Sec. A.1), which is still the most popular dataset for pedestrian detection. We also report results on the more recent Caltech pedestrian dataset (see Sec. A.1) and conclude with a qualitative analysis of the detection results obtained on PETS06 and PETS09 (see Sec. A.2).

The experimental protocol we adopt is based on false positive rates

$$FPR = \frac{\text{FalsePos}}{\text{TrueNeg+FalsePos}} \tag{4.9}$$

and hit rate

$$TPR = 1 - \frac{\text{FalseNeg}}{\text{FalseNeg+TruePos}} \tag{4.10}$$

per window. Although this protocol is less popular than the analysis per image, we feel that it is a better way of judging the classification procedure performed at window level which is the main goal of our study. Indeed, the results obtained with this protocol are not biased by the heuristics usually implemented for object detection, such as non maxima suppression.

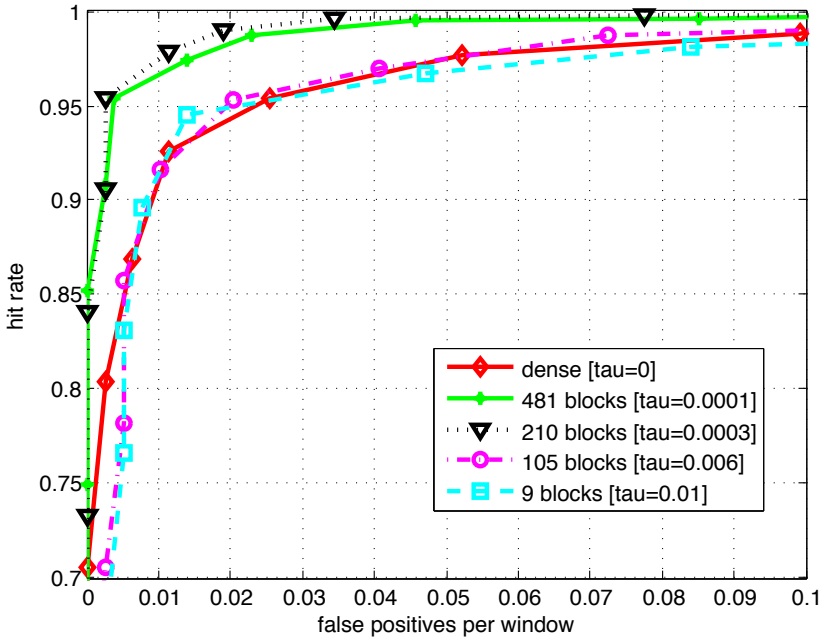


Figure 4.3: ROC curves obtained by different choices of the sparsity parameter τ with group LASSO, on the INRIA dataset.

As for the choice of the algorithms to be used for a comparative analysis with the state of the art we choose a direct comparison with original HOG and its boosted variant with the same representation. Also, for the INRIA dataset, we compare our results with the ones published in [SKHD09], highlighting a lower performance of our method to the benefit of a lower computational complexity.

The size of the dictionary used in the experiments is D =77040, while the number of groups B is equal to the number of blocks, 3608, and since not all blocks are formed by the same number of cells each block size differs. The parameter τ controlling the sparsity of the solution is chosen by K-fold cross validation (K=10) to be the one producing the smallest average classification error on the validation set.

INRIA dataset. The first set of experiments have been completed on the popular INRIA pedestrian dataset. We have trained the classifier on the INRIA pedestrian (see Appendix A) training set and selected the parameters by applying a 10-fold cross validation procedure. As a test set we use the original INRIA set. We perform feature selection with group LASSO and various choices of the sparsification parameter τ and, following a model selection procedure based on K-fold cross validation, we choose $\tau = 0.0003$ corresponding to 210 blocks.

Figure 4.3 reports a comparison in terms of ROC curves of different classifiers obtained by

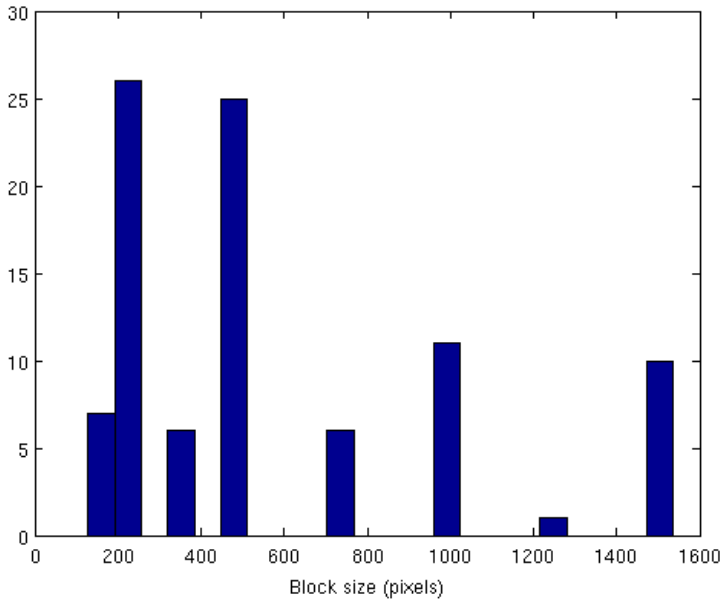


Figure 4.4: Size distributions in pixels of the 210 selected groups with group LASSO and $\tau = 0.003$ on the INRIA dataset. See text for details.

varying the sparsity parameter τ. Notice that, even if the representation needs twice as the number of fixed size blocks computed in the original HOG work, thanks to the implementation of HOG described in Section 3, the computational cost does not increase critically.

Figure 4.4 shows the distribution of the 210 selected blocks with respect to their size in pixels, where many selected blocks correspond to the fixed HOG size adopted for pedestrian detection (256 pixels), but it is also noticeable the presence of bigger blocks that should capture more global features within the pedestrian appearance. Figure 4.5 speaks in favour of the importance of additional features from different scales, showing a superior performance of the variable size combination of HOG for different sparsity levels (104, 210, 387 blocks) with respect to the fixed-size case on the same data. As expected the results show that adding redundant features does not add useful information and decreases the performance of the learning algorithm due to the increased dimensionality of the feature space.

Figure 4.6 shows the results obtained with a feature selection performed with Adaboost [ZAYC06] at different sparsity levels, using as weak classifier the Weighted Fisher Linear Discriminant Analysis (WLDA) [PSZ08]: in this case the optimal size of selected features is 100, but the results are inferior to the ones obtained with group LASSO and a similar sparsity level (105 blocks). The figure also reports the results obtained with very sparse solutions (9 blocks) selected both with Adaboost and group LASSO. We notice that, that group LASSO achieves better

performances than Adaboost even for small sets of features, suggesting that it would be possible to compute effective HOG cascades with group LASSO feature selection.

The drawback is that the computational cost of group LASSO is higher than the simple Adaboost procedure, especially if we aim to select a few features. However, in a case where the training is done only once, it may be an acceptable tradeoff to increase the accuracy or the final detection speed.

The good performance of a simple classifier with 9 windows is also confirmed by the results reported in Figure 4.7: the figure reports, in the line above, the bounding boxes selected as pedestrian with the sparse classifier formed by 9 HOG. Interestingly, this sparse classifier could effectively act as a fast filter to discard the easiest negative examples in a multiple layer architecture such as a cascade. Figure 4.7 (below) shows the result obtained with our classifier trained on 210 HOG (no maxima suppression has not been applied to highlight the performances of the method without post-processing heuristics).

Examples of difficult false positives and false negatives obtained with the classifier based on 210 blocks on the INRIA test set are reported in Figure 4.8.

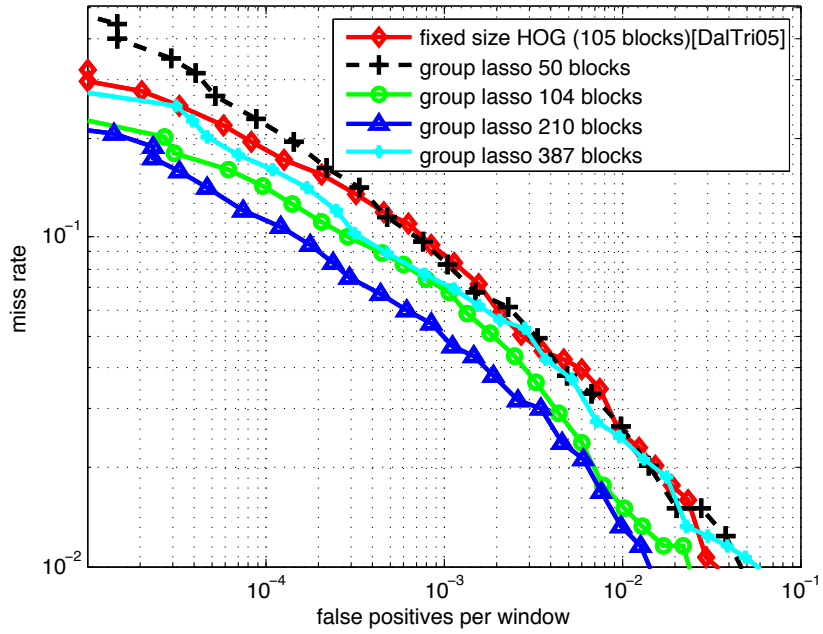


Figure 4.5: Comparison of the fixed-size HOG (105 blocks) with the group LASSO selection on a variable size HOG dictionary (at different sparsity levels) on the INRIA dataset. The results speak in favour of the latter.

To get a visual impression on the features selected by the method, Figure 4.9 (center) shows

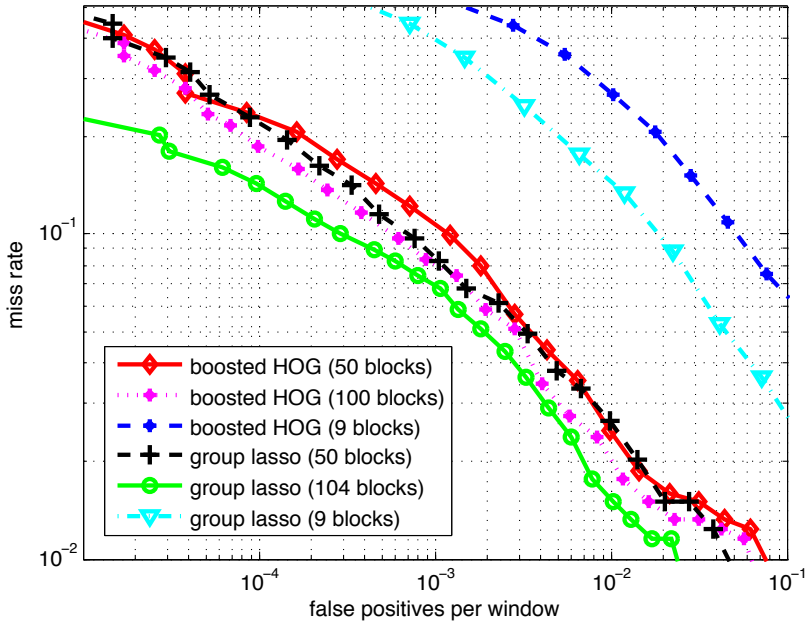


Figure 4.6: Comparison of group LASSO and Adaboost feature selection at different sparsity levels. See text for details.

the support of the 9 groups selected by group LASSO when choosing a value for τ (0.01) that strongly sparsifies the solution. The selected groups correspond quite nicely to meaningful and more stable parts of the pedestrian shape (head and feet) and are of a relatively small size. Also, they appear to capture the symmetry of the training set, formed by mirrored images. It seems that the type of groups selected by the two approaches of group LASSO (in a single classifier) and Adaboost (in a cascade of classifiers) are quite different; Zhu et al. [ZAYC06] notice that in their boosted cascade of HOG classifiers most selected groups are quite big and this is confirmed by our experiments: Figure 4.9 (left) shows the four groups forming the first layer of the cascade, whose size is higher and the spatial support less intuitive. The impression is that while Adaboost is based on finding single groups with a good discriminative power, group LASSO prefers a combination of groups and the combination of relatively small groups may be more effective than a single bigger group.

On the INRIA dataset Schwartz et al. [SKHD09] reports a miss rate below 10^{-1} for 10^{-5} false positives per window, which appears to be to date the best performing pedestrian detector. With respect to this method our approach reports lower performances (refer to Fig 4.5). The algorithm is based on a very rich initial representation, that includes HOG, co-occurrence matrices and colour frequencies. Then, instead than performing feature selection the authors apply a dimensionality reduction approach that controls the space complexity, and then address time complex-

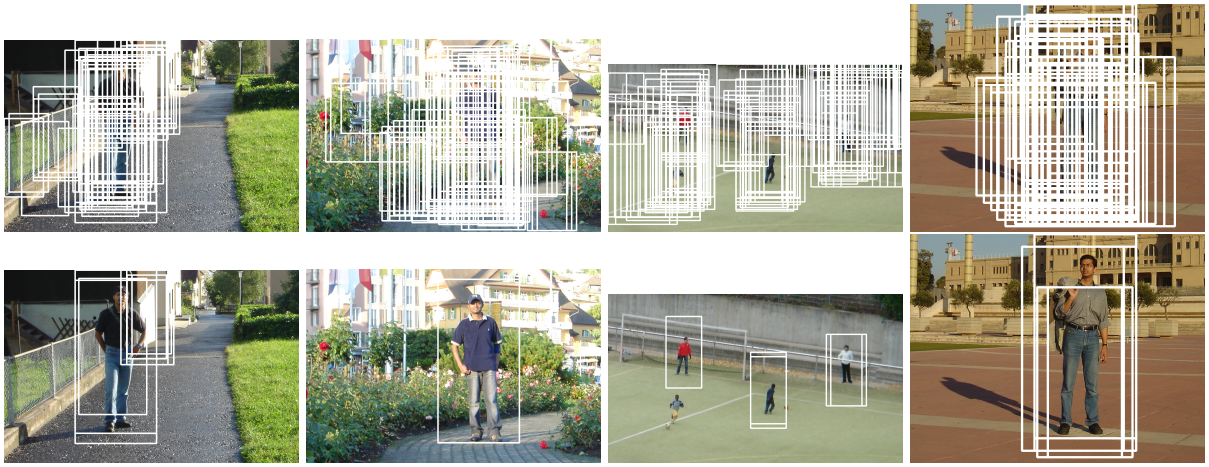


Figure 4.7: Sample images from the INRIA test set comparing the results obtained with group LASSO on a variable size HOG dictionary, with a sparse solution made of 9 groups (above) and the chosen best performing solution made of 210 groups (below) without non maxima suppression.

ity with a two layers classifier. However the method is computationally intensive as each window description requires between 3500 and 170820 values (depending on the level) and the results reported suggest that it is not suitable for real time systems.

We conclude our experiments on the INRIA dataset by reporting a comparison between group LASSO and a feature selection with conventional LASSO, that does not take into account any group structure. Conventional LASSO allows us to achieve a slightly higher performance (an equal error rate of 98.8% against the 98.5% obtained with group LASSO) but at the expense of a more complex representation. Indeed, in terms of single features (or bins) the data matrix has over 77000 columns: LASSO produces a sparser representation (5292 non-zero entries of vector β, against the 6894 obtained with group LASSO), but they are distributed among various blocks which need to be computed entirely to obtain the value of a single bin. Thus, at run time, with LASSO we would end up computing 2046 blocks, versus the 210 selected by group LASSO.

Caltech dataset. A second set of experiments have been done on the Caltech dataset: we use the subset of the training set labelled as "reasonable" (50 pixels or taller pedestrians, unoccluded or occluded up to 35%) subsampling data one every 5 frames to limit the amount of patches coming from the same dynamic event (see Figure 4.11 for an example of how similar data from the same video are, even after subsampling) obtaining a set of 2227 positive images. The dataset is splitted in two different ways:

- Split it into training, validation, test of cardinality 1142, 588, 497 respectively. Use training and validation to choose the best representation of data and classifier with group LASSO;

Figure 4.8: Sample of errors on the INRIA dataset: false negatives (above) and false positives (below). Samples have been chosen as the hardest to classify for the SVM; interestingly false negatives include many reflected pairs.

evaluate the performance on the test set.

- Use the whole set for testing, and keep the models trained on the INRIA dataset (similar to what suggested Dollar et al. [DWSP09b], although there is no guarantee the two datasets may be seen as different realizations of the same probability distribution).

Figure 4.10 shows the results obtained on images from the Caltech dataset. Two group LASSO selections are compared: one carried out on the Caltech training set, another on the INRIA dataset. The latter produces the best results. Indeed, the groups selected on the Caltech dataset appear to be less informative even to a visual inspection (see Fig 4.9, right), possibly due to the training set variability, too small to capture the complexity of the class and to a lower level of details and size of the images. The selected groups are more generic, although they maintain a symmetric layout and are mostly localized on head and feet. A comparison with other methods on the same set of data reported on the Caltech webpage (performances on the training set) is not straightforward, since the adopted metric is different, but the results obtained with HOG appear in line with the ones obtained in our experiments.

PETS06 and PETS09 datasets. Finally the pedestrian detector trained on the INRIA dataset has been applied for a frame-by-frame analysis on the benchmark datasets PETS06 (including indoor environments) and PETS09 (outdoor environments with crowds of different densities). Lacking an appropriate ground truth we relied on manual annotation and a qualitative evaluation of the results. Figure 4.12 reports sample results from different video sequences.

Figure 4.9: Spatial support of the first groups selected by different classifiers. **Left:** the first cascade level computed with Adaboost on the INRIA dataset; **center:** the 9 groups selected by group LASSO on the Inria dataset ($\tau = 0.01$); **right:** the 9 features selected by group LASSO on Caltech ($\tau = 0.01$). See text for details.

4.4.3 Adopting other dictionaries

The adopted feature selection approach could be combined with a variety of other feature vectors or combinations of them. In this section we discuss the choice of a different dictionary of features: *covariance features* [TPM08b] and a multi-scale generalization of the variable size HOG.

Covariance features

Covariance features are an over-complete set of features derived computing the covariance within a region of interest between 8 statistics (pixel location, first order partial derivatives of the intensity, intensity magnitude, edge orientation, second-order partial derivatives of the intensity) and normalizing them w.r.t the considered image window. A 8×8 covariance matrix is then associated to each image region.

It should be observed that the computational cost of computing covariance features is quite high. The number of operations needed to compute them may be greatly reduced by introducing integral images, to the price of having to pre-compute the integral images of each feature and of the product of each possible couple of features. This procedure allows us to compute each covariance matrix in constant time independently on the region of interest, however it requires to compute $d + (d \times d)/2$ integral images, where d is the number of adopted statistics (d=8 in the original work).

The computation of the covariance matrix elements through integral images and their normaliza-

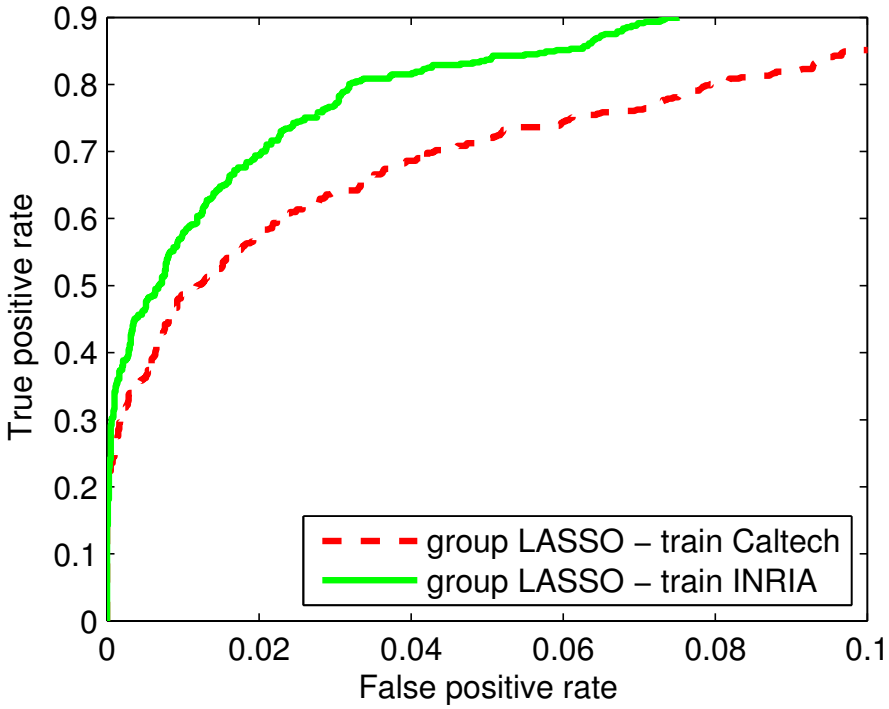


Figure 4.10: Comparison of the variable size HOG- group LASSO solution trained on the INRIA dataset and on the Caltech dataset. All tests have been performed on Caltech. See text for details.

tion allow us to extract efficiently all the value of a matrix together. Moreover considering they produce a dense and rather big description of each image patch (in our experiments we associate a feature vector of 8910 blocks by considering various subregions of each 64×128 patch, in previous works even bigger descriptions have been obtained), they also can be chosen as an initial dictionary on top of which to perform group selection (here the groups are the covariance matrices).

Previous works devoted to diversifying the description produced by covariance features have been presented: Paisitkriangkrai et al. [PSZ08] proposed an Adaboost cascade on a vectorized version of covariance matrices, which is then projected on a 1D line via weighted linear discriminant analysis obtaining results comparable to the original, with improved performances.

In our setting, the sparse description based on covariance features do not achieve comparable performance to HOG. Figure 4.13 compares the results obtained on the INRIA dataset, showing a better behaviour of the former. Our results highlight a higher sensitivity of covariance features to the effect of false positives, possibly in previous works [TPM08b, PSZ08] this behaviour was attenuated by the influence of the cascade.

Figure 4.11: Samples from the Caltech dataset: one image every 5 frames out of a sequence is shown.

Multiple scales features

Previous works (e.g. [HEH06, PRF10]) showed that the classification performances can improve using techniques that take into account the scale of the observed features.

A classical way to compute multi-scale detection is to use a fixed size detection window that scans the same image resized at multiple scales. For example in Figure 4.14 we have an image with two people of different size: to detect both the pedestrians it is needed to resize the image, so that the fixed-size red window can capture both the pedestrians at the right size.

Since the same image patch would be considered at different scales in the detection phase, and thus its description needs to be computed in any case, our aim is to consider it explicitly in the representation phase.

In this section we consider variable size HOG, as its structure and the computational method described in Sec. 4.4.1 fits well with the multi-scale approach, however different descriptors may be suitable.

While a multi-scale approach to the detection is not novel (the method proposed by Park et al. [PRF10] is one of the most effective examples), we focus on a simple model that increases only marginally the computational cost of the single scale approach.

Given an input image we first resize it to multiple sizes, so that scanning the set with a window of fixed size we can detect objects in the desired size range. To this aim we compute a set of Gaussian Pyramids computed starting from a set of images whose size is resized with different factors $(0.5, 1)$ (an higher factor sample more densely the scale space).

In the second step, for each image, we proceed as with the single scale algorithm computing the cells of the HOG description. Note that, once the grid is computed, the associated image is no more required.

An example of the result of these two steps is shown in Figure 4.14, where we consider a simple

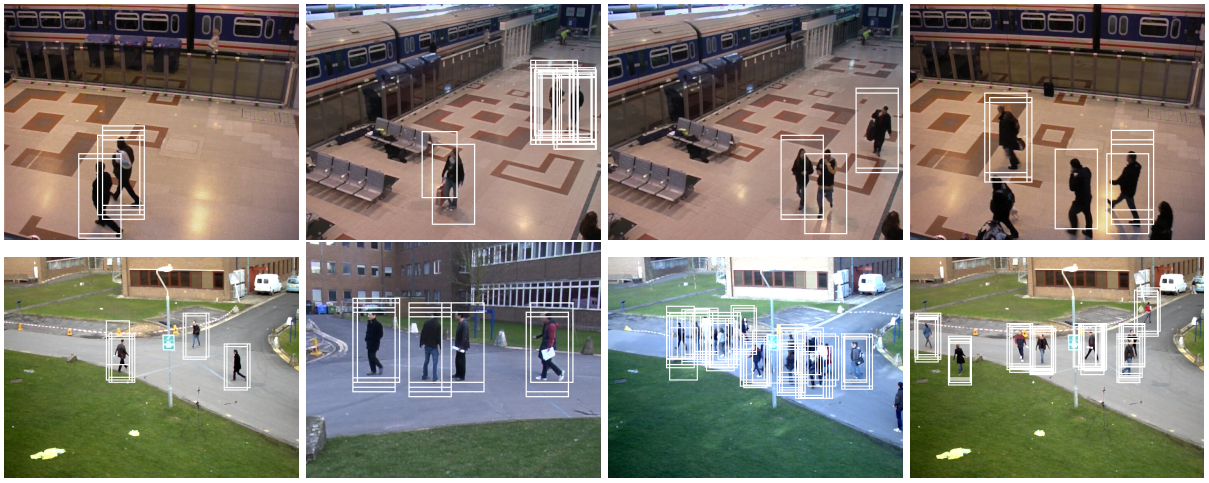


Figure 4.12: Detection results on video frames from the PETS06 (above) and PETS09 (below) datasets without non maxima suppression.

case with only two scales that are needed to be able to detect both the people with the fixed size continuous red window. If we compute the grids of all the scales before scanning the image, while we compute the description of the windows of the upper image we can access at no additional cost to the cells computed on both the scale levels to enrich the description (e.g. the cells within the continuous red bounding box in the upper image and the dotted red bounding box in the lower image).

The idea is to consider a dictionary that is composed by the features computed at different scales. In this setting the feature selection framework presented in Sec. 4.3 is crucial as the dictionary of possible blocks increases with the number of considered scales.

If we fix to M the number of scales to describe a single image window ($M = 2$ in the example of Figure 4.14), the additional computational cost between a single scale and a multiple scale descriptor is the computation of $M - 1$ images (and their HOG cells) at the smallest scales. This is needed as to detect the biggest considered object we have to consider $M-1$ further scale levels to fill the description: for example in Figure 4.14 to describe the bigger pedestrian we have to compute the description on another image whose size is the half of the lower image. Despite that, the increase in the computational cost is still marginal as those $M - 1$ levels are smaller than the smallest image needed for the detection.

A further advantage of such structure is that it is straightforward to extend the multi-scale description to work with pedestrian smaller than the detection window: usually in a single scale setting to detect smaller objects the image is up-scaled interpolating the information, in the multiscale setting instead it is natural to select only the subset of information available (scales) avoiding an image interpolation step that does not add any information and increases the computational cost.

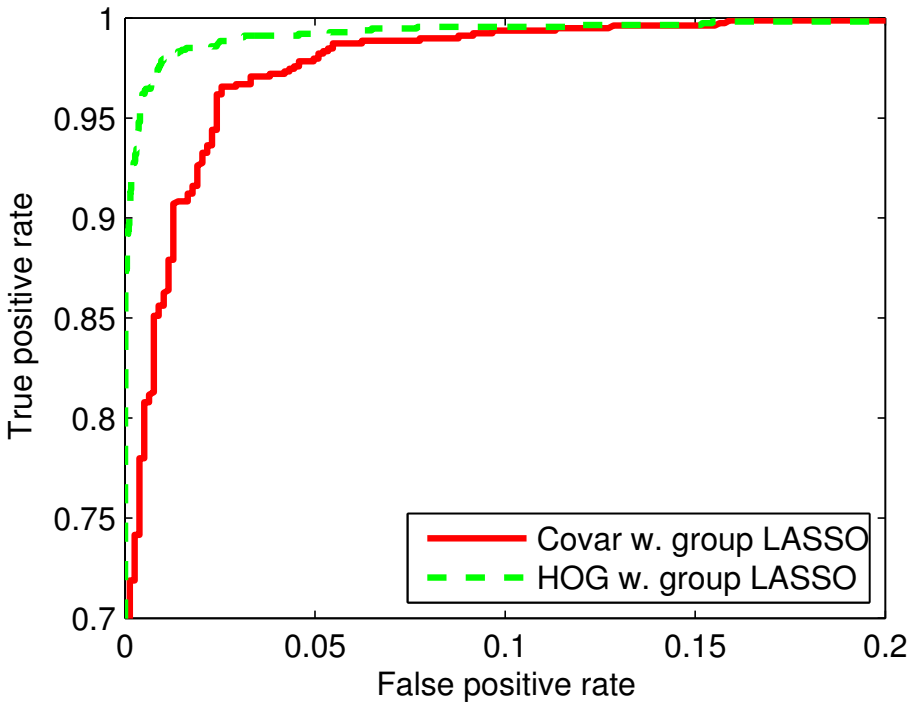


Figure 4.13: Performance of a classifier trained on the HOG dictionary and the best one trained on covariance features dictionary. See text for details.

4.5 Discussion

In this chapter we have presented our study on a framework based on regularized and structured feature selection to find a set of features meaningful for pedestrian detection (or, more in general objects detection) starting from an over-complete set of candidate features.

Regularized feature selection methods are not widely applied in computer vision, to the advantage of greedy methods based on boosting. However our results suggest that such methods are promising, and, if needed, are able to include a priori knowledge or requirements on the problem (e.g. the structure of the solution with group LASSO).

In this chapter as an example we have proposed an algorithm that selects meaningful features from an over-complete set of variable size HOG using group LASSO, whose results have been published in [ZO11]. The regularized approach achieves better performance w.r.t. the more known Adaboost on the same description. Moreover we have designed a setting able to merge in the same framework information from multiple scales of a given descriptor that may improve the accuracy with a negligible additional computational cost.

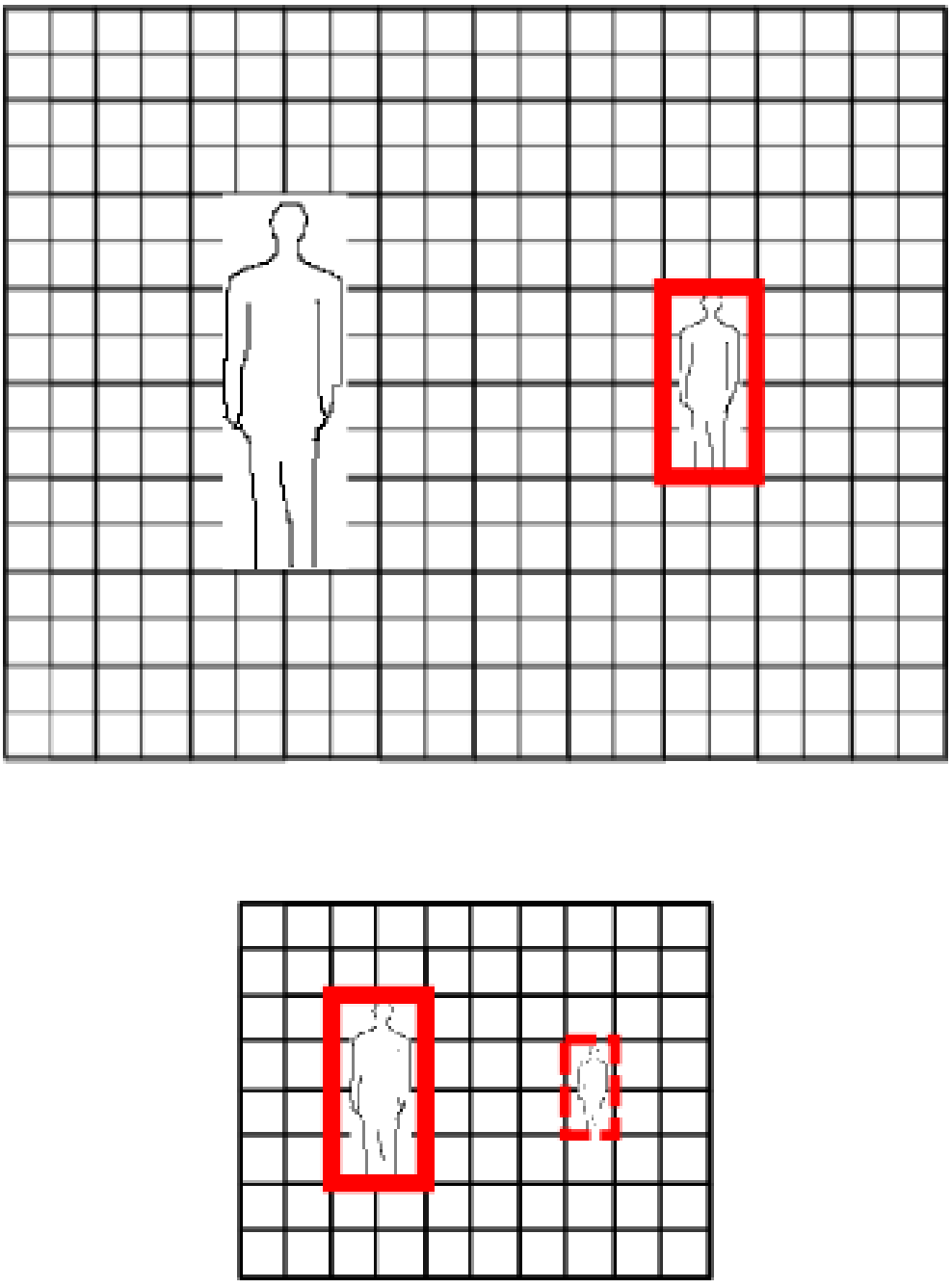


Figure 4.14: The information (e.g. the grid HOG histograms) computed at different scales to detect people of different size using a fixed window (continuous red) can be used to enrich the description. For example it is possible to merge data from the continuous and dotted red bounding boxes of the pedestrian on the right, as the description of the dotted bounding box has to be computed to detect the pedestrian on the left. See text for details.

The techniques presented in this section combined in a detection algorithm, can detect people in real-time on modern hardware. Both the accuracy and the performances of the algorithm can be improved by combining different techniques: for example a change detection algorithm (see Sec. 3.3.2) can select the parts of the image where there is something different from the background and the detector can restrict the search to this area to select the parts that corresponds to the objects of interest (e.g. people).

Further study should include extensive tests of the use of the framework together with multiscale features and the integration of the computed descriptions in a full object detection pipeline that could not be completed during this research.

Chapter 5

Crowd estimation

The objective of this chapter is to address the problem of real-time people counting from videos and to propose a simple but effective solution that is characterized by a low computational cost. First Sec. 5.1 introduces the problem and defines our objectives, Sec. 5.2 overviews the literature on people counting, in Sec. 5.2 we describe the approach that we have developed and in Sec. 5.6 we provide a detailed evaluation of the performances in different scenarios and we compare our results with the state of the art.

5.1 Introduction

The problem of estimating the number of people or, more generally, the density of a crowd in images and videos is of interest for a broad range of applications. It may be useful in a security framework to detect an unusual or potentially dangerous crowding level; in a commercial context to model the number of people that visit a shop as a function of time and other variables; to estimate the number of people in a demonstration; to plan public transports; as an input for a higher level video surveillance algorithm.

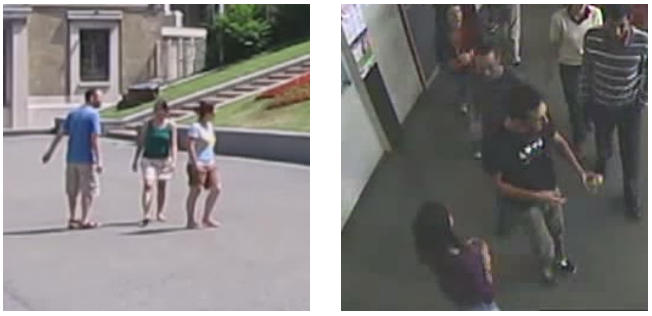
While computer vision is not the only possible tool to address the problem of people counting (see for example [HKM+98, GBTTO12]) it is an appealing approach as it can exploit already existing video surveillance infrastructures.

For a human being, distinguishing between dense crowds, sparse crowds or small groups of people is an easy task, however this is strongly related to the ability of localizing partially occluded people, estimating their size from partial observations and segmenting the crowd. From a computer vision point of view none of those tasks can be considered solved as the segmentation can have different solutions and hence is often considered an ill-posed problem and the detection algorithms fail in presence of occlusions and with low resolution images (for a more

(a)

(b)

(c)



(d)

Figure 5.1: An arbitrary scenario may have: (a) very different crowding levels; (b) different intra person distances; (c) different complexity of the background; (d) different viewpoints.

detailed analysis of detection see Chapter 4). The main source of complexity is the high degree of variability due to extrinsic factors to the crowding levels as for example the distance of the camera, the complexity of the background, distortions due to perspective effects and the presence of camera motion.

We propose a simple model-free algorithm based on a geometrical analysis of the objects in the scene that exploits temporal information from a video to compute a reliable estimate. Our objective is to show that, despite the simple approach, it achieves performances comparable or better to those reported in the state of the art algorithms in a broad range of scenarios.

5.2 State of the art

In the literature have been proposed a broad range of solutions for crowd density estimation and people counting that can be classified in three different groups:

- Based on foreground map analysis.
- Based on people detection.
- Based on texture analysis.

Each class of methods has its strong points that make it suited for specific scenarios. In principle a method based on the analysis of the foreground map is suited to work in densely crowded scenario with frequent occlusions, but it needs a precise segmentation and it may be mislead by static people. A method based on people detection can be very accurate with people close to the camera, not too low resolution images and with none or small occlusions. Instead, a method based on texture analysis will be able to distinguish between highly crowded situations, but it may be less precise in people counting.

In Figure 5.1 are shown examples of different types of scenarios that a people counting system may need to solve: the strong difference between them suggests that, depending on the particular scenario, different approaches may be needed.

Another possible characterization of the algorithms is on the requirements or on the assumptions on the acquisition system. Examples are:

- Planar scenario.
- Fixed cameras.
- Calibrated cameras.

- Specific orientations of cameras.
- Presence of an on-site training step.

Assumptions on the planarity of the scene, on the motion of the cameras or on its calibration are common in all the classes of methods but the one based on the detection. For example the planarity of the scenario, together with the use of fixed and calibrated cameras are needed to give a uniform meaning to the size and to the texture of objects over different points on the image. Foreground segmentation algorithms usually assume a static camera to apply background modelling and subtraction techniques. Moreover all those constraints are useful also for detection methods as they allow to shrink the computational requirements and to limit the false positive rate not considering hypothesis of positions and sizes incompatible with a pedestrian.

An example of specific requirement on the orientation of the camera is to assume it to be perpendicular to the ground [VTH06, CCC06, HC03, BCS07]. In this case the geometrical model of each person is very simple, to the disadvantage that the camera needs to be positioned ad-hoc, and this is not practical for usual video surveillance systems. In other cases the camera can be assumed parallel to the ground [MLHT04], but this requirement is less strict and the correct results can be approximated in a wider range of configurations.

An on-site training step can be exploited with general methods to improve accuracy (e.g. on site bootstrap can improve detection performances), in other cases (e.g. [SMS96]) it is a key requirement as the algorithm needs to learn the specific scenario before being able to analyse it.

Some works in the literature have proposed to combine different features (e.g. texture and foreground segmentation) usually given in input of a machine learning algorithm. This kind of methods can have two different objectives: to improve the performances on a specific scenario, or to give a more general algorithm able to work with a broader range of different scenarios. However, to guarantee the ability to cope with very different scenarios, a non linear model and a big amount of training data may be needed. Two examples that make the learning of a generic solution challenging are: (i) the texture background, that may not be plain and difficult to distinguish from the one induced by a crowd (ii) the physical boundaries that may constraint crowd to assume scene specific configurations (e.g. the shape of groups, different intra person distances). To our knowledge there is no work in the literature that shows explicitly the ability of a people counting algorithm in the generalization of a learned solution to different scenarios.

In the following parts of this chapter we give a more detailed overview of the methods from the literature depending on the classification of their approach. In Figure 5.2 it is shown a sketch of a camera system together with the notation used in this chapter.

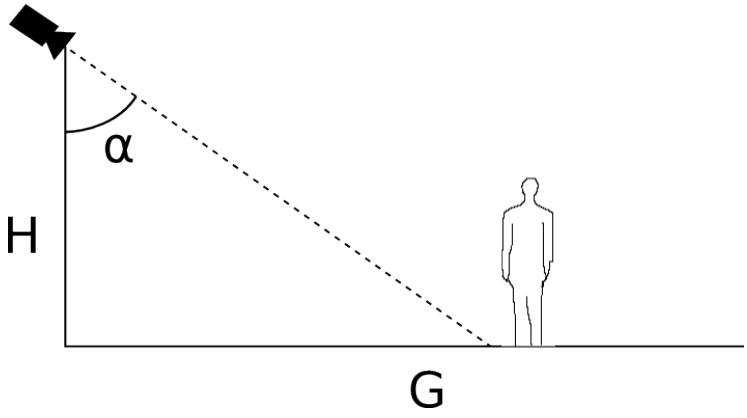


Figure 5.2: Scheme of a general camera system with the associated notation. Depending on the method there might be some requirements on the camera height H or on the angle α. Most of the algorithms in the literature assume that the observed scene is planar.

Methods based on foreground extraction

In this class of methods the input is the foreground segmentation mask and information on the geometry of the scene (usually assumed planar). The straightforward idea is to assume that the size of each blob in the segmentation map is correlated to the number of people that have generated it and thus can be exploited to estimate their number.

Since the apparent size of a person is related to its distance from the camera, it is not uniform all over the image. A perspective model is needed to compute a uniform relation between the position on the image and the size of the observed object. This can be achieved observing a person in different positions or exploiting the vanishing points.

If no pedestrian is occluded, the sum of the foreground pixel size, weighted to compensate the projective distortion, can be linearly correlated to the number of people in the scene and hence used to count them.

A problem of the perspective normalization is that the size to be assigned to each pixel is ambiguous and it depends on the position of the feet of the pedestrian that have generated it. Analysing the problem from a theoretical point of view [MLHT04] it can be shown that, with a single non occluded person, the error induced considering for all its pixel the scale of the lowest point (usually the feet), the error induced is a constant independent on the position in the image. The latter result does not hold in presence of groups of people and with occlusions, but it is possible to apply different heuristic to correct the distortion [CHH06] using: (i) a constant factor for all the pixels of a blob [MLHT04] (ii) a factor that grows linearly with the height of the blob (iii) the first strategy mixed with the second one only for blobs over a specified size (iv) as the third strategy but using the linear weighting only for the upper part of the blob (v) trying to fill the maximum

number of bounding boxes in the blob defining a priori the minimum space between two people as a function of the width of the bounding box. The results show that the first technique gives the best results. A similar idea has been extended combining information from KLT features [TK91] and using an Expectation-Maximization (EM) algorithm [DLR77] to localize people inside the foreground map [HP11].

Further optimizations exploit the temporal coherence of the number of people in close instants to refine estimates based on the analysis of the blob w.r.t. the size of a virtual person in the same position [HP11].

Assuming that the size of pixels is corrected w.r.t. perspective distortion, Bunster et al. [GBTTO12] compared different techniques to compute a mapping between the size of the segmented foreground blobs and the number of people (Linear Regression Model, Probabilistic Neural Networks, K-Nearest Neighbours). Moreover in the paper are presented the results of four methods for object detection (HOG, Discriminatively Trained Part Based Models detection [FGMR10], Haar based object detector [VJ01]). Their results show that the best results can be obtained by using linear models with in input the perspective corrected size of the blob. The performances of the listed object detection methods are sensibly worse and have a counting precision between 2 and 4 times worse than the perspective model.

Kilambi et al. [KRJ$^+$08, FSM$^+$09] propose to compute the space on the ground starting from the change detection map and to assume that it is linearly correlated to the number of people. To estimate the area occupied on the ground they compute a set of cylinders in the world whose projection on the image has the maximum overlap with the change detection map. The area of all the bases of the cylinders is the area occupied by the observed people. To initialize the parameters of each cylinder and to fix their number, given a segmented object the shape B_T in the world positioned parallel to the ground at a fixed height T whose projection corresponds to the segmentation and a second one (B_0) positioned at height 0 are intersected (see Fig 5.3). Each connected component extracted from the map resulting from the intersection is associated to a cylinder.

While the cylindrical model is needed to compute the height of each person in the scene, it introduces the strong assumption that the occupation on the ground of each group of people can be modelled with an ellipse. This is reasonable with small groups of people, but is not valid in general, for example for a large number of people that walk in a constrained environment (e.g. Figure 5.4).

Independently on the model adopted, with all the segmentation based method presented above, the numerical stability of the algorithm decreases with distant people observed with values of α that approach 0 due to occlusions and to the high errors of the estimate that can be induced by a small number of wrong segmented pixels.

If multiple calibrated cameras are available it is possible to count people by projecting on the ground the change detection maps from all the cameras and merging the results to check points

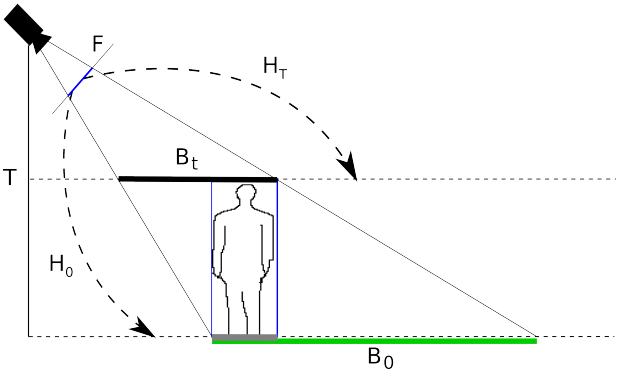


Figure 5.3: Given the image of an object is possible to infer its occupation on the floor (grey) intersecting the blobs needed to generate the same image if the object is placed horizontally on the floor B_0 (green) and at its maximum height B_T (black)

Figure 5.4: Not all the shapes of a general group of people can be represented as a cylinder with an elliptical base. Similar or more critical configurations can be often found in real world scenarios.

on the ground plane that are occupied from all the views [YGBG03]. The ambiguity due to occlusions is recovered by building a history tree and by searching on it. Such an approach is feasible in some constrained situations and might be very accurate with a small number of people, but has the drawback of requiring a relatively high number of overlapped, calibrated and synchronized cameras. Moreover, as other methods based on the change detection map, it is not suited for very crowded scenarios.

A specific subclass of methods that exploit foreground segmentation to count people is composed by algorithms with very strict requirements on position and orientation of the camera. The most common constraint is requiring a camera oriented perpendicularly to the ground [VTH06, CCC06, HC03, BCS07] as it simplifies the problem from the geometrical viewpoint removing most of the perspective distortion and occlusions. The drawbacks are that it limits the counting area to a subimage where people appear parallel to the optical axis and requires an ad-hoc camera.

Methods based on people detection

To avoid any assumption on the structure of the scene (e.g. a planar scenario, static cameras) and not to depend on background-foreground segmentation that may induce instability in the algorithm, a set of methods based on the direct counting of detected pedestrian has been proposed [MAT10, LTR$^+$05, RTK05, SLBS06].

Generally the drawbacks of this class of methods are the computational complexity and the instability with complex images (e.g. occlusions, complex background) that may lead to many false positives and false negatives.

To stabilize the results of the detection algorithm (e.g. the heads obtained with HOG) it is possible to use a tracker to minimize the false positives [PES10]. In this case the results are filtered using tracking and exploiting a prior assumptions on the *uniformity* of the motion of humans. Zeng et al. [ZM10] proposed another method based on the direct count of detected people based on the detection of head-shoulder part of pedestrians using HOG processed with Principal Component Analysis (PCA).

A possible method to decrease the computational cost and limit the number of false positive is to combine the algorithm with the results of the change detection. However the methods that fall in this class share the same requirements of the foreground segmentation method (e.g. static camera) and are sensitive both to errors from detection and segmentation. For example, given the foreground map, it is possible to extract the features to localize the pedestrians that have generated the map [LTR$^+$05]. Similarly Sidla et al. [SLBS06] apply together a Canny edge detector and a background subtractor to look for the shape of the upper part of human body. The results are refined using a tracker to exploit the temporal coherence to stabilize the count of the people that pass through a virtual gate.

Chen et al. [CTTC07] proposed to combine the same scheme of joint background subtraction and detection, using the mean-shift algorithm [CM02] to look for symmetric shapes (the assumption is that a pedestrian is symmetric) in the foreground map.

As with the algorithms based on foreground segmentation [KCCK02, CCC06, HC03, BCS07] the problem can be simplified requiring a camera perpendicular to the ground. In this simplified setting a feature extraction based on foreground edges may be sufficient [YCSX08]

Moving feature points can be used as a feature description to estimate the number of people as, at a fixed scale, they can be assumed to be proportional to the number of people in the scene [ASAM09]. The possible solutions range from linear models with corners [ASAM09] to SVR with in input properties of clusters of SURF (size, density, distance from the camera) [CFPV10].

Even if no geometrical model is adopted, most of the counting algorithms based on detection show a dependence on the angle α and assume implicitly that it is close to 0. This requirement is due to the deformation effect of the perspective that changes the appearance of the pedestrian and modifies the aspect ratio of its bounding box (the latter is assumed constant in most of the detection methods).

Texture-based methods

With very dense crowding levels (Figure 5.1 (a) left) both the detection and the foreground extraction methods fail: the first one because of the strong occlusions in a crowd make too difficult to detect each person, the second one because, once the crowd has filled the foreground map, it is not possible to distinguish different density of crowds. Methods based on texture analysis have the advantage of being theoretically able to work in those difficult contexts, however for this class of algorithms it is more challenging to obtain a good estimate with smaller occupation levels.

The general idea is that groups of different densities of people induce textures of different roughness and hence textures can be exploited to estimate the crowding level of the observed area.

Texture can be analysed with Gray Level Dependency Matrices (GLDM [HSD73]) in conjunction with information as contrast, homogeneity, energy and entropy [SWP09] or with wavelets and GLDM [XLH06]. Generally texture features cannot be easily correlated to the number of people and a machine learning algorithm is needed to obtain the final estimate of the number of observed people.

The same scheme of combining an SVM with texture descriptors can be extended to a spatio-temporal case analysing the texture both temporally and spatially by employing the Sparse Spatio Temporal Local Binary Pattern (SST-LBP [Mäe03]) [YSZ$^+$11].

Further descriptions includes the Minkowski fractal dimension [MVCL97], Translational Invari-

ant Orthonormal Chebyshev Moments (TIOCM) and GLDM. Rahmalan et al. [RNC06] proposed an empirical evaluation of those descriptions together with Self Organizing Maps (SOM) [Koh01] to classify images in five level of crowd density. The results shown suggest that the GLDM and the TIOCM obtain better performances w.r.t. Minkowski fractal dimension. A Further comparison [MVCL98] tested GLDM and spectral descriptions in the Fourier domain that appear to achieve similar accuracy.

As for the size of the foreground pixels, the perspective of the image warp the texture. A possible solution [WLLX06] is to analyse the GLDM starting from a set of cells on the image that are resized in order to compensate for the perspective distortion. Errors and instability due to imprecisions in the grid can be limited looking to local extrema of the Harris-Laplace space of the image.

Texture and foreground segmentation based methods

A straightforward extension of the methods proposed in the previous sections is the combination of multiple features extracted from the change detection map, texture information and other descriptions extracted from the image pixels, to obtain a single and more robust estimation.

Simple properties of the segmented blob as position and vertices can be exploited with a SVM [YZ07]. Similarly Kong et al. [KGT05] proposed a learning method based on the extraction of histograms of the edges orientations and blob sizes. As with the previous algorithm the data are normalized before the estimate to avoid perspective distortion.

Neural networks have been applied to compute the number of people starting from basic features extracted from the foreground mask as blob area, perimeter, perimeter-area ratio, edges length and aspect ratio [YST10]. Each descriptor is normalized by using a homography constructed to map the image into a new one where the size of people is invariant w.r.t. their position. Chan et al. [CLV08, CV12] proposed a similar combination of foreground blob and edge features mixed together with texture information (e.g. GLCM) given in input to a Gaussian Process Regression algorithm.

Since learning algorithms may require a big amount of data to compute a robust solution, Tan et al. [TZW11] proposed a semi-supervised regression method based on Elastic Net [ZH05] to lower the requirements on the size of the training set.

5.3 Proposed Method

Our objective is to study and develop a method able to balance precision, human interaction, and constraints on the type of scenario while maintaining very low computational requirements

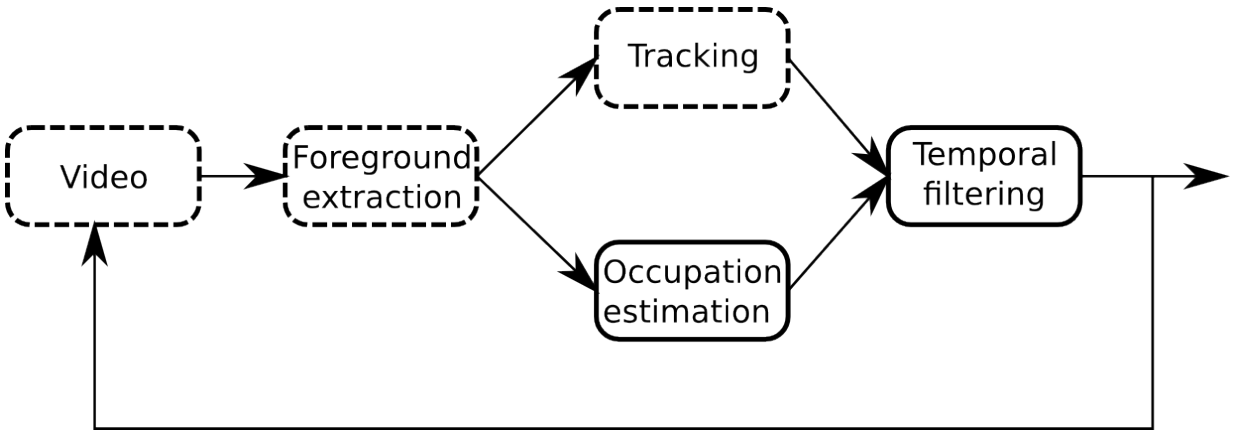


Figure 5.5: Pipeline of the proposed method. The dotted steps are not strictly connected with people counting and hence are not described in this chapter. The tracking is optional and can be discarded to have a lower computational cost.

to make it suited to be used in real-time standalone or on embedded systems or as an input to a higher level algorithm.

To fulfil our objectives we have chosen to design an approach based on foreground extraction. The main motivations are: (i) a foreground-background segmentation is often required by computer vision algorithms and hence, in these cases, it does not add any computational cost to the pipeline (ii) this class of approaches is suited to distinguish between a broad range of different crowding level.

Classical background-foreground segmentation algorithms (e.g. change detection) constraint the system to be used with static cameras and make our system not suited to distinguish between very high levels of crowding. We assume also the ground to be planar to simplify the algorithm and the configuration of the system, however, having in input the 3D model of the ground, the method can be extended to work in a more general context.

As a starting point we have chosen the algorithm proposed in Kilambi et al. [KRJ+08] and tested in Fehr et al. [FSM+09] since it does not assume or require any specific camera configuration rather than the presence of a planar ground, being more general and better suited to manage occlusions than simpler techniques (e.g. [MLHT04]).

Our claim is that it is possible to remove the model-based approach to obtain an algorithm suited to work with general group shapes and with low computational requirements that, at the same time, is able to achieve equal or better accuracy of the algorithms in the state of the art.

The main pipeline of the algorithm is shown in Figure 5.5, a detailed description of all the steps is provided in the following subsections.

5.3.1 Ground occupation analysis

We assume we have in input (or we may estimate): the homography H_0 that maps the image plane to the ground plane and the homography H_T that links the image plane to a virtual plane parallel to the ground at height T.

For each frame of the video we require the set of connected components $C_t = \{c_k^t\}$ extracted from the foreground map $F_t(i, j)$ computed at the instant t.

We define the matrix A of the same size as the frames of the considered video such that each entry is the size of the pixel with the same coordinates once it is projected on the ground.

In the first step we compute the area occupied on the ground of each connected component assuming that each object have a fixed height T in the world (see Figure 5.3). Each pixel $p = (i, j, 1)$ of each connected component c_k^t extracted from the foreground map F_t is projected to the ground and it is retro-projected to the image from the height T

$$p_1 = (H_T^{-1})H_0 p \tag{5.1}$$

If $p_1 \in c_k^t$ the area associated to the connected component is incremented of $A(p)$. To speed-up the algorithm we precompute the mapping of all pixels of the image.

With this procedure we estimate the area occupied on the ground with a procedure equivalent to the heuristic algorithm of [KRJ+08] that proposes it as a standalone method or as an initialization step for the model-based approach.

There are four sources of error that affect this estimate:

- The difference between the fixed T and the real height of the object.
- The variability of intra-person distances.
- Ambiguity due to the loss of information in the perspective projection.
- Errors in the change detection map.

We are not interested in studying errors due to wrong segmentation as we have no reliable model. In the next paragraph we analyse more the remaining elements.

Analysis of the error induced by wrong object height

The approximation of considering a constant height for all the objects causes an error on the estimation of the area occupied that depends on: (i) the distance from the camera (ii) the inverse of the height of the camera (iii) the error between the fixed height and the real height. Consider

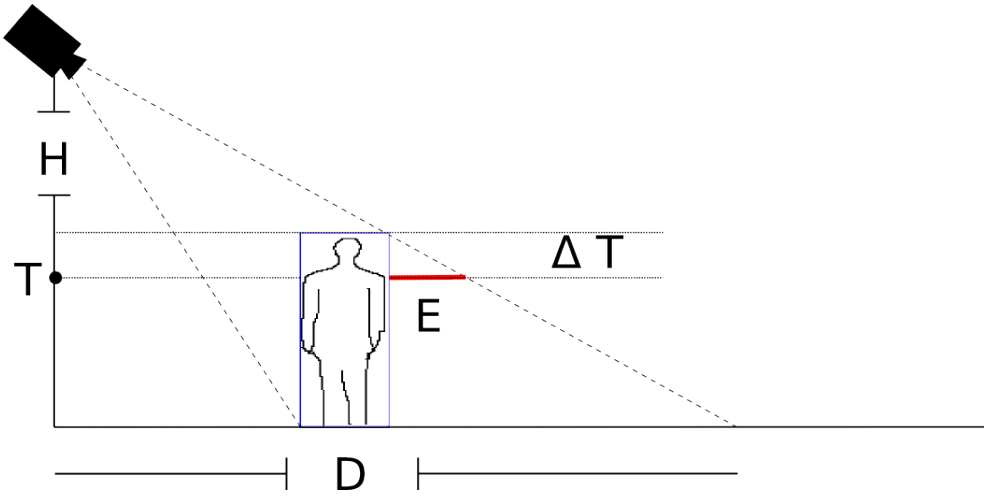


Figure 5.6: The error induced by a wrong estimation of the height of the object in the scene is proportional to the distance from the camera, to the inverse of the height of the camera and to the error ΔT between the real height and the estimated one.

the two dimensional case of Fig 5.6: given the distance D between the projection on the ground of the camera and the projection on the ground of highest point of the connected component, the error ΔT between the height of the object and T and the height of the camera C the error is:

$$E = \frac{D}{H}\Delta T \tag{5.2}$$

if $\Delta T \geq 0$. In cases where T overestimates the height of the object, the intersection between the virtual blobs B_0 and B_T (see Figure 5.3) underestimates the occupied area and it may be null.

Note that this method leads to a correct result only for objects with an upper side flat and parallel to the ground. In case of curved object (as with the head of humans) the area estimated using the real height would be a point and it is meaningless for the estimation.

Moreover, if we consider groups, this error will concern only people whose projection on the ground do not intersect the contributions from other people in the same group. Thus, the influence decreases as the group cardinality grows (see Figure 5.7).

In Figure 5.8 we show an experimental evaluation of the error considering the estimated occupation on the ground of groups of people of different cardinality while they walk away from a camera (PETS 2009 view 1). In Figure 5.8 (a) it is clear the effect for single pedestrians as the occupation increases as they walk away; in (b) we can see that the same effect is not visible for bigger groups of 4 and 6 people, where the instability of the segmentation dominates the error induced by ΔT. Moreover since the variability of small groups is negligible w.r.t. the area occupied by bigger groups (Figure 5.8 (b)) it is reasonable to consider the approximation as noise and not to compensate directly the error.

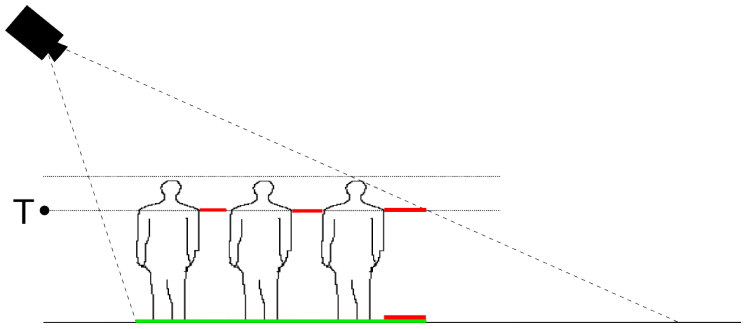
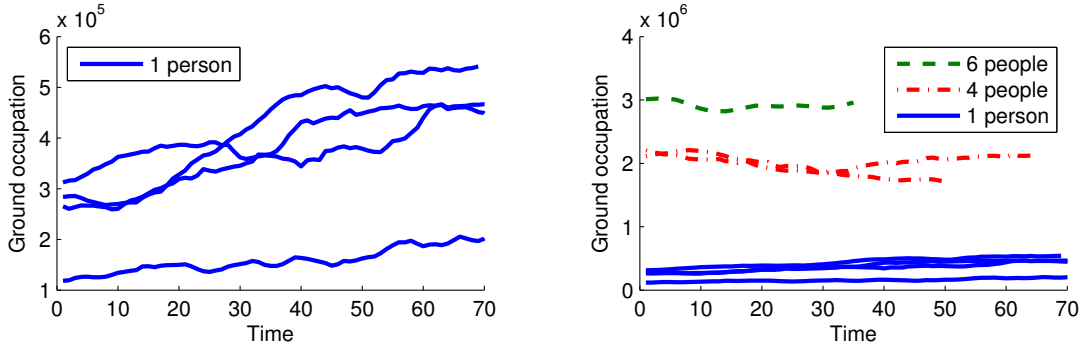


Figure 5.7: Given a group of people, the error on the area occupied on the ground (green) induced by an underestimate of the height of people is only due to the farthest row of people in the group. The errors of all the other people have no effect as they involve areas occupied by other people in the same group.



(a) (b)

Figure 5.8: Occupation on the ground computed on different people manually tracked while they walk away from the camera on the PETS 2009 S2 dataset. (a) as expected the estimated occupation increases as single people move away from the camera due to the error in the estimation of the height; (b) with bigger groups of people the influence of this error is dominated by segmentation errors and the variation of the occupation of a single pedestrian is negligible w.r.t. the occupation of bigger groups.

In cases where it is expected to observe only small groups of people that move along the optical axis, it may still be reasonable to employ a tracking algorithm to solve for the correct space occupied on the ground exploiting observations of the same blob in different positions.

However this is an extreme case as, with scenes having a high depth of field, probably the error would be dominated by numerical instability and occlusions in the farthest part of the image.

In our experiments we fix T to underestimate the expected real height in order to detect the highest number of people possible being robust to partial errors in the foreground extraction.

Lower bound computation

To compute the uncertainty due to occlusions generated by the perspective projection, we compute a lower bound of the number of people that can generate the foreground segmentation map in input. As opposite to confidence intervals, this bound is not a constant shift in the estimate, but depends on the shape of the map and on the geometry of the scene.

The estimation depends on the maximum area that a person can occupy, but also on the ambiguity in the shape of the connected component: in Figure 5.9 we show two similar blobs generated by a different number of people.

This uncertainty is not common to all people configurations: if the blob is disposed along the optical axis of the camera there is no possibility to discriminate between a dense row of people and a lower number of people as long as the head of the person in front touches on the image the feet of the following person; in the case of people disposed along a line perpendicular to the optical axis there is not such a degenerate configuration and the gap between the estimate and the lower bound of the estimate is close to zero.

To compute the lower bound, we project the lower perimeter of each connected component on the ground in the world and then back in the image after having raised it to a fixed height T' (an upper bound on the height of the expected people). With the line obtained we split the connected component in two different parts and the process is repeated on the upper part until the convergence is reached (see Figure 5.10). Note that the homography needed to map points from the plane parallel to the ground at height T' to the image plane can be derived from H_0 and H_t and thus is not required in input. Finally we compute the space occupied on the ground with the same algorithm used to compute the original estimate to this new set of blobs.

Note that the lower bound is non linear w.r.t. the size and position of the blob, as connected components (or parts of them) that are not coherent with the presence of people are not considered.

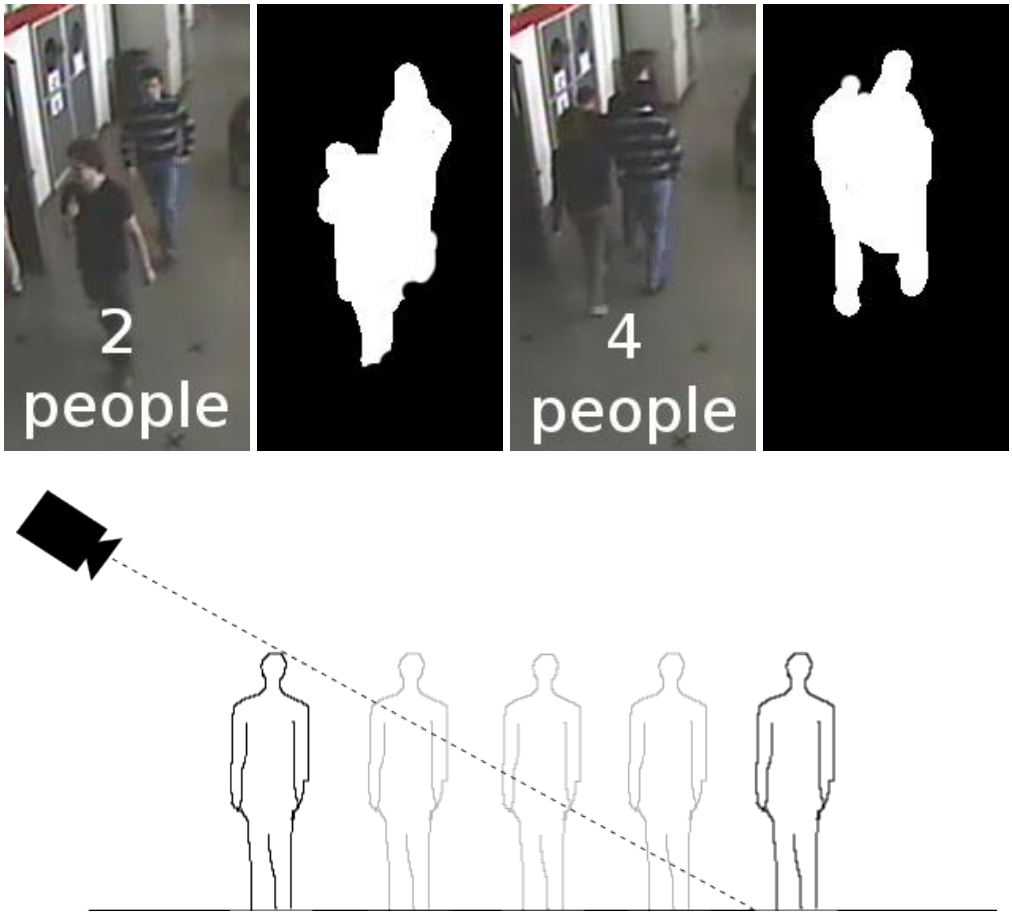


Figure 5.9: Despite their similar appearance two similar segmentation maps can be generated from a different number of people. This problem is more evident with rows of people parallel to the optical axis. As far as the projections on the image of the first and the last person are connected it is not possible to understand if there are people in between them.

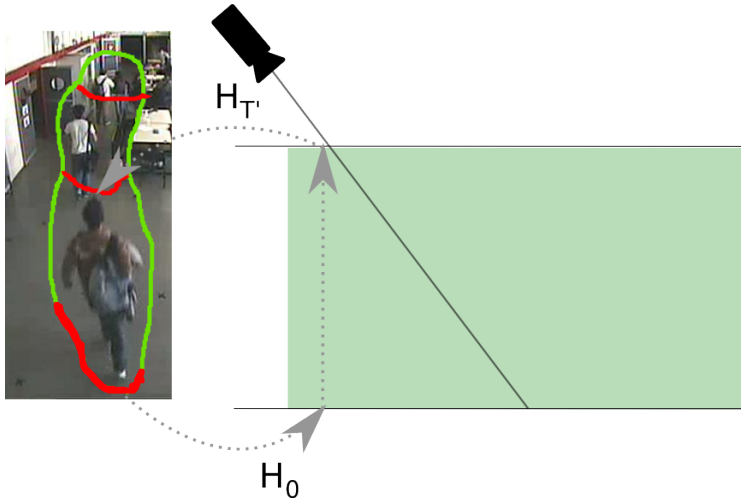


Figure 5.10: To compute the lower bound each connected component (green) is split in parts by projecting its lower perimeter (red) to the ground and then back to image from the height T'. The process is repeated until convergence. Each of the new connected components corresponds to a single row of people positioned perpendicularly to the optical axis. Each row is assumed to be as far as it is possible from the others considering that their projections on the image have to be a single connected component.

Non linear space-people relation

To compute the density of the groups in the image we assume that the space occupied on the ground is proportional to the number of people. However, while a linear model seems reasonable and has been successfully applied in different contexts (see [GBTTO12]), it does not take into account that the intra person space is counted for the estimation and has a different impact depending on the size of groups.

In a group both the space occupied by people and the intra person space is considered as occupied on the ground, hence the relative weight of the space really occupied by people and the space between them is different depending on the number of people being less important in small groups and null with single pedestrians.

To avoid to underestimate single pedestrians and small groups (or to overestimate big groups) maintaining at the same time a simple approach we filter the estimate with a piecewise linear function. With small occupations we amplify the result of a factor β with higher values the output grows as the input (see Figure 5.11):

$$o(x_k) = \begin{cases} \beta x_k, & \text{if } x_k < \alpha \\ \beta\alpha + x_k - \alpha, & \text{if } x_k \geq \alpha \end{cases} \tag{5.3}$$

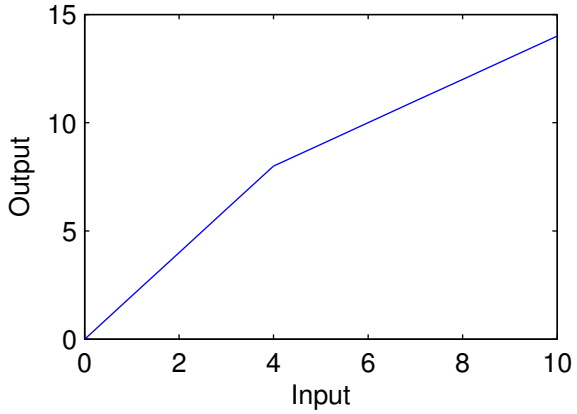


Figure 5.11: Plot of the function (with $\beta = 2$) used to amplify the occupation of small groups of people.

where x_k is the estimate, α and β two parameters of the algorithm.

This amplification aims to compensate for the absence of intra people space in very small groups or single pedestrians: in Figure 5.6 we show an empirical proof that, while the space occupied by 4 and 6 people is almost perfectly proportional, the space occupied by a single pedestrian (if we consider a mean over the time) is underestimated.

5.3.2 Temporal filtering and refinement

In this step we aim to exploit information from different instants to obtain the final estimate filtering out outliers in the estimate given by temporary degenerate configurations.

Since the uncertainty in the information extracted is not constant in time and we have a direct measure of it in the gap between the estimate and the lower bound of the number of people, we filter out peaks associated to high gaps. Given a parameter D_{max}, at each instant we store the last D_{max} estimate/lower bound values and we give in output the values associated to the smallest gap. This easily filters unreliable peaks in the data due to degenerate geometrical configuration. The drawback is that we might introduce a delay proportional to D_{max} in the estimate. The parameter has to be chosen carefully to balance the need of a fast update of the output and the ability of filtering persistent degenerate configurations. In Figure 5.12 we show an example of sequence that we aim to solve without the introduction of false peaks in the estimation. Figure 5.13 shows a result from a real world example, where an error in the estimate is successfully filtered using the proposed algorithm.

Furthermore, in cases where a reliable tracker is available, we may use consecutive temporal information to stabilize the results and to exploit multiple observations in multiple positions to

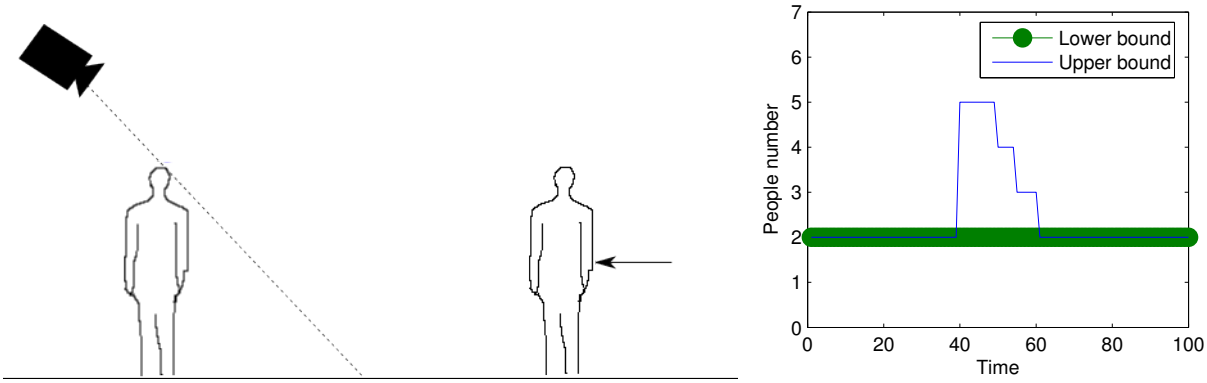
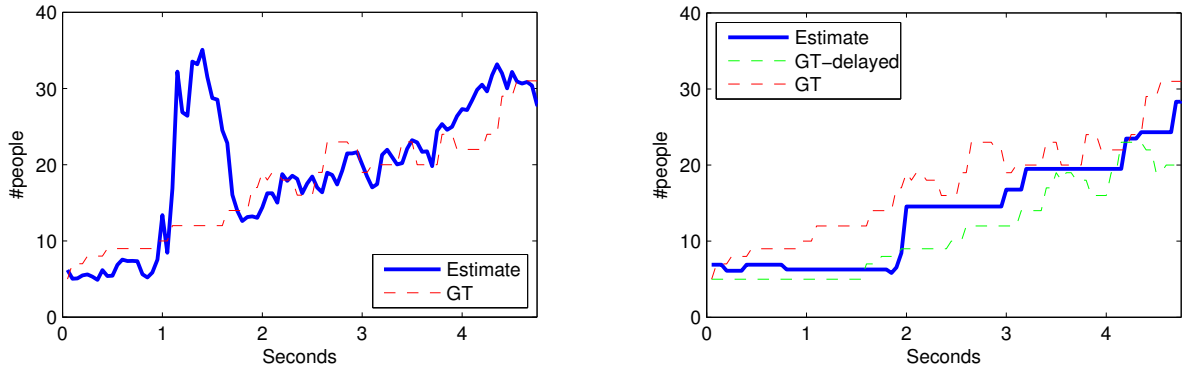


Figure 5.12: Synthetic sequence where the proposed filtering helps to avoid false peaks in the number of people due to temporary critical people configurations. As soon as the head and the feet of the two people merge on the image the estimate increases. Searching in the recent history for a close estimate-lower bound we can easily filter such temporary configurations.



(a) (b)

Figure 5.13: The proposed temporal filtering can remove temporary errors due to ambiguous configurations. (a) the unfiltered output shows a high error due to an ambiguous change detection map configuration; (b) the filtering detect and discard the ambiguous information obtaining a more stable and correct estimate. Due to the delay induced by the filter the estimate has to be considered correct if it is included in the area delimited by the ground truth and its delayed copy (in this specific case the delay is 1.5 seconds).

compensate for the error induced by ΔT. However, in our experiments, we have chosen not to adopt any tracking method as we are interested in maintaining low computational requirements.

In Figure 5.14 we show a scheme of the steps of the people counting algorithm as it has been configured in the experiments.

5.4 Computational complexity

The computational complexity has been one of the key aspect that we have taken into account during the study of the method. Once the foreground map has been obtained, the first step is the extraction of the connected components, that can be done with linear complexity w.r.t. the number of pixels of the image (e.g. [WOS09, CCL04, SHS03]). The following step is the split of the connected components to compute the lower bound as it has been described in Sec. 5.3.1. This involves only manipulations of the perimeter of the connected components and the number of operations is bounded by the number of pixels. More precisely the number of splits is bounded by the minimum number of people positioned in a row parallel to the optical axis that are required to fill the image.

The computation of the estimate and of the lower bound has the same computational cost and is linear w.r.t. the number of pixels in the connected component, since for each pixel are needed at most three access to three different maps: (i) the map to obtain the coordinates result of Eq. 5.1 (ii) the foreground map (iii) an access to the matrix A.

The cost of the filtering step depends on the parameter τ_{max} as it is needed only an update and a search on a circular list of length τ_{max}.

As result the algorithm has a complexity that is linear in the number of pixels of the image and linear w.r.t. the number of connected components.

5.5 Parameter selection

The proposed algorithm has six parameters: $\alpha\ \beta\ \tau_{max}, T, T'$, the area occupied from a pedestrian.

T and T' can be set manually considering the mean human height and the parameters of the amplification can be set manually looking to the data or optimizing the results on a training set. Since they aim to amplify the signal of a blob with no intra person area, it is reasonable to choose a value that corresponds to less than one/two people and to amplify it to reach a value close to the correct one.

In a real application the value of τ_{max} is related to the speed of the output needed, and the only important requirement is that it should not be bigger than the time that a person that we are

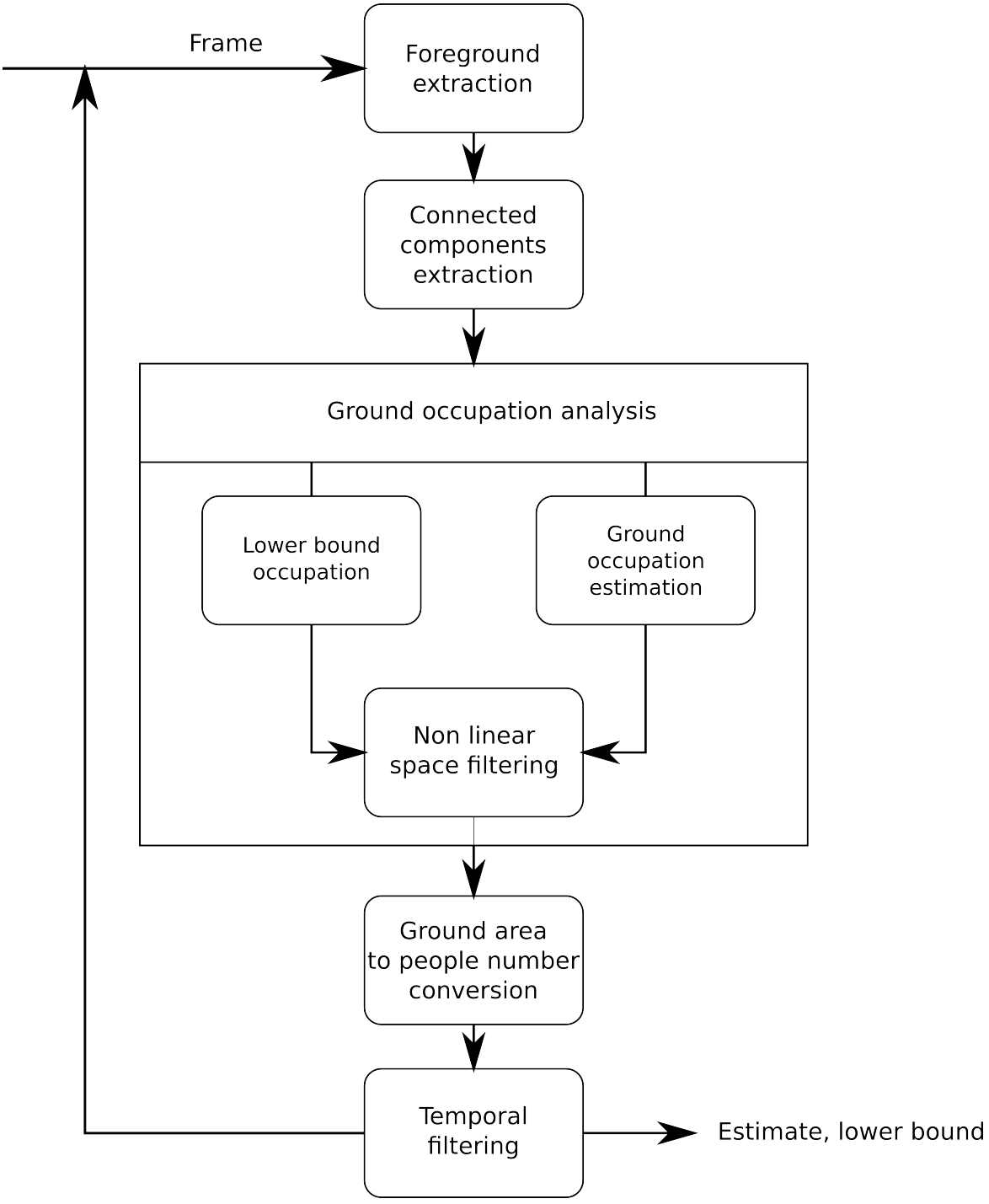


Figure 5.14: Main steps of the people counting algorithm. See text for details.

Error measure	PETS 2009								
	R0					R1			
	13-59	13-57	14-06	14-17	14-31	13-59	13-57	14-06	14-17
mean occluded	0.8	1	0.9	1	0.8	0.5	1.1	1	0.7

Table 5.1: Manual estimate of the mean number of people that are partially visible in the PETS 2009 dataset. This is a index of the uncertainty in the ground truth, as for a human being it may be correct to count them or not.

interested in counting passes into the scene, as bigger value may filter out its contribution.

However, as to compare with the literature we needed to optimize the results w.r.t. an instantaneous measure, we have balanced the effect using a set of training data (details are provided in the next section).

Moreover our filtering step is penalized if compared with a frame by frame measure, to limit this we have employed an averaging between neighbour frames to interpolate our results between two estimates.

5.6 Experimental evaluation

The objective of the experimental evaluation was to assess the performance of the method and its ability to work with different scene geometries.

The videos chosen for the experiments are from the datasets (see Appendix A.2) PETS 2009, PETS 2007 and In-house indoor.

The first dataset has been chosen since the challenge related to it reports the results on its videos of a good number of state of the art algorithms and it is updated to 2010. The dataset of the people counting challenge is the view 1 of PETS 2009 S1. The sequence 14-33 of S1.L1, that has been used by only a few methods has been removed as all the frames show a group of people in the center of the image and hence we have not been able to compute the background. We have added the sequence 14-31 from S1.L2 to compare our result with the one of Bunster et al. [GBTTO12].

In Tab. 5.1 we report the mean error that can be expected considering an algorithm that provides perfect results but discards all the people that are not yet fully included in the image when it is compared with a ground truth that considers all the people that have a part in the image. This gives an idea of the best result that is reasonable to expect and helps to evaluate the difference in performance that can be expected from two methods with a similar accuracy.

While the task of PETS 2007 does not include people counting, we have chosen to test the third view of its subpart S00 to compare our method with the results of [KRJ$^+$08, FSM$^+$09]. The

third camera shows a stair in the upper part that is not included in the region of interest for our algorithm as we consider only planar environments. Feher et al. [FSM+09] report the results both on the third view and on the first view of PETS 2007. However the presence of static people for full length of the video does not allow to build a reliable background model for the view 1 and they report high errors that are not related to the algorithm itself.

The third view shows a totally different setting w.r.t. the first camera of PETS 2009 with a camera that observes from a different perspective a closer scene, hence, together with the latter dataset it is a good benchmark to test the versatility of the algorithm.

We have adopted the in-house indoor dataset to test the algorithm on longer sequences in a real-world uncontrolled video surveillance setting with a rough calibration and different perspective. Moreover we have used it to test successfully the ability of the algorithm to process in real-time the video streams directly from the camera.

5.6.1 Experimental setting

To be able to compare with the broader range of algorithms available in the literature we have adopted two measures of errors, namely the Mean Average Error (MAE):

$$MAE(o^*, o) = \frac{1}{|o^*|} \sum_{i=1}^{|o^*|} |o(i) - o^*(i)| \tag{5.4}$$

and the Root Mean Square Error (RMSE):

$$RMSE(o^*, o) = \sqrt{\frac{1}{|o^*|} \sum_{i=1}^{|o^*|} (o(i) - o^*(i))^2} \tag{5.5}$$

where $o(i)$ is the number of people computed at the instant i and $o^*(i)$ is the ground truth at the same instant.

In our experiments we report the MAE as it gives an intuitive measure of the error. The RMSE is reported where it is needed to compare with results from the literature.

To test the algorithm we have employed a simple background subtraction model based on a background frame computed using a running average that has allowed us to obtain a real-time system with a few parameters to set.

We have used the video 13-59 of PETS 2009 to select the parameters of the people counting algorithm, that is a filtering window of 15 frames (two seconds), $\beta = 3$ and α equivalent to the area occupied by 0.4 people (before the amplification of the signal). The same parameters have been applied for all the videos of all the datasets, the occupation on the ground for each

person has been computed once for all the datasets (the first quarter of the PETS 2007 video and a training video of our indoor dataset) as the unit measures of the calibration were not consistent.

The background model of PETS 2007 has been manually tuned on other videos of the same view adding close and dilate morphological operations to avoid that a person with open legs has an empty intersection between B_0 and B_T.

Method	PETS 2009										PETS 2007
	R0					R1					-
	13-59	13-57	14-06	14-17	14-31	13-59	13-57	14-06	14-17	14-33	S00 (RMSE)
Proposed	1.72	1.80	2.01	2.00	2.47	0.69	1.59	1.73	1.31	-	1.38
	RMSE: 2.49					RMSE: 1.94				-	
Baseline	1.76	6.5	2.41	4.89	3.64	1.14	3.23	3.08	1.19	-	2.91
[FSM+09, KRJ+08]	-	-	-	-	-	-	-	-	-	-	1.67
[GBTTO12] LRM	RMSE: 5.35					-	-	-	-	-	-
[DPG+10],[ASAM09]	3.86, 1.84	2.8, 1.45	5.14, -	2.64, -	-, -	-, -	-, -	-, 1.94	-, 1.4	-, -	-, -
[CFPV10], [DPG+10]	1.59, 2.24	1.14, 1.92	5.12, 4.66	2.2, 1.75	-, -	0.82, -	1.46, -	1.99, -	1.62, -	-, -	-, -
[CMV09]	1.4	2.5	-	-	-	0.77	2.23	5.87	0.97	-	-
[PKS09]	1.3	7.18	-	-	-	0.77	4.85	-	-	-	-
[CFB09]	3.3	1.35	-	-	-	3.68	2.23	3.68	0.67	16.8	-
[WJY+11]	0.96	1.03	1.74	0.99	-	-	-	-	-	-	-
[AJBV09]	4.1	-	-	-	-	1.84	-	6.69	1.01	9.6	-
[PES10]	2.3	2.7	-	-	-	1.55	2.62	6.40	4.66	24.4	-

Table 5.2: Results of our algorithm compared with the state of the art in terms of MAE (or RMSE where it is indicated). All the values have been extracted from the cited papers, or from [EF10].

5.6.2 Comparative analysis

In this section we report the results on the test datasets and we compare them with the state of the art algorithm.

In Tab. 5.2 we show the results in terms of MAE w.r.t. the most promising algorithms from the literature (see [EF10] and references therein). Each row of the table refers to a method. Rows with multiple references include results for a given method presented in different papers, and referring to a different subset of videos. Some of the algorithms (e.g. [DPG$^+$10, CMV09, WJY$^+$11]) rely on machine learning techniques therefore they require a training stage that is on one hand time consuming, but on the other hand provides robust solutions tailored on a specific setting. In any case the comparison is done regardless this important difference between the various methods, similarly to what presented in the official PETS 2009 challenge [EF10]. Also, we compare with the baseline method, whose results are obtained by simply applying the homographies (i.e. the heuristic method of [KRJ$^+$08]) using the same background adopted in our approach.

The only algorithm that obtains consistently better results than ours is the one proposed in Wen et al. [WJY$^+$11]. However the method does not run in real-time and, more importantly, it performs a training procedure on a training set composed by frames extracted from the test videos selected to represent the available configuration in the test set. Such procedure is prone to overfitting thus it is not clear what are the real generalization capabilities of the approach. A similar observation can be done for Conte et al. [CFPV10].

Besides that, our results are consistently comparable or better than the available algorithms. The advantage of our method is a higher stability on sequences of different complexity, where other algorithms show performance falls (see, for instance the crowded sequence 14-06 both on regions R0 and R1). Notice how errors differing for a few decimal point can be considered as comparable: for example in PETS 2009 the average number of people that are in the regions borders and only partially in the scene (and hence may be arbitrary counted or not also by a human) is close to 1 (see Tab. 5.1).

On PETS 2007 we have processed the same video of Fehr et al. [FSM$^+$09] to assess the improvements over the algorithm that we have chosen as starting point for our study. The RMSE of our algorithm is 1.38, while the best result reported in Fehr et al. [FSM$^+$09] on the same sequence and with a more complex background model is 1.67. Similarly to the tests presented in Fehr et al. [FSM$^+$09], our background model suffers the presence of a group of people standing in the same position for a long time.

The proposed method is running on a video surveillance prototype system installed in our department and is used to process information in real-time. Figure 5.15 reports a comparison between the estimated number of people and the manually labelled ground truth, on a subset of our in-house indoor dataset. The overall performances of the method on our indoor dataset can be summarized with a $MAE = 0.75$ that speaks in favour of the versatility of the algorithm.

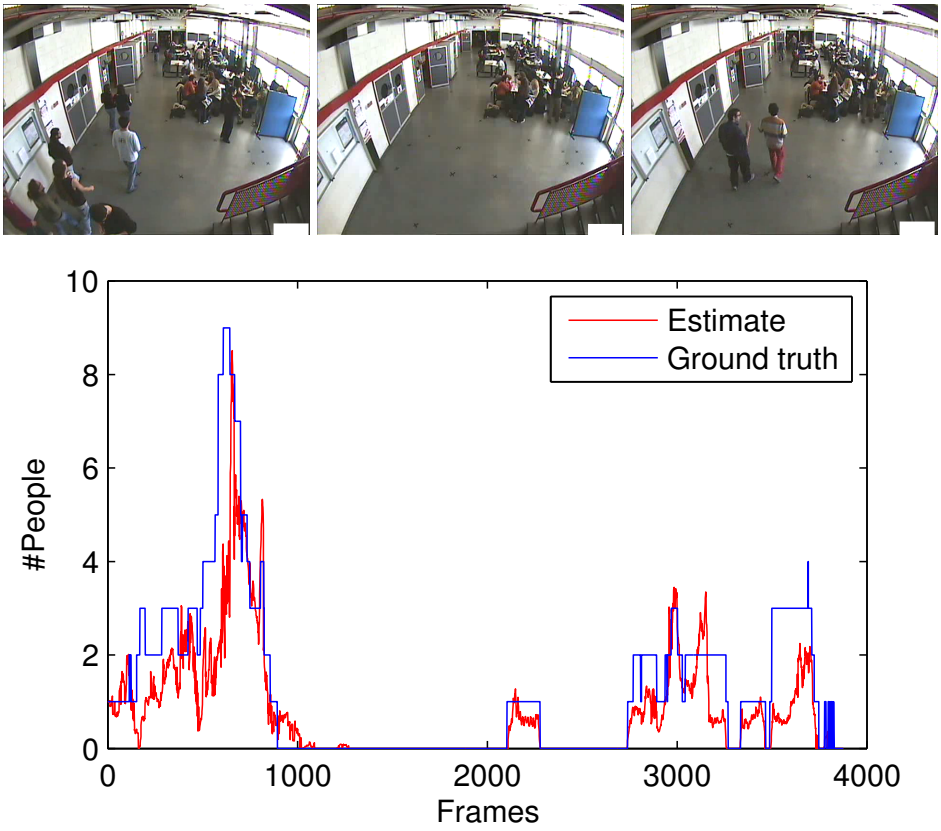


Figure 5.15: People counting results on an example sequence of our in house dataset. It is visible the delay introduced by the temporal filtering.

The computational requirements have proved to be fairly low also from an empirical standpoint: a moderately optimized single thread implementation (including video decoding, and foreground segmentation) runs in real-time on a Intel i5 laptop processor requiring around the 40-60% (a precise quantification is not feasible due to the frequency scaling ability of the processor) of the resources both on PETS and on our indoor dataset.

5.7 Discussion

In this chapter we have presented a method for real-time people counting able to achieve results comparable or better to the state of the art using very limited resources.

Starting from the change detection map and the geometry of the scene we compute the space occupied on the ground and we process it to take into account the intra-person distance and the possible ambiguity in the geometry. With simpler computation and a model free approach we

can achieve better results than a model-based algorithm [KRJ+08, FSM+09] that shares the same initialization step.

The increased accuracy of our method w.r.t. the heuristic method and the model-based approach of [KRJ+08] tested in [FSM+09] is due to the following improvements: (i) the application of a general model-free approach based on the heuristic method of [KRJ+08], that allows us to represent groups of different shapes (ii) the explicit computation of the ambiguity due to the perspective (iii) the use of a piecewise linear function to relate the space occupied on the ground and the number of people (iv) an algorithm to detect and filter out outliers in the estimate due to temporary degenerate geometrical configurations.

The main requirements are the presence of a static camera, a planar scenario (the generalization to a non-planar world is possible, but requires a complex configuration) and a calibrated environment. Those constraints have been chosen as a tradeoff with the ability of working in real-time, from different viewpoints and obtaining a good precision with low and medium crowded scenarios. Moreover the few parameters of the algorithm have proved to be robust to different settings and the absence of a training step reduces the human interaction needed to configure the system.

From the user viewpoint the most time consuming step is the calibration, however the system has proved to work well both with the precise calibration of the PETS datasets and the rough calibration on our indoor dataset that has been obtained using as a pattern a coat rack.

From the precision viewpoint the critical issues are errors in the background subtraction and the handling of very crowded scenes. The first issue can be partially addressed using more complex background subtraction algorithms than the one chosen for our experiments. The second one is common to all the people counting algorithm that are based on foreground segmentation, as, when the foreground map fills the image, the information are not sufficient to distinguish between a dense or sparse crowd. In this case our algorithm computes all the information that a method with the same input can provide, that is an estimate of the people number if packed in a group with a "normal" intra-person space and a lower bound of the minimum possible number of people. Finally the low computational requirements make the algorithm suited to be a module together with other features to extend the range of feasible scenarios.

A preliminary version of the proposed algorithm has been published in [ZNO10], a complete description of the method is under review [ZON13].

Chapter 6

Video synchronization

In this chapter we consider the problem of synchronizing an unconstrained multi-cameras system using video streams. The chapter is organized as follows. Sec. 6.1 introduces the objectives of our study, Sec. 6.2 overviews the state of the art on video alignment and analyses the strength and weakness of existing algorithms. In Sec. 6.3 we describe the proposed method. Finally Sec. 6.4 discusses the experimental results and comparisons.

6.1 Introduction

The objective of synchronization (alignment) is to associate together the frames from two videos to minimize the time differences between their acquisition time.

The synchronization of a multi-camera system is often a crucial requirement in video surveillance and, together with the geometrical calibration, allows us to coordinate multiple cameras to extract and process information from multiple viewpoint or to present coherently to human operator multiple streams.

The synchronization can be achieved with hardware solutions, however it may not be possible in the case of remotely connected cameras systems that suffer from delays, interruptions and frame drops or of the analysis of off-line videos acquired by independent cameras.

The general objective of the proposed approach is to reduce at minimum the requirements on scene and cameras system, the computational cost and human interaction. To this aim we present a synchronization method based on the analysis of the behaviour of people in the scene that, thanks to the high level information extracted, is able to reduce the requirements on scenario and camera system.

6.2 State of the art

The methods available in the literature can be classified depending both on their approach and on the requirements/a priori assumptions on the cameras.

From the point of view of the approach we have two main classes of algorithms:

- Based on a **sparse feature matching** where the algorithm looks for a set of stable points in the space-time volume that can be easily re-identified and matched.
- Based on a **frame by frame analysis** where it is extracted a description for each frame and it is computed the association that minimizes the distance between the sets.

Examples of the first class of algorithms are [WHK07, SBWS10]: the first one extracts corners in the space-time volume and describe and match them with the local jets [LL06], the second one detects flashes in the videos and exploits the associated instants to align the streams.

In the second class of algorithms we have methods that check the coherence of the geometry between the position of dynamic objects (e.g. [PCSK10, RGSSM03]) or a representation of the dynamic of the scene (e.g. [DPL09]) on a frame by frame basis.

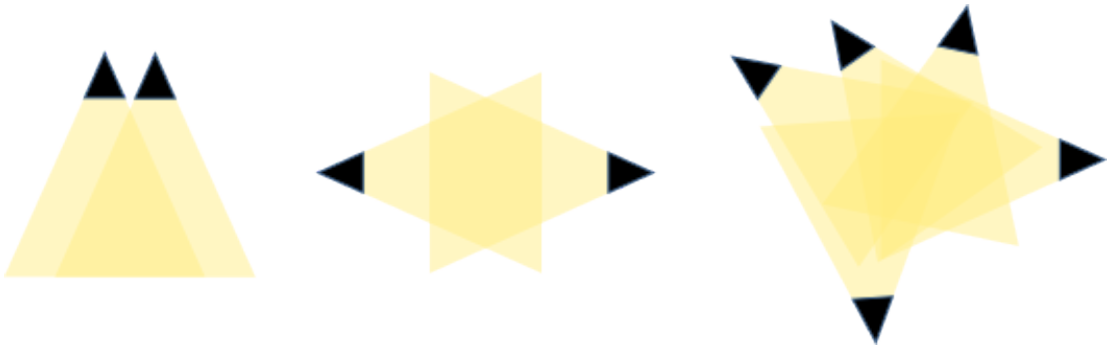
If we base the classification on the implicit and explicit assumptions on the cameras system, the methods can be grouped depending on their requirements on:

- Relative pose of the cameras.
- The knowledge of the geometry of the system.
- The structure of the time misalignment.

The characteristic of the approach and its assumptions are in general only weakly linked, however, as it will be explained later, the match of sparse space-time features limits the possibility to cope with general time misalignments.

Even if it is not directly related to our objective, there are alignment algorithms whose aim is to align sequences acquired in different instants (e.g. [CWX+10, EB11, EDSL11]): in this case the assumption is that the two cameras are following approximatively the same path and the objective is to obtain two sequences where, in corresponding frames, the cameras observe the same area of the world.

Table 6.1 summarizes the main characteristics and the assumptions of state-of-the-art alignment algorithms, in the following part of this section we present an overview of the approaches depending on their assumptions on scene and camera system.



(a) (b) (c)

Figure 6.1: Relative camera poses: (a) paired; (b) frontal; (c) generic. Implicit or explicit assumptions in state-of-the-art algorithms limit the methods to a subset of configurations only. The proposed approach works instead with generic camera configurations.

Relative camera pose

The relative pose type among cameras can be classified into one of three groups (see Figure 6.1): paired, frontal or generic. Cameras are *paired* when their relative rotation is negligible [DPL09, UI06, SDLÁ07] and they differ for translation or focal length [CSI06, WHK07]. Cameras are *frontal* when they are framing each other and their relative rotation is approximatively 180 degrees [CSI06, DPL09]. Cameras have a *generic* relative pose when they have non-negligible relative rotations and translations [PCSK10, Ste99].

This type of requirements can be enforced in the geometrical model used to solve the alignment (e.g. [SDLÁ07]), exploited using heuristics that are more suited for a subset of cameras configurations (e.g. [CSI06]) or requiring a strong overlap of the Fields Of View (FOVs) (e.g. [DPL09]) that cannot be guaranteed in case of generic poses.

Examples of algorithms that may constraint the relative pose between cameras are: Caspi et al. [CSI06] that matched moving image features using descriptions of their motion that are more suited for frontal or paired cameras; Serrat et al. [SDLÁ07] that maximized a pixel based representation of the appearance assuming a negligible translation between the views; Dexter et al. [DPL09] that extracted Self Similarity Matrices from the positions of tracked corners in the image assuming that the FOVs have a high overlap (so that most of the corners can be observed in both the views).

Geometry requirements

A characteristic related to the pose of the cameras is the prior on the geometry of the camera system. In this case we distinguish the algorithms that include or not an explicit representation

Method	Moving cameras support	Geometry independence	Partially overlapped FOV	Temporal warping
[CSI06]	no	F	yes	affine
[DPL09]	yes	yes	no	monotonic
[PCSK10]	no	F	yes	affine
[SBWS10]	yes	yes	-	monotonic
[RGSSM03]	no	F	-	monotonic
[TR09]	no	F	-	affine
[WHK07]	-	-	yes	affine
[Ste99]	no	H	-	constant
[UOD06]	yes	-	-	constant
[LM10]	no	H	-	monotonic
[TVG04]	yes	-	-	constant
[DZL06]	no	H	-	constant
[CI02]	no	yes	yes	affine
[WZ06]	no	yes	-	constant
[TVG00]	-	yes	-	constant
[SDLÁ07]	no	no	no	monotonic
[YLY12]	yes	yes	yes	affine
[LM11]	no	H	yes	monotonic
[LKZI12]	no	yes	yes	constant
[ESK+12]	no	F	yes	affine
[LC10]	no	H	yes	monotonic
[UI06]	no	H	-	affine
proposed	yes	yes	yes	monotonic

Table 6.1: Comparison of characteristics and assumptions of alignment algorithms. When an algorithm has no explicit limitations or proof for a feature, the symbol ”-“ is used. (Key. H: homography, F: fundamental matrix; FOV: field of view).

of the geometry of the camera system. The difference between the knowledge of the geometry and the assumption on the relative poses of cameras is that in the latter case the constraint limits the possible geometrical configurations, but it does not enforce the computation of the geometry and for example the method may work with moving cameras as far as the motion does not change their relative pose.

If the geometry is required, it implicitly encodes other variables: for example the geometry can be encoded with a homography between the views or with their fundamental matrix, and in the first case the method can be applied only to planar scenes. Moreover if the motion is not modelled,

the algorithm is limited to static cameras.

The geometry can be explicitly required as input of the algorithm [PCSK10, RGSSM03], or it can be assumed that the geometrical relations can be retrieved precisely by feature matching [CSI06, WHK07]. In this case the pose is an important variable both in the estimation of the geometry and in the alignment, since restrictions on the configurations permit to exploit specific algorithms and heuristics (e.g. [CSI06]).

Examples of algorithms that require an *explicit geometry* estimate as input to compute the alignment are [PCSK10, RGSSM03]. Starting from a tracked object and an estimate of the fundamental matrix, they look for time alignments that result in geometrically consistent tracks. Epipolar geometry can be exploited directly looking for intersections of tracks and epipolar lines [PCSK10]; whereas the trajectory can be warped to make it a valid set of points for the computation of the fundamental matrix [RGSSM03].

Finally, methods exist that compute the alignment *without any geometric information* by matching positions and trajectories with robust statistics and then estimating the alignment [LKZI12] or both geometry and alignment [Ste99, CSI06]. The computation of the solution requires a filtered set of matches, which can be obtained either with assumptions on the geometry (e.g. planar [Ste99]), or by using heuristics (e.g. [CSI06]), which restrict the type of camera configurations whose videos can be aligned.

Time misalignments

As for the time misalignment, algorithms might assume a *constant shift* [Ste99, UOD06, TVG04, LKZI12, ESK+12], an *affine warping* [CSI06, PCSK10, CI02, YLY12], or just a *monotonic* relationship among recordings [DPL09, RGSSM03, SDLÁ07, LC10]. Methods making the weak monotonic assumption employ Dynamic Time Warping (DTW) to obtain the alignment [DPL09, RGSSM03, LM10, SDLÁ07]. DTW [SC78] is a similarity measure between sequences that assigns *each* frame of a video to at least one frame of the other video, thus assuming a continuous transformation of the timeline. This is in general a very strong assumption as it assumes a complete association that is not available in case of stream interruptions or frame drops.

The structure of the time misalignment that an algorithm can manage is limited a priori in the case of methods based on the extraction of sparse spatio-temporal features. In those cases usually the time misalignment can only be constant or affine, as it is not possible to detect frame drops, interruptions or variations in the frame rates between two key-points.

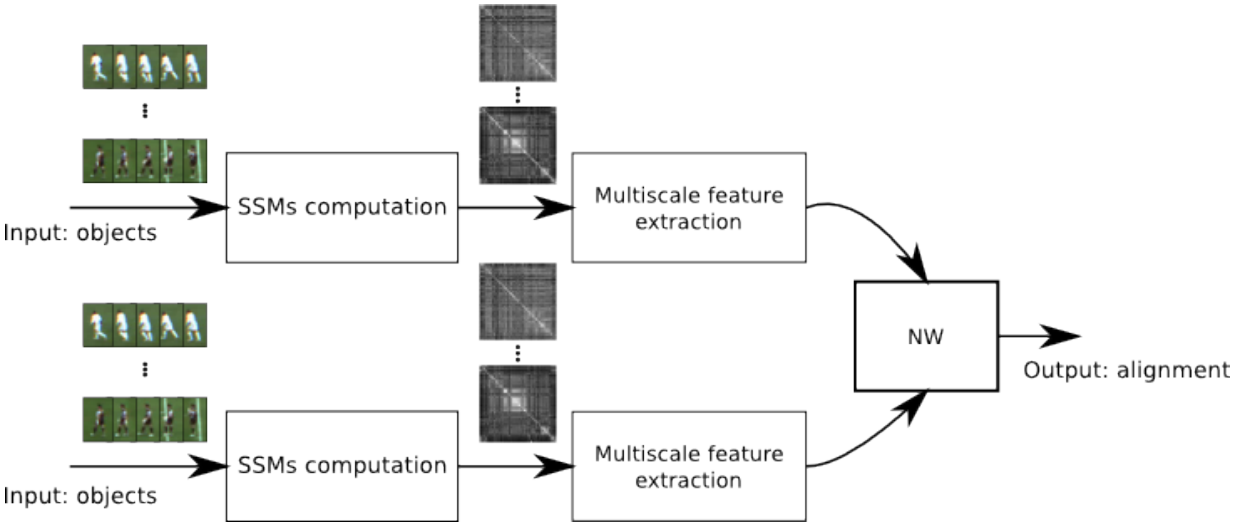


Figure 6.2: Main pipeline of the proposed alignment algorithm.

6.3 Proposed method

State-of-the-art methods lack the capability of aligning a set of videos without implicit and explicit requirements on the geometry and on the time misalignment. Our objective is therefore to extend video analysis to data acquired by different devices in arbitrary poses (generic configuration) and with non-linear and non-smooth time-warping functions.

We present a method to compute a frame-level video alignment with the assumption that cameras view articulated objects and that objects association information is given as input. From the observation of objects actions, we compute a robust estimate of the alignment by exploiting both normal or periodic actions, such as walking or running, and isolated or anomalous events, such as a jump or a fall. The key insight is that, even if the same action appears differently from different viewpoints (as an example, see Figure 6.3), repetitions of the same pattern are approximatively view invariant [JDLP11].

Unlike existing works, we employ an alignment algorithm that models synchronization as a frame association problem, instead of a continuous time warping. The use of a view-invariant description of objects actions allows us to align videos *independently of restrictions on the geometry of the observed scene* using a multiscale representation of the actions over time to compare each instant being invariant w.r.t. time misalignment.

The starting point of our study is the algorithm proposed in Dexter et al. [DPL09] and Junejo et al. [JDLP11], however we have extended the approach to consider a broader range of scenarios (Dexter et al. [DPL09] require strongly overlapped FOV, while Junejo et al. [JDLP11] consider only single objects and is more focused on action recognition).

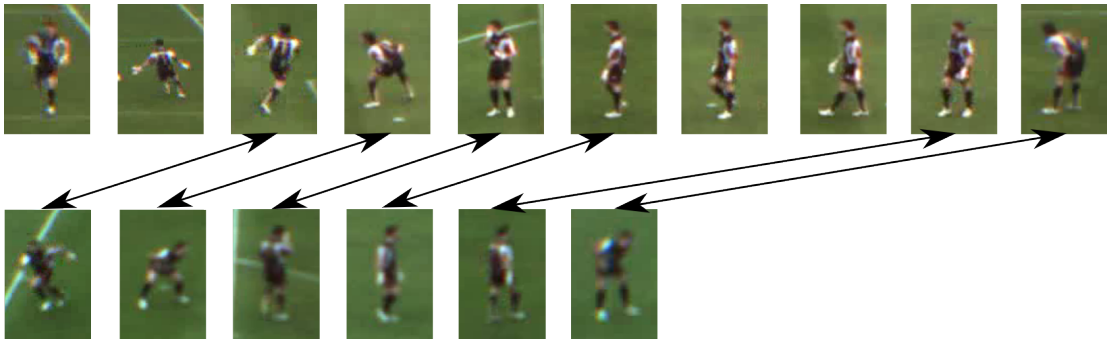


Figure 6.3: An action captured by two frontal cameras with different frame rates. Even if the setting is simple the appearance of each frame differs significantly.

The proposed algorithm is a two-step approach to video synchronization. In the first step we extract from each camera independently a description of the actions of moving objects. In the second step we fuse and compare the data from all the cameras to produce the video alignment. These two steps are detailed below.

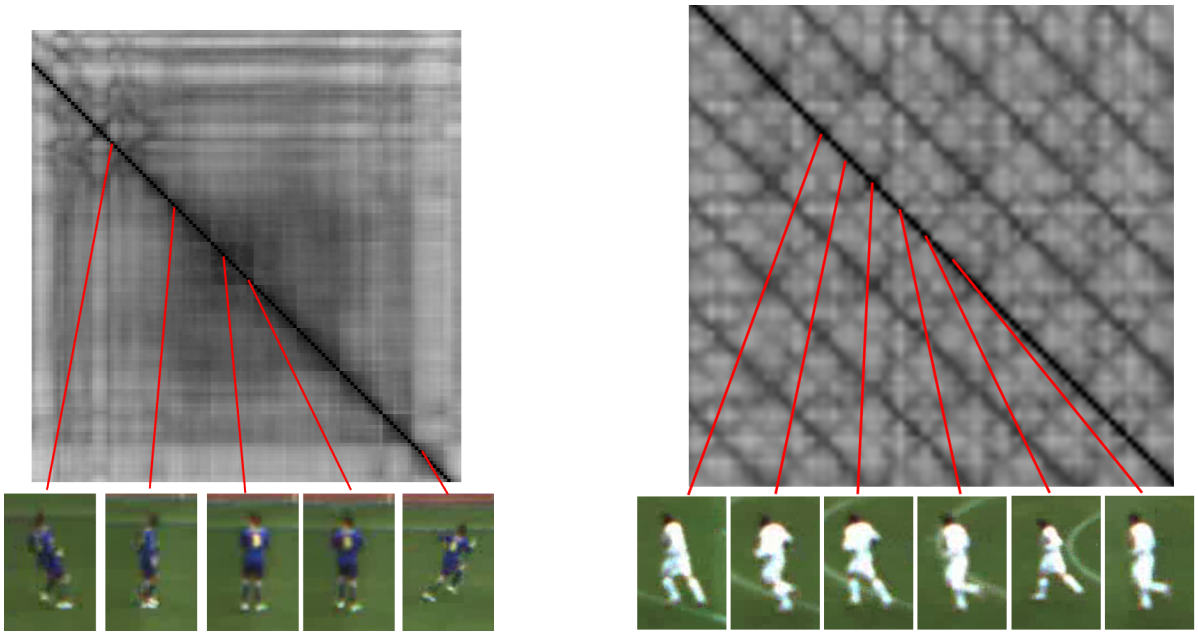
6.3.1 Action description

Let $V_1 = \{f_1^i \ : \ i = 1, 2, ..., N_1\}$ and $V_2 = \{f_2^j \ : \ j = 1, 2, ..., N_2\}$ be two views of the same scene, where f_1^i and f_2^j are their frames whose total number is N_1 and N_2. Let V_i^k be a sequence of observations of object k in V_i and let $|V_i^k|$ be the duration, in frames, of V_i^k. Given object association information, we first extract a description of the action of the objects detected and tracked in each camera and we encode their appearance variations. To this end, we describe the appearance of objects within each bounding box as a sequence of Histograms of Oriented Gradients (see Sec. 3.3.4) in each view. Then we compute a $|V_i^k| \times |V_i^k|$ Self-Similarity Matrix (see Sec. 3.4) , S_i^k:

$$S_i^k(y, x) = 1 - ||\phi(V_i^k, x) - \phi(V_i^k, y)||_2, \tag{6.1}$$

where $\phi(V_i^k, j)$ is the HOG description of the area containing object k in frame j of V_i and x and y are frame indexes.

In Figure 6.4 we show two typical structures induced on the SSM by common actions of articulated objects. A standing person generating small movements and a walking person producing very different structures that we will aim to match between the two views.



(a) (b)

Figure 6.4: SSMs computed on two football players during a match. (a) a player that stands mostly in the same position generates an irregular blob; (b) a player that runs generates a grid, as the same apparent configuration is repeated regularly.

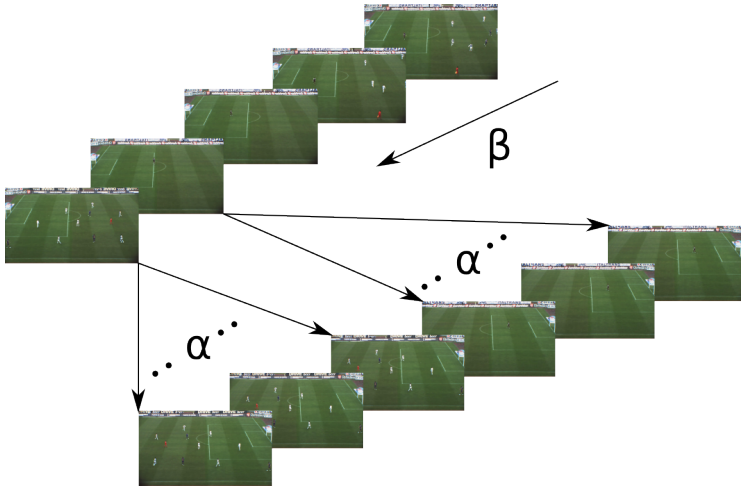


Figure 6.5: Two videos warped with an affine transformation. The first frame is delayed by β frames and their frame rates differ of a factor α.

6.3.2 Multi-scale temporal description and matching

Our objective is to define the set $A_{1,2} = \{(f_1^m, f_2^n) \ : \ m \leq N_1 \ , \ n \leq N_2 \ \}$ of frame pairs that were acquired at the same time instant. Let $T_i(f_i^j) = t_i^j$ be a function that computes the timestamp for each frame of V_i. When the frame rate is not fixed (due for example to bandwidth limitations or frame dropping), we can only assume that $T_i(\cdot)$ and its inverse are monotonic. When the frame rate can be modelled as constant, this leads to an affine relation between the frame indexes of the two videos: $T_1(i) = T_2(\alpha i + \beta)$, where α models the frame rate difference and β is the offset between the first frame acquired by each camera (i.e. the offset between V_1 and V_2). Finally, when the frame rate is known or assumed to be constant and identical in both videos, we can relate the two functions $T_i(\cdot)$ with a constant shift only (Figure 6.5).

We convert the description of the action of each object in a structure that describes how it changes over time. The key idea we pursue is to create a multiscale description of $S_i^k(.)$ for each video V_i and each object k with two aims: (i) to obtain a more representative description; and (ii) to be able to match precisely structures with different scales on the SSM (i.e. different video frame rates).

To this end, we extract a description from each point of the matrix diagonal, aiming to capture the structure of the SSM (that is the structure of the repetition) in a time interval centred on the considered frame. We assume that, locally, the time misalignment can be modelled with a linear function, whose effect is a resizing of the SSM (Figure 6.6).

We use a Polar HOG (PHOG) structure that has been shown effective [DPL09, JDLP11] with a modified grid (see Figure 6.7) to work better near the borders of the SSMs and to avoid smaller cells, whose histograms may be less stable with a few samples (i.e. small radii of the description).

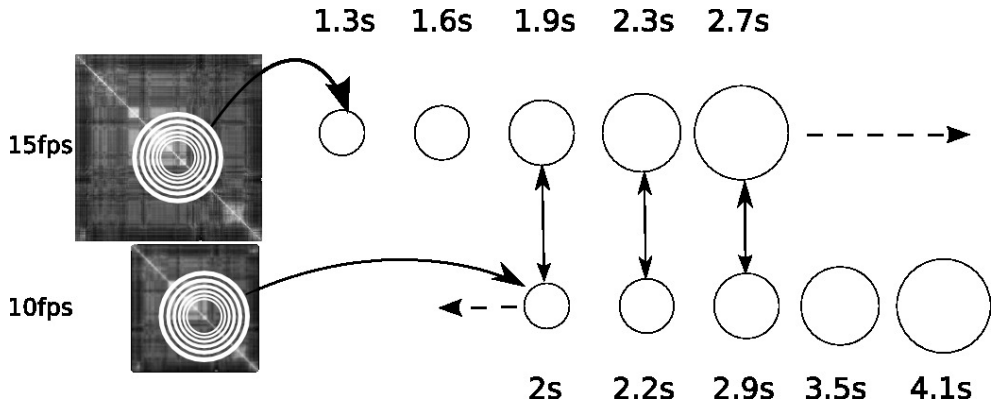


Figure 6.6: A comparison of two SSMs with different frame rates. The distance between their multiscale description is computed by shifting one description until a minimum distance is reached. The correct scale for the alignment is shown with black arrows and is the one that gives the best match between the temporal scale (in seconds) of the representation. The overall distance will be the mean of the distances of the descriptors connected by the lines.

However, instead of estimating the radius of the PHOG from the maximum of the Laplacian [Lin93] as proposed in Dexter et al. [DPL09], we compute a multiscale description that embeds information of different time extents (radii) as follows: for each frame f_i^j for each object k we extract a multiscale PHOG description $P_i^k(j)$ from $S_i^k(.)$ centred in (j, j)

$$P_i^k(j) = \{p(S_i^k, j, r_l) \ \ \forall r_l \in R\}, \tag{6.2}$$

where $p(S_i^k, j, r_k)$ is the PHOG description with radius r_l centred on pixel (j, j) and R is the set of radii parametrized by the minimum radius r_{min}, the maximum radius r_{max} and a constant b. The radius of the level l of the multiscale representation has a radius $r_l = r_{min} b^l$.

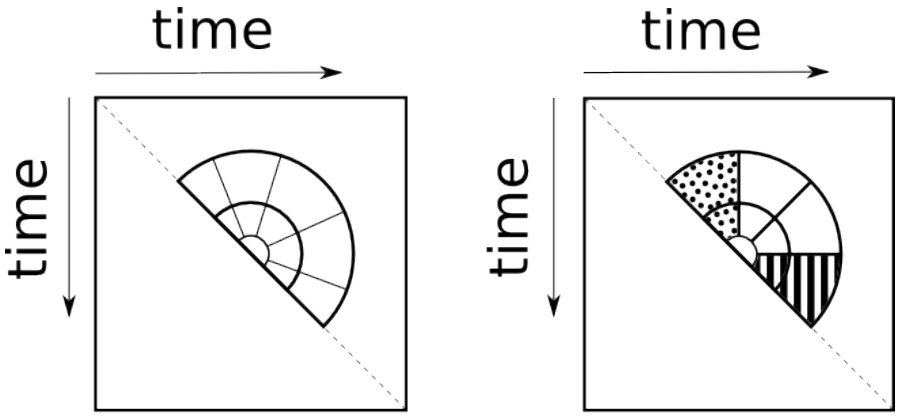
We now want to compare two descriptors $P_1^k(m)$ and $P_2^k(n)$ using an invariant measure. The two descriptors contain a subset of levels that are in common and are shifted in the representation (see Figure 6.8).

We compute the distance $D(\cdot)$ as the distance between the optimal alignment of the two descriptions:

$$D(P_1^k(m), P_2^k(n)) = \min_s \frac{1}{L_s} \sum_{l=0}^{|R|} ||P_1^k(m)_{l+s} - P_2^k(n)_l||, \tag{6.3}$$

where the difference is zero if $l + s \leq 0$ or if $l + s > |P_i^k(\cdot)|$ and L_s is the number of the level in common shifting the levels of one structure of s. To compute the alignment, we extract the list of distances $D(P_1^k(m), P_2^k(n))$ from all possible pairs (m, n) for each object k in the scene. The result is used to derive the sequence of the desired paired frames

$$A_{1,2} = \{(f_1^m, f_2^n) \ : \ m \leq N_1 \, , \ n \leq N_2 \, \} \tag{6.4}$$



(a) (b)

Figure 6.7: Comparison between structures used to compute the PHOG description. (a) PHOG structure from [DPL09]; (b) proposed structure. The stripped area contains all the possible comparisons between the considered instant (i.e. the center of the support of the description) and the past, the dotted one compares it with the future. In the corners of the SSM at least one of them can be computed and it contains all the available information. Moreover all the cells of (b) have the same size.

The alignment algorithm should be able to manage explicitly situations where a description has not a correspondence in the other video (due to different fields of view, occlusions, or time misalignment). DTW may not be robust in this case, as it searches for *at least one* correspondence for each frame. To overcome this problem, we propose to use the Needleman-Wunsch (NW) string alignment algorithm [NW70] that can be optimized with dynamic programming. The notion of a gap in a string can be transferred to a frame that has no correspondence in the other video. To compute the alignment, we first create the $N_1 \times N_2$ matrix C_{NW}:

$$\begin{aligned} &C_{NW}(m,n) = \max((1 - d(m,n)) + \\ &C_{NW}(m-1,n-1), C_{NW}(m-1,n) + G, \\ &C_{NW}(m,n-1) + G), \end{aligned} \tag{6.5}$$

where $d(\cdot)$ is the distance function between frame m of V_1 and frame n of V_2, and G is a constant defined in the range $[0,1]$ that controls the similarity of a frame without association. The alignment is found by looking at all the pairs of coordinates that give the maximum-score path starting from the lower-right corner of matrix C_{NW} and going toward the upper-left corner (see Figure 6.9). The difference between NW and DTW is in that the latter looks for the minimum-score path on the matrix:

$$\begin{aligned} &C_{DTW}(m,n) = d(m,n) + \\ &\min(C_{DTW}(m-1,n-1), \\ &C_{DTW}(m-1,n), C_{DTW}(m,n-1)), \end{aligned} \tag{6.6}$$

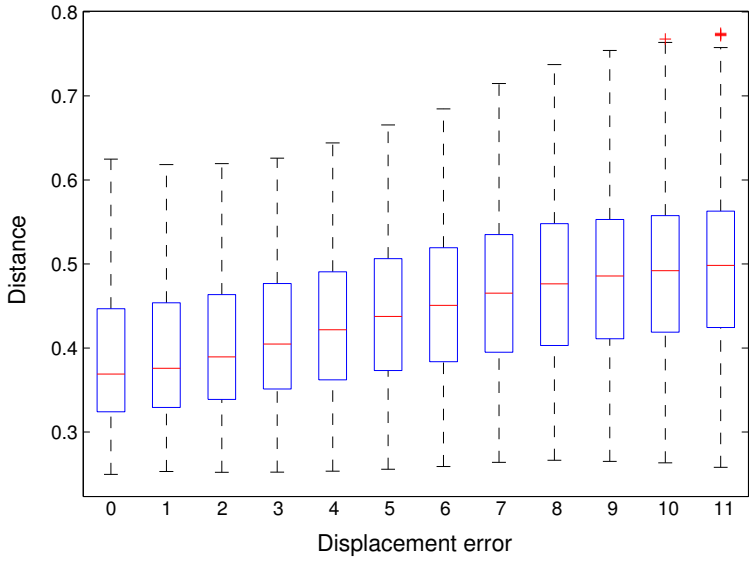


Figure 6.8: Distances between descriptions computed by varying their relative misplacement (horizontal axis) from the correct value (0 displacement) on a test football sequence.

i.e. the cost of an unpaired frame in NW is fixed. The parameter G gives also the possibility to tune the tradeoff between trusting the data (and the noise they contain) and having a smooth solution that strongly penalizes insertions of frame dropping. As with DTW, in case of strong noise or very weak signal, the solution will be biased toward the diagonal of the matrix: in case of DTW this is true as the diagonal of C is the shortest path between the two corners and, in the presence of uniform similarities, it will be the path that accumulates the smallest cost. In the case of NW, this depends on the value of G that, if it is set too small, will block the algorithm from discarding frames.

We compute the distance $d(\cdot)$ between two frames in NW as

$$d(m,n) = \mathcal{M}_{k \in B}(D(P_1^k(m), P_2^k(n))), \tag{6.7}$$

where $\mathcal{M}$ is the median that filters out the noise on the tracking data and errors due to occlusions between objects, and $B = O_{V_1(m)} \cap O_{V_2(n)}$, with $O_{V_1(m)}$ ($O_{V_2(n)}$) being the list of objects that appear in $V_1(m)$ ($V_2(n)$).

6.3.3 Computational complexity

The first step of the algorithm extracts the HOG description from each bounding box and updates the SSM by comparing each HOG against the last $2r_{max}$ descriptions. Since the information

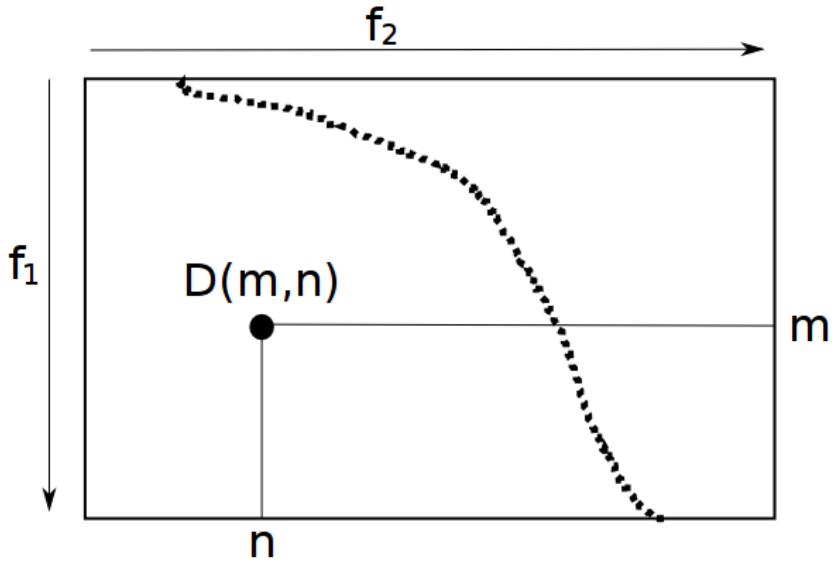


Figure 6.9: Given the matrix containing all the distances $d(m, n)$ between the frames f_1^m of the first video and f_2^n of the second video, the alignment algorithm looks for the path from the lower right corner (the end of both the videos) to the upper left corner (the beginning of the videos) with the aim of passing through the coordinates that correspond to the indexes of instants framing the same dynamic action (and hence from the same instant).

needed to compute a PHOG description is in the lower right part of the SSM of size (r_{max}, r_{max}), the rest can be discarded. Therefore the cost of this part is *constant* with time and *linear* with the number of objects in the scene. In the second part of the first step we extract PHOG description of each instant and for each track. The cost of this part is *constant* for each object and for each instant and is *linear* with the number of frames and the number of objects.

The descriptions of each object in each instant are given in input to the NW algorithm. The cost of each comparison is *linear* with the number of objects and *constant* with all the other variables, hence the cost of this step and of the whole algorithm is dominated by the cost needed to fill the matrix C_{NW} that is equal to the product of the number of frames of the two videos.

For long video streams the quadratic complexity of the algorithm can be bounded by limiting the size of the matrix C_{NW} obtaining a constant computational cost at each step at the expense of bounding the maximum misalignment that it is possible to recover to a fixed amount of frames. All the steps until the computation of the alignment depend only on the number of objects and since no information sharing is needed they may be computed in a distributed way directly across cameras.

With respect to the algorithm proposed in Dexter et al. [DPL09], where the description is composed by only one level, we have a penalty on the comparison of two frames that is linear with

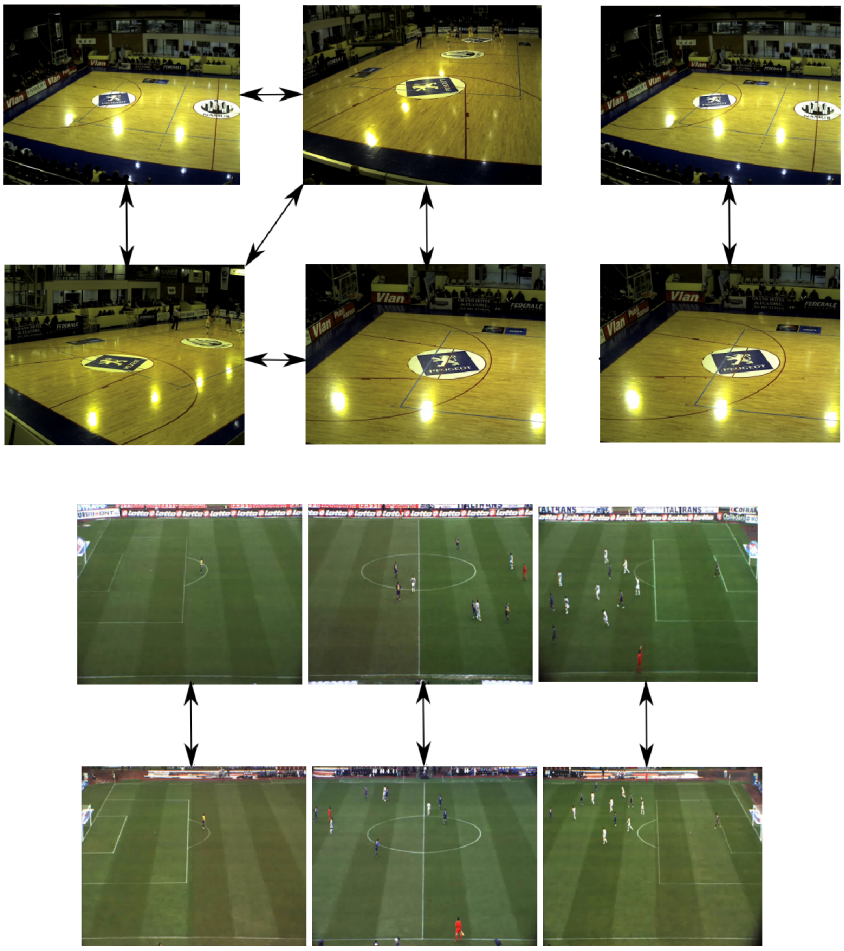


Figure 6.10: Sets of couples of videos in the experiments. Top left: general POVs APIDIS cameras. Top right: paired APIDIS cameras. Bottom: ISSIA football dataset.

the number of levels. However, since the levels are fixed, it is only a constant factor that does not influence the final complexity of the algorithm. Moreover, we do not need to compute the Laplacian for all the possible radius of the PHOG description as [DPL09], and, since all the parameters are fixed, all the steps in analysing an object terminate always in the same fixed time.

6.4 Experiments

6.4.1 Experimental setup

To evaluate and compare the alignment results of the proposed algorithm in real-world scenarios we test the algorithm on two public datasets, namely the ISSIA football dataset and the APIDIS basketball dataset (see Appendix A.2). The two datasets provide the annotations on the tracks that, since our focus is not tracking, have been used as input.

We have divided the APIDIS dataset in a set composed of sequences that share the same viewpoint and a set containing all the other combinations of cameras, which have different POVs (Figure 6.10).

To control and quantify the results, the data to align are created by misaligning the videos according to $t_w = \alpha t - \beta$, with $\alpha \in \{1, 1.1, 1.2, 1.4\}$ and β a constant shifting the frames by reducing the overlap up to 2/3 of the duration of the sequences. The model used is equivalent to a random frame dropping model as to simulate different values of α we remove frames from one of the two videos. All the tests have been repeated removing up to 200 frames by the end of the warped sequence in order to have a solution that is not in correspondence to the diagonal of C_{NW} and to obtain an unbiased estimation of the performances of the algorithm. We consider the starting point of the video in correspondence of the first object. For this reason, the APIDIS videos start with up to 500 frames of difference: setting β varies this displacement.

The following algorithmic parameters are the same for all the experiments. The HOGs have blocks of 2 cell of 16 pixels and 9 bins. The multiscale parameters and G have been chosen to minimize the error on the sequences of three players of the ISSIA dataset (i.e. a subset of a single video): $r_{min} = 20$, $r_{max} = 75$, $b = 1.2$ and $G = 0.2$.

Errors will be reported as the median error (in frames) w.r.t. the correct timeline. The median has been chosen to reduce the influence of big errors on a few frames (e.g. the first frames of two strongly misaligned videos). Note that, due to the monotonicity of the solution, only the outliers in the first or of the last frames of the videos are filtered.

We set as reference the errors reported in Dexter et al. [DPL09] that, using *frontal* cameras *shifted and not warped* (i.e. $\alpha = 1$) reports an error of 7.29 frames. Our aim is to work with different FOVs and general viewing angles with an error bounded by the same order of magnitude. In the next section we discuss the results of the comparison between our proposed approach and [DPL09], which is the only method in the literature that shares similar hypotheses to ours and thus the only meaningful comparison.

6.4.2 Comparative analysis

Because in Dexter et al. [DPL09] the input to compute the SSM makes the algorithm not suitable to work with different FOVs, in order to conduct a fair comparison of the individual steps we adopt their pipeline with the HOG description in input. In order not to modify the original pipeline we use as the input one object (player) for each sequence. We carry out the comparative analysis considering all the objects with a long continuous track in the football dataset, and the four objects with the longest continuous track from camera 1, 2, and 7 in the basketball dataset (combination used: camera 1 and 2; and camera 1 and 7).

The improvement of our pipeline is computed as relative difference:

$$E = \frac{1}{N} \sum_{i}^{N} \frac{E_l(i) - E_m(i)}{min(E_m(i), E_l(i))}, \tag{6.8}$$

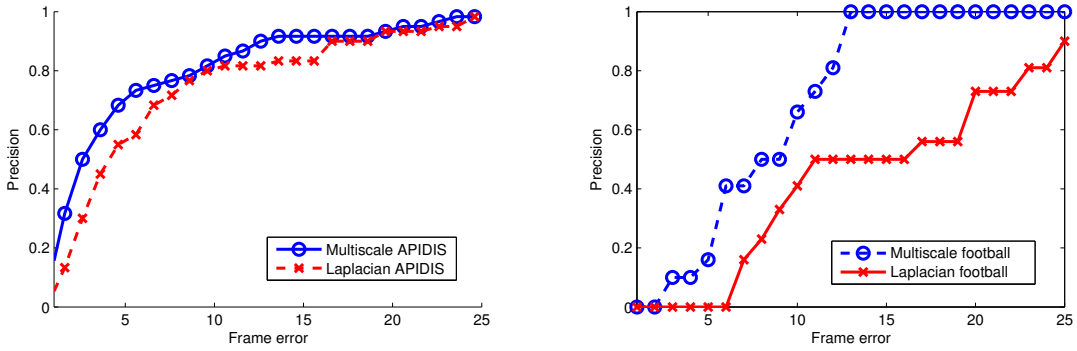
where N is the number of experiments, E_l is the set of the errors (in frames) obtained in the tests with the original method based the maximum of the Laplacian, while E_m contains the corresponding results obtained with our multiscale feature comparison. Based on the experiment described above we obtained a consistent improvement over the original algorithm, with $E = 1.49$ for the basketball dataset and $E = 0.96$ on the football dataset (see also Figure 6.11).

The second comparison is between the commonly adopted DTW and the method we propose based on NW (Figure 6.12). It is possible to notice a clear superiority of NW, which is mainly due to parts of the videos that have not got correspondence and for the effect of static scenes, on which NW is more robust. In Figure 6.13 are shown two examples of two warped timeline of two videos to show how NW is more stable and robust to ambiguous configurations retaining the ability to identify real shifts in the data. In complex configurations where the NW algorithm fail due to lack of sufficient information in the data, DTW usually fails similarly (see Figure 6.13 (b)).

6.4.3 Overall performances

This section discusses the results obtained on the full dataset and the analysis of the robustness of the proposed approach.

Table 6.2 summarizes the results of the proposed algorithm obtained with all the objects in the scene using different values of α; β is not reported explicitly as it has no significant effect. Fixing α and varying β has in fact produced a standard deviation of the error of 0.63 frames. A more critical parameter of our algorithm is the number of objects that are needed to align the video to a certain accuracy. Figure 6.14 shows the error for a different number of objects in ten random experiments. The algorithm starts being stable with five objects. This result is an upper bound of the error as objects were extracted randomly and could not be always in the scene at the same



(a) (b)

Figure 6.11: Comparison of our pipeline with the multiscale PHOG descriptor with the PHOG-Laplacian descriptor [DPL09] on a subset of the ISSIA (a) and APIDIS (b) dataset chosen to be compatible with [DPL09] (see text for details). The performance gap is wider with the more complex sequence of the APIDIS dataset.

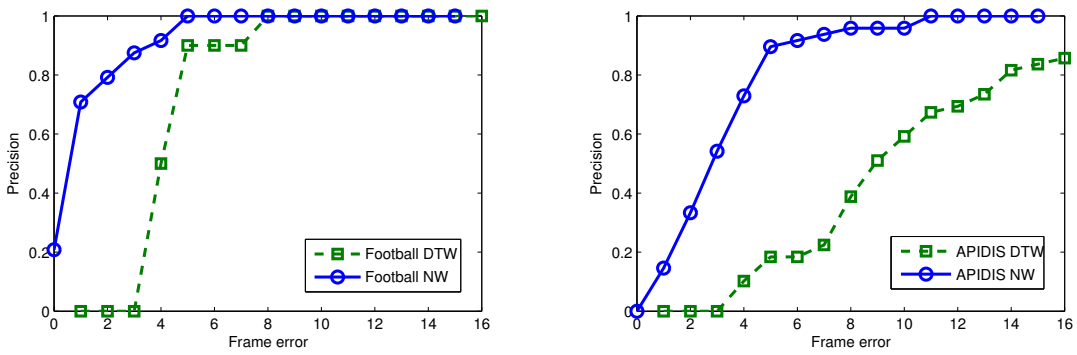
	α			
	1	**1.1**	**1.2**	**1.4**
APIDIS (camera 1 and 7)	1.00	1.00	1.50	1.50
APIDIS (all other cameras)	4.00	4.49	3.83	4.33
ISSIA	0.66	1.16	2.00	2.16

Table 6.2: Mean errors obtained with different warping parameters α.

time. Moreover, the results with few people also suffer from the occlusions by all the other objects that are still present in the video, even if in this experiment they are not considered for the alignment.

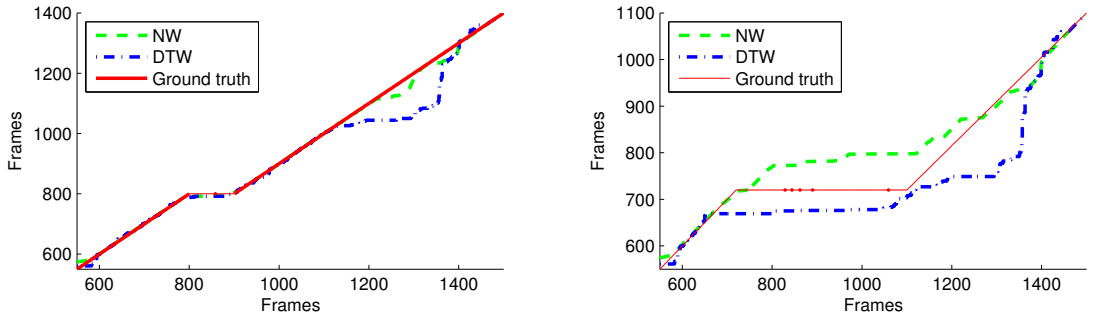
To analyse the robustness of the proposed algorithm to work with noisy detections we modify the object bounding boxes by varying their size and position with uniform random noise. To reach significant results we test the noise with $\alpha = 1.4$ using β equal to 1/3 of the video. Figure 6.15 shows the results: the noise is the maximum displacement due to a given noise in two consecutive frames, and is expressed as a percentage of the size of each side of the bounding box of an object. The break point of the proposed method in this difficult setting is above 10% of noise (see Figure 7.14). Notice that the original annotation data themselves are not always accurate and therefore this amount of noise induces considerable errors that can be usually attenuated by dynamic filters. As a reference an unfiltered Mean-Shift tracker [CM02] applied on a random subsampling of sequences with no id-switch has shown an error comparable to 4% of uniform noise (see Figure7.14).

Another source of noise is given by association errors both during the tracking and between the



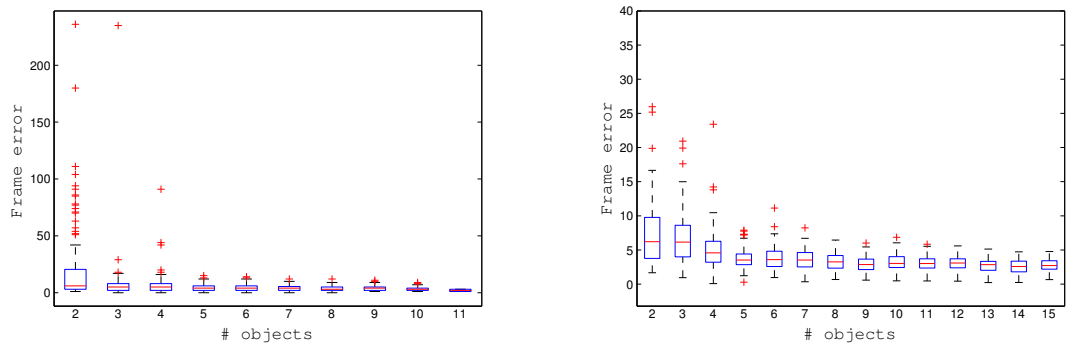
(a) (b)

Figure 6.12: Comparison of NW with the commonly used DTW on a subset of the ISSIA (a) and APIDIS (b) dataset.



(a) (b)

Figure 6.13: Two alignments from real noisy data computed with DTW and NW. (a) it is visible how the parameter G affects the action in ambiguous situation by regularizing the solution, but it does not compromise the ability to adapt to real signal in the data; (b) in more critical scenarios with greater noise and big gaps in the data w.r.t. the length of the timeline both the algorithm may fail in similar ways.



(a) (b)

Figure 6.14: Comparison of the performance of the proposed algorithm with a varying number of objects in input for the alignment. The errors are generated using randomly selected objects of different cardinality and different misalignment parameters. (a): results for the APIDIS dataset. (b): results for the football dataset. The results shown have to be considered an upper bound to the error of the algorithm (see text for details). Note that, to include the outliers in the visualization, the vertical scales of the two plots are different.

camera views. Our experiments in this setting have shown that, thanks to the filtering effect of the median used to compare the frames, with the 20% of wrong associations, the algorithm is able to align all the videos in the football dataset and the 92% of the difficult sequences of the basketball dataset introducing at most 3 frames of error. This result is significant since our experiments have shown that, by using a brute force approach on subsets of data to minimize the functional of NW, it is possible to ignore the input association obtaining the equivalent of the 15% of errors on the ID switches.

6.5 Discussion

We presented a general method for video alignment based on the observation of the actions of multiple objects that considers high level information on the appearance of the dynamic of multiple objects to obtain a constraint free algorithm.

The general idea of using the SSM to have a view-invariant representation was exploited in Junejo et al. [JDLP11] for action recognition and in Dexter et al. [DPL09] for video alignment. Unlike [DPL09], where the analysis is performed at feature level (the input is the tracking of a set of points extracted from the observed videos), we use explicitly multiple objects appearance and exploit implicitly and simultaneously information from the pose of the object and its motion. An alignment algorithm tested with close up of a single actor with static background, that considers multiple descriptors (including HOG) combined with SSM was proposed in Junejo et

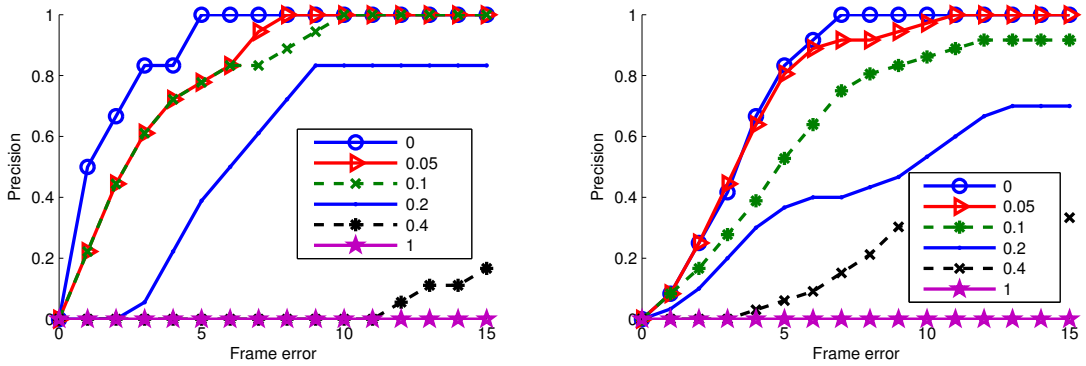


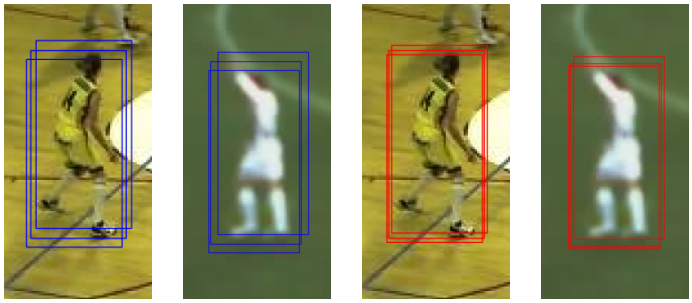
Figure 6.15: Performance of the proposed alignment algorithm when adding uniform random noise on the position of the bounding box. The range of the random movements is reported in the legend as a percentage of the size of the sides of the bounding box. (a) Errors on the football dataset; (b) errors on the APIDIS dataset.

al. [JDLP11] together with the action recognition method. However, our feature extraction from the multiple SSM, the algorithm that compares actions and instants, and the alignment method differs both from [JDLP11] and [DPL09].

Unlike methods based on assumptions on geometry [PCSK10, CSI06], we remove constraints such as planar scene, known geometry, restrictions on the time misalignment. The cost for relaxing the assumptions on the geometry, the camera configurations and the restrictions on the temporal misalignment is that we reach a frame-level accuracy, instead of a sub-frame accuracy [PCSK10, CSI06].

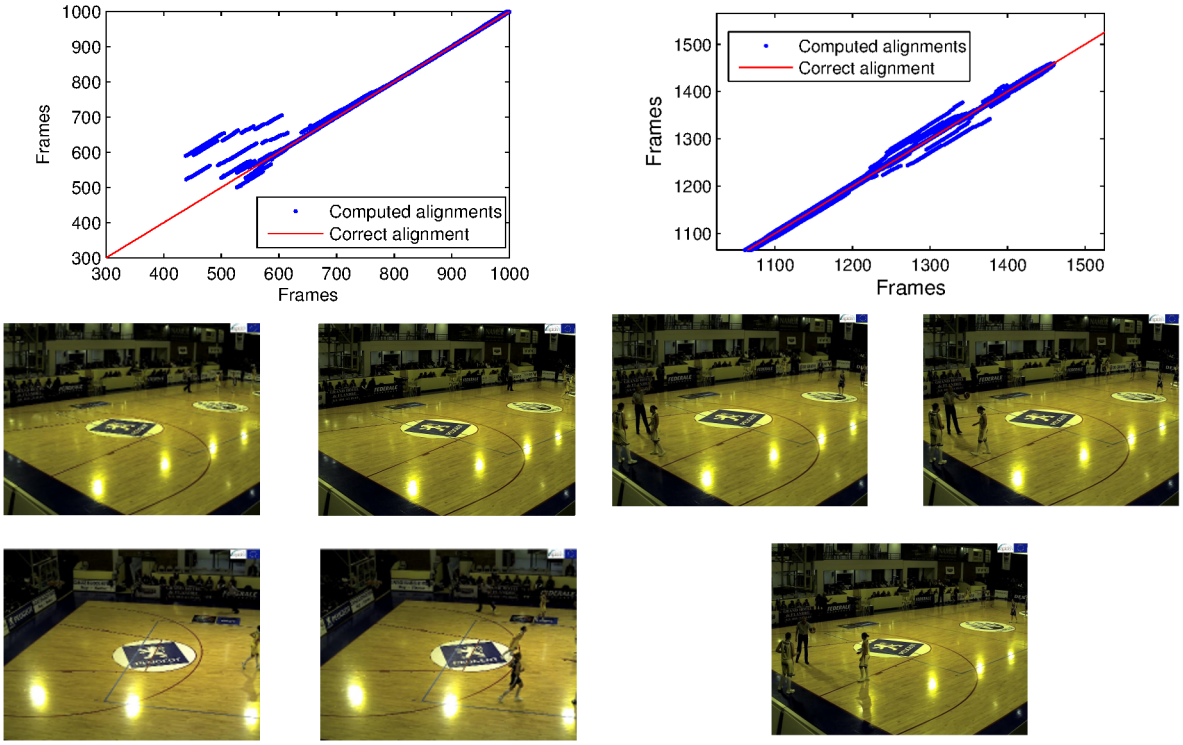
A source of error for the proposed algorithm is obviously the absence of relevant actions in the time window observed for the alignment. Moreover, when there are just a few objects with similar actions in different time intervals, if the relative order between actions is preserved there can be a mismatch in the alignment (see as example Figure 6.17). Overall, the alignment of videos with very different viewpoint and frequent occlusions may need a considerable temporal overlap: as an example the videos of the general POVs APIDIS can be aligned with an error bound to ten frames when the temporal overlap is at least 2/3 of the timeline. Simpler videos can be aligned even if they have larger misalignments. This characteristic is shared by DTW and NW: if the signal is not strong enough with respect to the shift, the cost of moving away from the diagonal of C_{NW} could be dominant in the computation of the solution.

A description of the method and its experimental results is under review [ZCO13].



(a) (b)

Figure 6.16: Shifts from the correct position of the bounding boxes that can be observed in two consecutive frames with two level of noise. (a) 10% of noise; (b) the level of noise registered with a mean-shift tracker (≈4%).



(a) (b)

Figure 6.17: Configurations where the accuracy is reduced in presence of noise due to ambiguous data in the scene. (a) Similar actions and only an object in common between the views. (b) In a scene that is static for a long period all the frames are similar.

Chapter 7

Multi-view object association

In this chapter we analyse the problem of finding the correspondence between the same person (or a more general articulated object) between multiple cameras that share a part of their field of view. The chapter is organized as follows. Sec. 7.1 introduce the setting, Sec. 7.2 summarizes the state of the art related to our work, Sec. 7.3 describes the proposed method, Sec. 7.4 reports the experimental analysis.

7.1 Introduction

Multi-camera systems [TC11] have populated the computer vision literature of the last decade, especially in video-surveillance, traffic monitoring, human motion analysis domains. One of the building blocks of such systems are algorithms able to associate the same person (or more in general the same object) between multiple cameras. This general task can be named as *people association* or *people matching* in a general context, however it can be found as *consistent labelling* [KS03] if the cameras are more than two or *re-identification* [BVC11, BMPI05] if we aim to find correspondences between people observed in different instants.

With ”traditional” multi-cameras system designed beforehand, where all cameras have been positioned appropriately and possibly calibrated [BE02, KS03, ST03] we can exploit geometrical constraints to solve this task. However nowadays we may exploit additional ways of gathering videos besides static or PTZ cameras. Both off-line videos acquired from uncalibrated camera systems and amateur videos acquired by hand-held mobile devices or mobile cameras are easily accessible and may offer useful insights to the analysis of the overall scene, but pose new challenges.

Inspired by this new perspective, we start considering a heterogeneous multi-camera system where the different points of view may have a different nature: hand-held devices are side by

Method	FOV	Feature	Camera motion	Geometry
Proposed	overlapped	behaviour	free	general
[BE02]	overlapped	geometry	static	general
[MD01]	overlapped	geometry & appearance	static	planar
[DT01]	overlapped	geometry	static	general
[CGO00]	overlapped	geometry & appearance	static	general
[KS03]	overlapped	geometry	static	planar
[ST03]	overlapped	geometry	static	planar
[MMH06]	overlapped	appearance	free	general
[PGX+08]	overlapped	appearance	static	general
[CCP08]	overlapped	geometry	static	planar
[KCM03]	overlapped	geometry & appearance	pan-tilt-zoom	planar
[FBP+10]	general	appearance	free	general
[BCPM11]	general	appearance	free	general
[BVC11]	general	appearance	free	general
[BMPI05]	general	appearance	static	general

Table 7.1: Overview of the requirements of the algorithms from the literature for matching (FOV = overlapped) and re-identification (FOV=general) and comparison with the proposed method. Methods based on geometry require or compute calibration.

side to fully calibrated systems, different temporal resolutions have to be considered, diverse optics produce distortions and colour information will vary. In this case consistent labelling based on the knowledge or the inference of the scene geometry may not be so effective, and methods relying on the invariance of appearance descriptors will in general not work.

In this chapter we present our study on a method for people association between cameras with overlapped fields of view based on the analysis of the dynamic of the appearance. W.r.t. the algorithms in the literature, it has the advantage of not relying on the geometry or on the static appearance, so that it does not need any kind of calibration, training, assumptions on motion of the camera or on the structure of the world.

We assess the method on different types of video, framing sports events and a typical outdoor video-surveillance environment. We perform an extensive analysis to show its robustness to errors and fragmentation of the initial tracking, to a decreasing FOV overlapping, to heterogeneous video qualities (including different frame rates and different resolutions), and to the presence of hand-held moving cameras.

7.2 State of the art

The literature related to matching is vast, Table 7.1 summarizes the methods for people matching, highlighting their main features, in comparison with the proposed approach.

This overview reports the contributions from the state of the art for people matching with overlapped FOVs and to three different problems that are related to it:

- People matching with non overlapped FOVs (*re-identification*).
- Feature matching.
- Action recognition.

People matching and multi-camera tracking

Articulated objects matching between cameras with overlapping fields of view is usually addressed using geometrical constraints from calibration information [MD01, BE02, DT01, CGO00] or from an estimate of it (e.g. [CCP08, KS03, ST03]).

The geometry of the camera system can be estimated in a training step exploiting a person that moves on a planar environment [KS03, CCP08]. Once the fundamental matrix of the camera system is available it is possible to associate people depending on the geometry and to refine the results exploiting multiple observations in time. Moreover it is possible to handle cases where on one camera the observed people are segmented together [CCP08].

People association between Pan-Tilt-Zoom (PTZ) cameras can be handled on the basis of a multi-camera tracker that assumes a planar environment [KCM03]. While the association is done with the geometry, tracking and motion is estimated with a method based on background modelling and a joint model of motion and appearance.

An alternative to geometry is to rely on appearance. Colour similarity is a common choice: in this case it is usually needed a *colour transfer* function between cameras to compensate for the colour deviations [JJ08] due to different lenses, sensor and settings (e.g. white balance, gamma, contrast and saturation). However such calibration is reliable only with fixed lightning and may not be useful in outdoor settings where both the intensity and the temperature of the light change.

Re-identification

Re-identification is a problem related to people matching, that addresses the case where the fields of view of the cameras are not overlapped: in this case the algorithms aim to capture information that does not change over time (e.g. colours of clothes) rather than instantaneous measures

(e.g. pose), hence colour and texture information is mainly extracted [JJ08, PGX+08, FBP+10, BMPI05].

Rather than using global colour information that includes the background, it is possible to compute a static weighting mask related to the probability of each pixel of a bounding box to be background to weight the contribution of different pixels while computing the descriptor [PWS07].

Bird et al. [BMPI05] instead choose as description the median HSV values within ten horizontal stripes of each bounding box and removes the background using change detection to segment it.

Farenzena et al. [FBP+10] extract more complex features (Maximally Stable Colour Regions, Colour Histograms, Recurrent High-Structured Patches) and select a different subdivision of the the mask related to body parts exploiting symmetry principles.

A more complex approach [BVC11] proposes to describe each person by means of a 3D model where are stored local information on the appearance. The model is built incrementally adding information on the visible parts as soon as new frames are received and the orientation of the person w.r.t. the camera is detected [BVC12].

Action recognition

To the best of our knowledge, in the literature there is no work aiming to solve the problem of matching articulated objects using high level behaviour analysis or any other kind of dynamic analysis. Instead, behaviour analysis is exploited in the field of action recognition [Pop10]. The main challenge for action recognition algorithms is to be able to recognize the same action from different points of view. Our method shares this challenge since it aims at describing an object by means of its behaviour, with matching purposes.

The approaches in the literature exploit shape and silhouette of objects [BD01, CCCL06, SB08], their evolution over time (e.g. space-time shapes [GBS+07]), the analysis of a fixed grid of a Region Of Interest (ROI) with descriptors of shape and of it evolution [TS08, JDLP11, KZP08, TH08], or consider sparse image descriptors [DRCB05, LMSR08, SVG08].

Robustness w.r.t. strong viewpoint differences can be addressed by reconstructing and analysing the 3D pose [YKS08, WBR07], considering an explicit geometrical model, by training the algorithm with a dataset of multiple poses or considering explicitly the problem in the training step (e.g. with transfer learning [FT08]). Alternatively it is possible to aim to be independent of the viewpoints without taking into account the geometry of the scene or a specific training set [JDLP11]: in this case the authors exploit the Self Similarity Matrices (SSM) of a descriptor (e.g. optical flow, Histogram of Oriented Gradients) computed in a ROI over time. Our work follows this view, subscribing the claim this descriptor, by encoding the recurrence of the same configurations of the object and not the configurations by themselves, is robust w.r.t. view variations.

Feature matching

Finally, the more general problem of feature matching has been applied to image retrieval, motion estimation and geometry computation. Usually it refers to spatial features (such as corners, SURF [BTVG06], SIFT [Low99], Gradient Location-Orientation Histogram [MS05]), as opposed to this work where the features to be matched could be seen as spatio-temporal. Space-time features [Lap05b] have seldom be used in feature matching. Local Self Similarity matrices [SI07] have been proposed to match data between images and videos and their authors test it for action detection, image and video retrieval and retrieval from sketches. As for image points matching it is worth mentioning [TVG00, ZDFL95, SZ97], where the matches are computed using geometrical information extracted from the image (e.g., by estimating the fundamental matrix) or obtained as input from a fully calibrated system. Points matching can be posed as a machine learning classification problem [LPF04] based on the generation of synthetically warped images. In Lepetit et al. [LLF05] the same approach is applied with randomized trees to apply the method in real time.

Another class of matching algorithms is based on the application of spectral methods for the analysis of the adjacency matrix of the graphs associated to the matching. Shapiro et a. [SMB92] proposed to compute an adjacency matrix from each set of points (i.e. for each image) and the matches are found exploiting the modal matrices of the two graphs. The same approach based on spectral analysis can be combined with an Expectation-Maximization algorithm to be more robust w.r.t. points that have no correspondence [CH03]. Moreover to compute the adjacency matrix of the graph can be used different weighting function [CH03, DIOV06], instead Wang et al. [WH06] proposed the application of kernels to embed the desired invariance to transformations (e.g. translation, rotation). This class of methods has been extended to consider a one to one association between the objects and to strengthen the spectral analysis of the graph exploiting the geometrical transformations between the sets of points [TLW$^+$07].

A seminal work in spectral matching is [SLH91], which proposed to solve the problem analysing the adjacency matrix of a graph that has the points as nodes and arcs only between nodes of different images. Since in this case the ideal solution is an orthonormal matrix (in the case where all the points have a correspondence), the algorithm forces this property on the adjacency matrix to obtain a solution that respects both the *principle of similarity* (similar objects should be paired) and the *principle of exclusion* (for each point is allowed only one match). Despite in the first proposal the algorithm was designed to match points minimizing the euclidean distance on the image plane (obtaining an algorithm able to work only with very similar POV), this method has evolved to consider wide baseline matching by merging appearance and geometrical information [Pil97, TC02]. Delponte et al. [DIOV06] suggested that, using a pipeline derived from [SLH91], it is possible to exploit only the appearance (e.g. the SIFT description) to obtain a good set of matches for large baseline image points. The latter class of methods shows a high degree of flexibility and very good performances in many settings, and for these reasons has been adopted in this work.

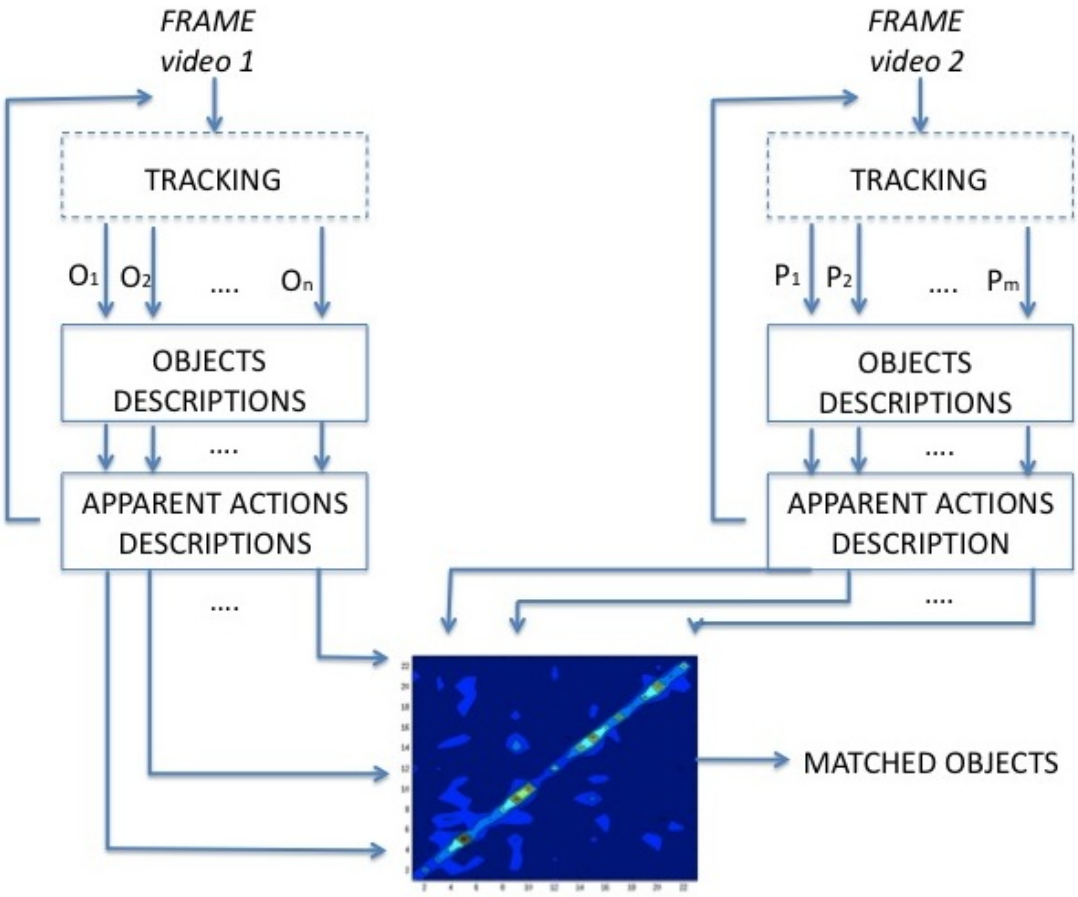


Figure 7.1: Main steps of the proposed method: given two videos and the data of the tracker, we extract an appearance-based representation of the behaviour of each object in the scene to compute the associations between the two views.

7.3 Proposed method

We propose a real-time method for associating articulated objects in different views in a constraint-free setting. The method is *independent on the geometry*, *adapts to camera motion*, *does not require a training procedure*, and can be applied to different situations and to scenarios of different complexity. The main novelty of our approach is to rely on the peculiarity of objects behaviours, computing a view-invariant representation of the action of each object in the scene and matching it with objects from other views. Starting from tracking information, we compute the matches between the objects in two videos exploiting the evolution of the appearance of the objects over the time (i.e., their ”behaviour”), making no assumptions on the geometry (e.g. planar world, static cameras, known geometry) and having weak requirements on the overlap of the Fields Of

View. The intuition we pursue is that meaningful features in a temporal behaviour pattern are more robust to view-point changes than simple appearance.

We exploit the Self-Similarity Matrix as a description of the behaviour of the objects [JDLP11], and extract from it local temporal descriptions based on the Generic Fourier Descriptor (GFD) [ZL02]. We then model our matching problem with an undirected bipartite graph and look for pairings between nodes via a spectral approach, by adapting the method proposed in Scott et al. [SLH91] to our needs. Figure 7.1 summarizes the proposed pipeline.

Since we rely on the evolution of appearance, the method is robust when associating similar objects (as it is in cases where people wear uniforms). Also, the algorithm has proved to be very effective in the association of people behaving similarly (and potentially causing ambiguities) as in video surveillance. On this respect Figure 7.2 reports an example of two sequences of people correctly associated, in spite of the similar behaviour.

This section describes the main steps of the proposed method. Figure 7.1 shows the pipeline of the algorithm: given two online video streams from two cameras and tracking information, for each object we extract a representation of its appearance and we employ it to update a view-invariant description of the appearance evolution over time. This information is then combined in an affinity matrix that encodes the similarity between behaviours observed in two different views that is used to compute matches between pairs of similar behaviours (and then, possibly, of corresponding objects).

7.3.1 Objects description

In the first step of the algorithm we want to extract a visual description of each observed object at a given time instant. Given a bounding box $T_i^j(t)$ extracted from the video j at the frame t for the i^{th} observed object, we compute a description $D_i^j(t) = \phi(T_i^j(t))$ that models its appearance.

Function $\phi(\cdot)$ maps a bounding box to a description that has to be robust w.r.t. tracking errors; robust to illumination changes; accurate to describe the configurations of the object; and descriptive enough to recognize the same configuration of the object in different positions and with different backgrounds.

We adopt Histogram of Oriented Gradients, in the experimental section we will compare it against alternative choices.

7.3.2 Apparent action description

For a given object tracked over time, we compute a Self Similarity Matrix of its HOG representation in time (see Sec. 6.3.1).

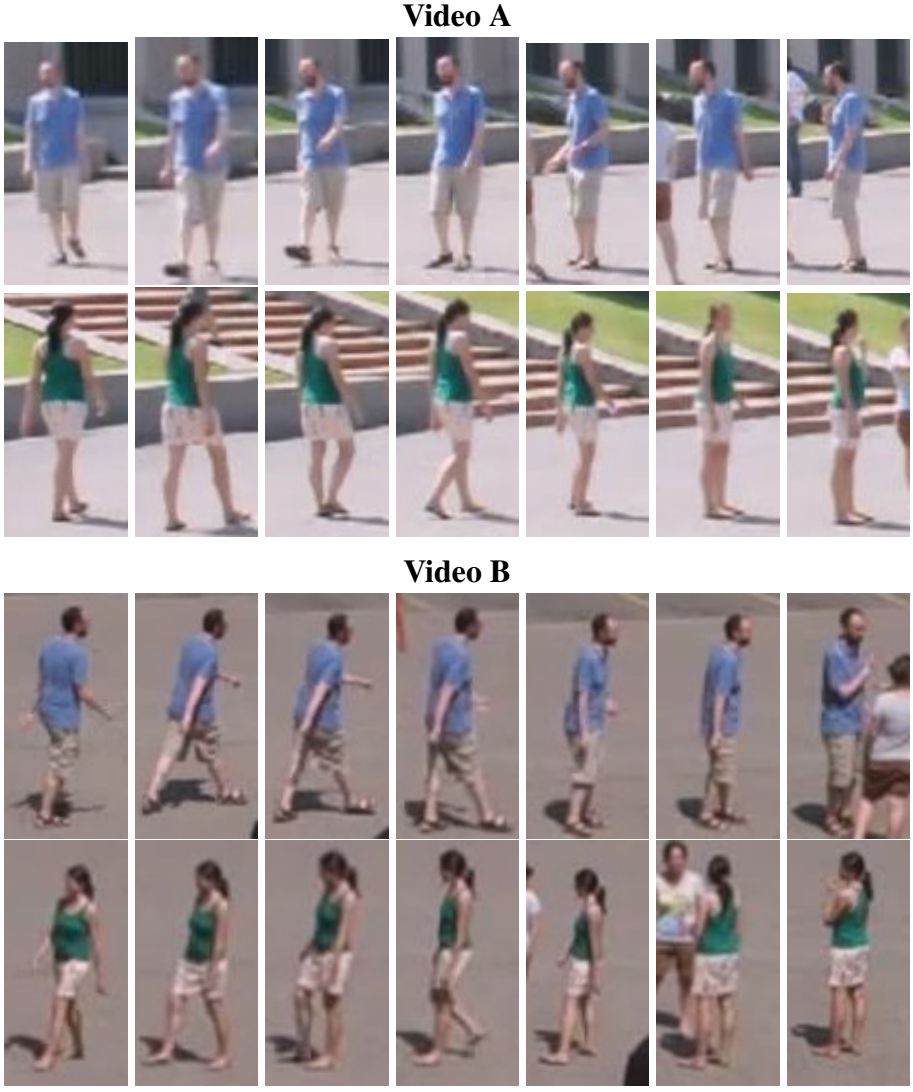


Figure 7.2: Two people that are assigned correctly by our method, despite the similar behaviour (walking and then standing), some partial occlusions, and a challenging background.

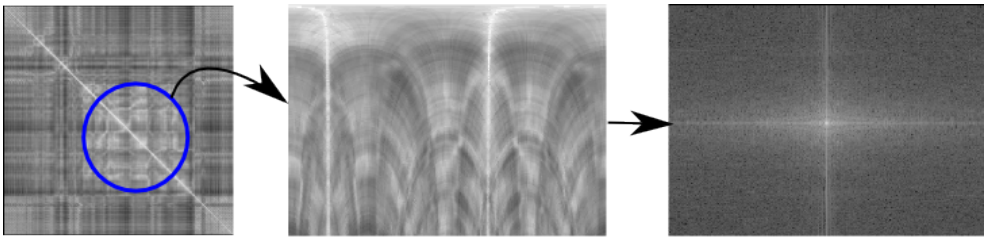


Figure 7.3: Pipeline of the computation of the Generic Fourier Descriptor [ZL02]. The SSM "image" is first converted in polar coordinates and transformed using the DFT. The frequencies are stacked in a vector and normalized to achieve scale invariance (see text for details).

It is worth noticing that the computation of SSM may be carried out online, as a new frame becomes available: at each time instant t we simply add a row and a column to the matrix S.

Then, the apparent behaviour of the object i on the video j at time t considers the last T_s seconds. To this end we extract a description from a circle of diameter T_s centred on the point $(t-\frac{T_s}{2}, t-\frac{T_s}{2})$ of S_i^j (see Figure 7.3, left). The diameter T_s corresponds to the size of the time window that we are interested in modelling: large diameter values include frames that are far away in time from each other, smaller values will limit the analysis to frames that are near in time. The value of T_s also controls the minimum amount of frames that we need to observe to be able to extract the first description.

In the case the input videos have different frame rates, T_s has to be modified to have the same temporal extent.

Since, T_s also affects the algorithm complexity, we describe the circular portion of the SSM by means of a Generic Fourier Descriptor (GFD), which is able to capture the apparent behaviour from a smaller T_s (a comparison with other descriptions is given in Sec. 7.4). The GFD is the normalized Fourier transform of the considered area of the image converted in polar coordinates: given a SSM S it is first transformed w.r.t the center (t, t) in its polar coordinates S_p that is the plot of the polar matrix in the Cartesian space and it is then transformed using the two dimensional DFT (see Figure 7.3), obtaining

$$\mathcal{S}_f(\rho, \phi) = \sum_{r}^{R} \sum_{i}^{K} S_p(r, \theta_i)^{j2\pi\left(\frac{r}{R}\rho + \frac{2\pi i}{K}\phi\right)}, \tag{7.1}$$

where R is the radial resolution of the sampled circle and K is its angular resolution. $\mathcal{S}_f$ is then unfold row-wise into a feature vector $P_i^j(t) \in \mathcal{R}^{RK}$, the description of an apparent behaviour, whose first value is normalized by the area of the bounding circle of the considered shape and all the other values are divided by $\mathcal{S}_f(0, 0)$.

7.3.3 Objects matching

We model the problem of computing the associations between the N_1 objects of the first video and the N_2 objects of the second one with a undirected bipartite graph $G = (V, E)$ where each node represents one object and each edge is weighed according to the similarity of the corresponding nodes.

At the instant t our objective is to select the sub-graph of G $G_s = (V, E_s)$ such that each vertex has at most degree 1 that minimizes

$$\arg\min_{G_s} \frac{1}{|E_s|} \sum_{(P_i^1, P_j^2) \in E_s} \frac{1}{t} \sum_{c=0}^{t} ||P_i^1(c) - P_j^2(c)||_2 \tag{7.2}$$

maximizing at the same time the size $|E_s|$ of the graph.

The correct graph can be encoded in an adjacency matrix $M \in R^{N_1 \times N_2}$ s.t.

$$M(i,j) = \begin{cases} 1 & P_i^1 = P_j^2 \\ 0 & otherwise \end{cases} \tag{7.3}$$

where $P_i^1 = P^2$ is true iff P_i^1 and P^2 are the descriptions of the same object observed by the two cameras. M has an orthonormal sub-matrix whose size is equal to the rank of M and it is composed by all the rows and columns whose associated objects have a match.

Since the computation of the best set of pairings minimizing a distance function has combinatorial complexity, we have to apply a heuristics to obtain an approximate solution in a tractable time.

A sub-optimal solution can be obtained enforcing the properties of M on the graph adjacency matrix using spectral methods (e.g. [SLH91, DIOV06])

Computation of the adjacency matrix

We compute an adjacency matrix A of size equal to the number of the observed objects N_1, N_2 in the two videos, such that $A(i_1, i_2) = Dis(P_{i_1}^1(t), P_{i_2}^2(t))$, where i_1 and i_2 are indexes that run respectively on the objects of the video 1 and 2.

The distance function Dis may be, for instance, the L_2 norm, but in our setting a source of complexity are occlusions, in particular those affecting one view only. To tackle this problem we formulate the following similarity function

$$A(i_1, i_2) = Dis(P_{i_1}^1(t), P_{i_2}^2(t)) = \frac{\sum_{t=0} w_{12}^t ||P_{i_1}^1(t) - P_{i_2}^2(t)||_2}{\sum_{t=0} w_{12}^t} \tag{7.4}$$

where w_{12}^t is a constant value w_o if one of the descriptions has been computed on information that are affected by an occlusion, 0 if at least one of the descriptors cannot be computed (i.e. if the object is not visible in the considered time frame) or 1 otherwise. The matrix can be computed online by storing a matrix containing the sums of w_o for each couple and another one that sums the weighted distances.

Since the estimated matrix A will be very different from the ideal M we may force some properties of M over A as follows. We first consider a new adjacency matrix with a Gaussian weight as

$$A^w(i_1, i_2) = e^{-\frac{A(i_1,i_2)^2}{2\sigma^2}}. \tag{7.5}$$

We then approximate the matrix A^w to an orthonormal matrix as M: this can be achieved using and SVD decomposition

$$UDV^\top = A^w \tag{7.6}$$

to recompose the matrix in A^I by replacing the matrix of the singular values with the identity

$$A^I = UIV^\top. \tag{7.7}$$

In Figure 7.4 we show an example of the visual results of the full procedure of Gaussian weighting and orthogonalization on two adjacency matrices of two different tests, a simple (on the right column) and a more complex one (left).

The width of the Gaussian kernel σ may be selected so that most of the values fall beyond 3σ, resulting in a negligible value. More precisely we set σ to one third of the distance of the mean 5^{th} most similar object. This non linear sparsification using the Gaussian scales the matrix such that only the most similar matches are likely to be considered, and controls the warping of A during the orthogonalization process.

Inpainting for addressing occlusions

A possible alternative to manage occlusions is to discard the data that are affected from them, and to fill the SSM under the assumption that, while an object is occluded, it continues the same activity observed in the previous and in the following instants, or it make a transition between two actions that can be modelled (i) as a repetition of some previous behaviour (ii) as a smooth transition between the two states.

Our idea is to apply an inpainting algorithm to reproduce consistently both the structures and the texture of the considered SSM in the areas that are effected by occlusions. We have studied three inpainting algorithms [Tel04, BBS01, CPT03] whose code is available online. While using the OpenCV implementation of [Tel04, BBS01] we were not able to reproduce consistently the texture of the observations, we have obtained realistic results with the implementation [1] of [CPT03]

[1] http://www.cc.gatech.edu/grads/q/qszhang/project/inpainting.htm

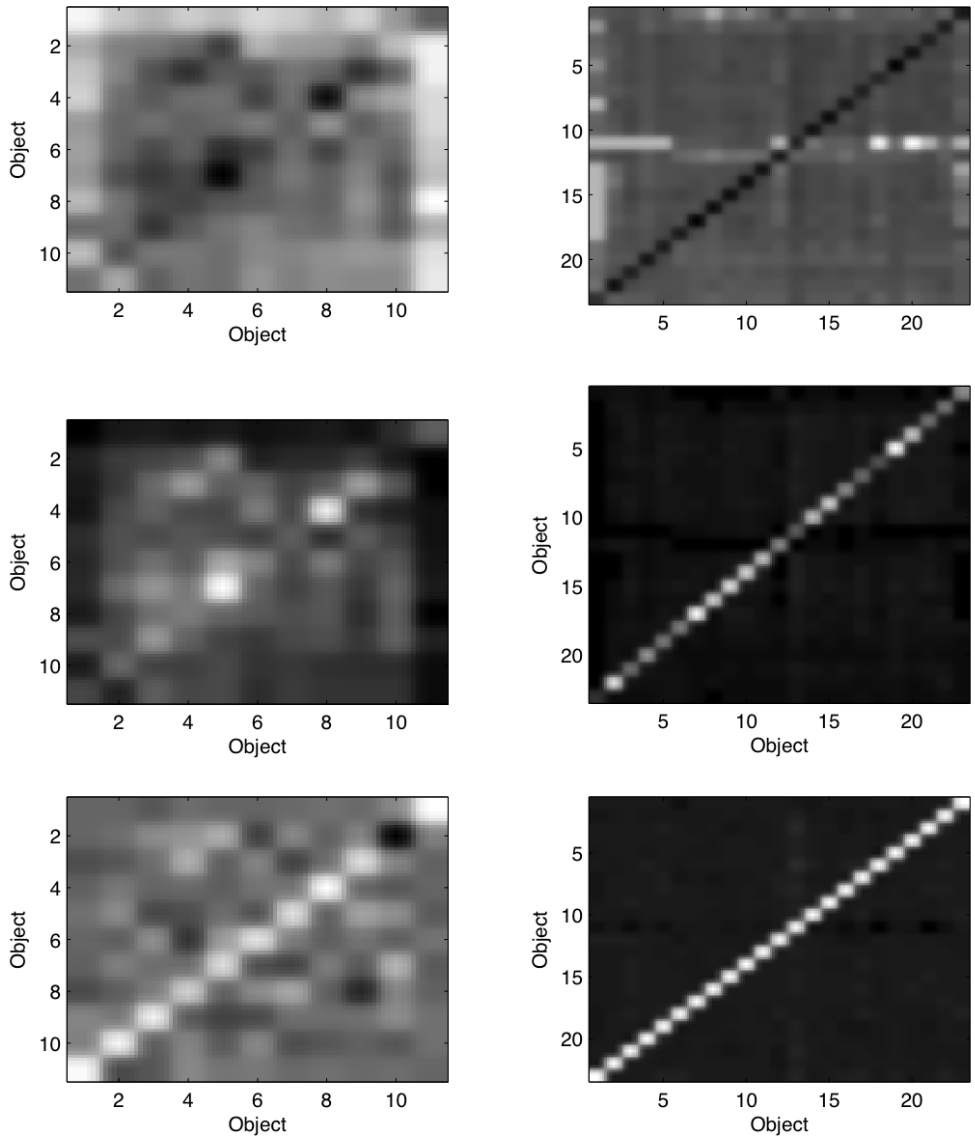


Figure 7.4: Examples of the effects obtained with the Gaussian weighting and with the orthogonalization on a difficult example (first column) and on a easier example (second column). On each column it is possible to observe (top to bottom) the matrices A, A^w, A^I containing the similarity values. The objects are ordered so that a red anti-diagonal is the right solution. It is possible to observe the strong effects of the orthogonalization, that helps to find a reasonably good solution even with a weak input.

(see Figure 7.5).

However the algorithms have proved to be too slow to process in reasonable time multiple SSM and their results have not been included in the experimental section.

Matching

Given an adjacency matrix A^* two objects P_i^1 and P_j^2 match if:

$$A^*(i,j) = \max(A^*_{i,\cdot}) \wedge A^*(i,j) = \max(A^*_{\cdot,j}) \tag{7.8}$$

that is, we have a match between P_i^1 and P_j^2 if the value $A^*(i,j)$ is the maximum in its row $A^*_{i,\cdot}$ and its column $A^*_{\cdot,j}$.

We recall A^I forces the associations between as many objects as possible, thus it implicitly assumes all objects have a match, then it produces a higher number of matches to the price of a possible precision loss. Such loss becomes significant if the number of true matches is much smaller than the rank of the affinity matrix (i.e., for very different FOV). In those cases it would be better to compute A or A^w instead than A^I.

Once we have extracted all the matches with the criterion of Eq. 7.8, we sort the results w.r.t. an approximation of the Signal to Noise Ratio (SNR) of the observed association. Our idea is to compute the SNR extracting the ratio between the similarity value of the considered match (signal) and all the values on its row/column (noise):

$$SNR(A,i,j) = \frac{A(i,j)}{\frac{\sum_m^{N_2} A(i,m) + \sum_m^{N_1} A(m,j)}{N_1+N_2}}. \tag{7.9}$$

Values with a higher SNR have more probability to be correct matches, and will be proposed first, while values with a low SNR are the one that are likely to be errors and will be proposed later.

7.3.4 Computational complexity

The proposed pipeline has a low computational cost, and can perform in real-time. As for space complexity, the method needs to store in memory: a buffer containing the last T images in the bounding boxes of each object; an SSM for each observed object of size $(T \times T)$ (i.e. the most recent information); a matrix whose dimensions are the number of the observed objects in the last frame that is needed to accumulate the mean distance of all the objects observed. Since all the objects which are lost by the tracker can be discarded, and the SSMs need only the information from the last T frames, the memory requirements are constant w.r.t. the length of the video, and grow as $O(N_1 N_2)$ with the number of objects observed in the last frame of the two videos.

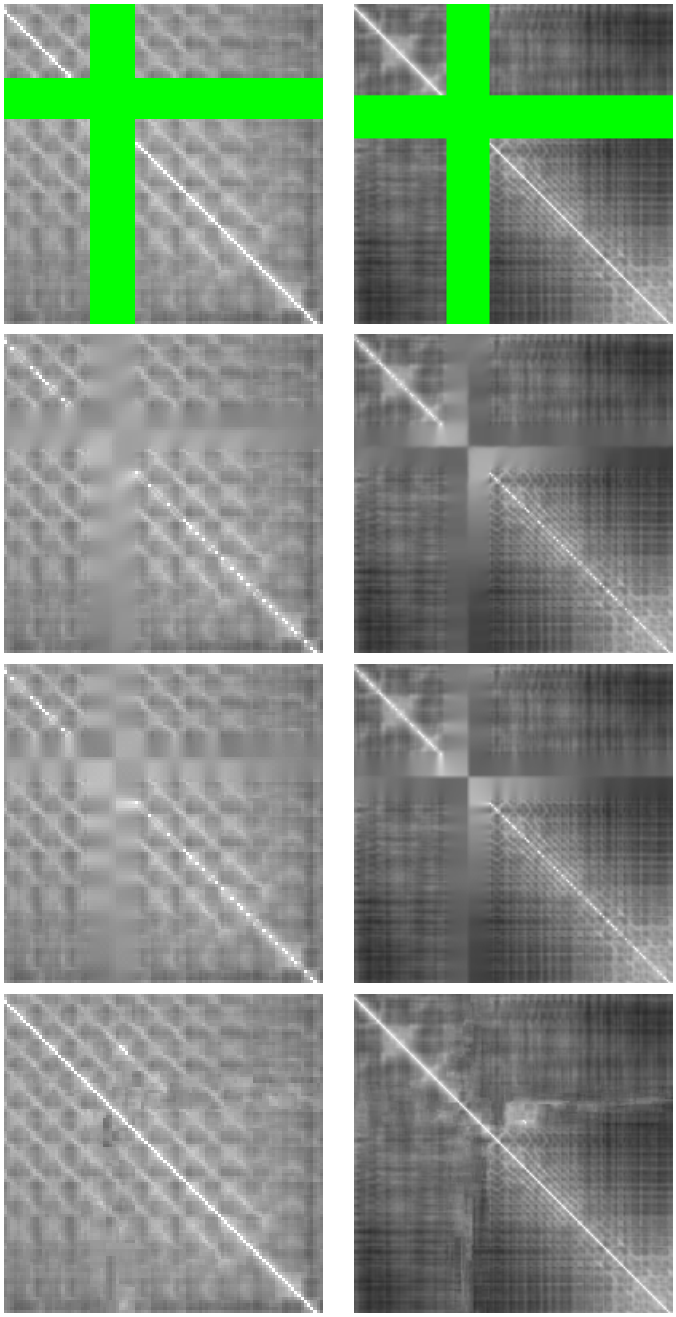


Figure 7.5: Results of the three inpainting algorithms tested. The first row shows two input examples, the second row are the results obtained with [BBS01], the third one with [Tel04] and the last one [CPT03]. It is clear how the first two methods are unable to reproduce the texture and focus only on the structures, while Criminisi et al. [CPT03] give good results reproducing both of them.

Since all the matrices have a bounded size, the time complexity is linear w.r.t. the length of the videos to be processed (i.e. the time to process a frame does not depend on the video length) and w.r.t. the number of objects observed in the last frame the cost is dominated by the cost of the SVD. While the cost of the SVD grows as $O(n^3)$, the number of objects that can be observed and tracked is not expected to be high and hence it is not a limitation to the performances of the algorithm.

7.4 Experimental evaluation

We have tested the algorithm on three datasets (see Appendix A): (i) APIDIS-basketball (ii) ISSIA-football (iii) In-house outdoor.

The first two datasets depict sports events, thus we expect to observe characteristic behaviours, while people appearance will be ambiguous because of the sports uniforms. The third dataset has complementary features, since people appearance is more distinctive while their behaviours are less peculiar. Moreover the third dataset has moving cameras, non planar environments and cameras with different frame-rates and resolutions. All the three datasets provide annotations on the tracks that, since our focus is not tracking, have been used as input.

7.4.1 Experimental setup

The datasets have been divided in two parts: a *validation set* used to tune the parameters and choose between different possible configurations of different algorithms and a *test set* to evaluate the final configuration of the proposed method once the parameters have been set. The former set is composed by one couple of views from the ISSIA-football dataset (sequence 3-4) and two couples from the APIDIS-basketball dataset, chosen to sample a simple scenario (sequence 1-7) and a complex scenario (scene 1-2) (see Figure 7.6). The latter includes all the other data.

To have a single index of performance, we consider the F_x measure:

$$F_x = (1 + x^2)\frac{PR}{x^2P + R} \tag{7.10}$$

where P is the precision and R the recall. Since certain matches are preferable to uncertain matches, we set the parameter x to 0.5.

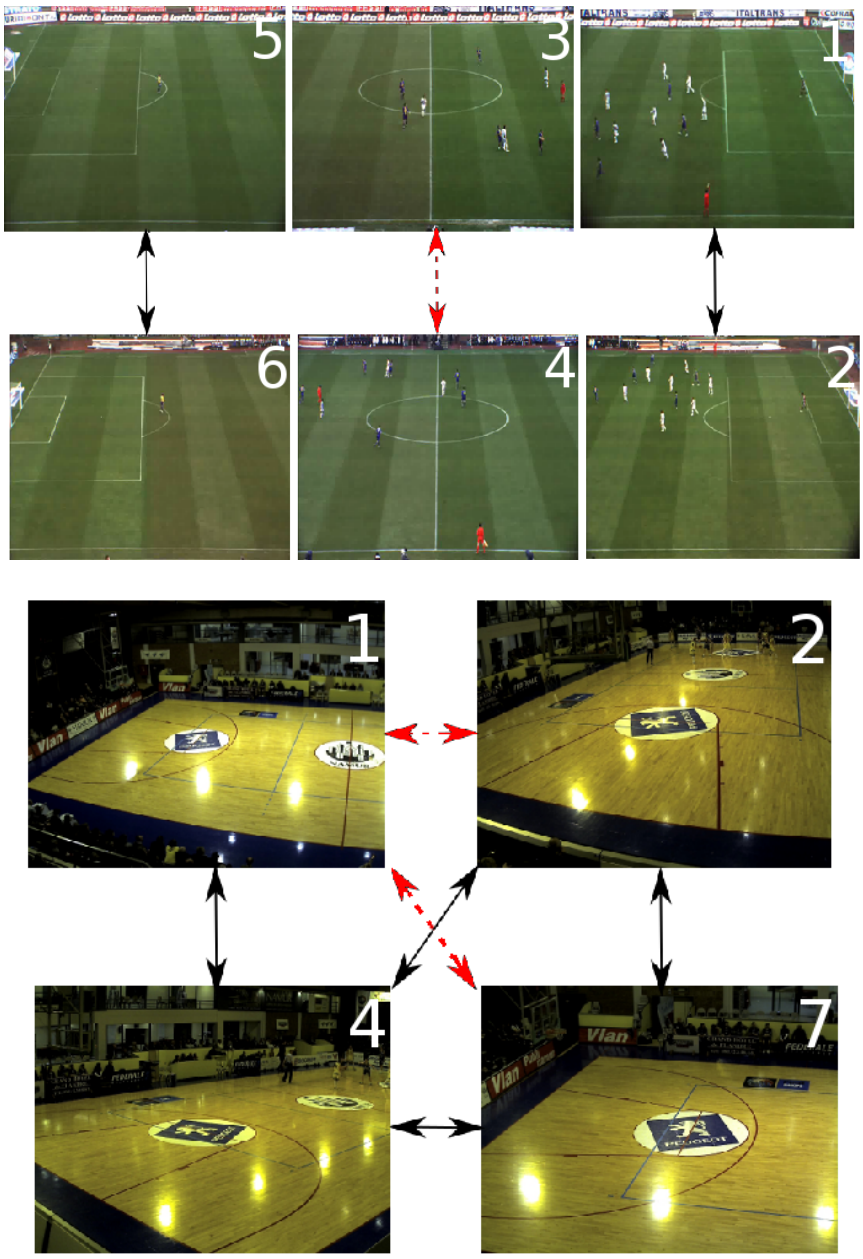


Figure 7.6: POVs of the datasets used in the experiments. The pairs of the test set are shown with continuous black arrows, while the red dotted arrows correspond to the pairs of the validation set.

Description	$F_{0.5}$	Precision	Recall
Optical Flow	0.64	0.88	0.37
HOG	0.96	1	0.85

Table 7.2: Comparison of the accuracy of two visual descriptions.

7.4.2 Method assessment and parameters choice

In this section we provide an extensive evaluation of the performances of each step of the algorithm as independently as possible on all the other steps and we justify from an experimental point of view all the choices done during the development of the algorithm. If it is not specified differently all the other parameters have been selected to maximize the $F_{0.5}$ measure on the validation set.

Objects visual description

Similarly to Junejo et al. [JDLP11] for an action recognition problem, we have compared the chosen HOG description to the optical flow computed with the LK algorithm between two adjacent bounding boxes. As expected, the results show that the optical flow is weaker when the hypothesis of constant illumination between adjacent frames does not hold, this occurring in the case of complex illumination and complex background. Tab. 7.2 shows the mean values of the algorithm using A to compute the solution. Note that the accuracy gap is mainly due to the drop of performances on the APIDIS-basketball dataset, since the difference of the $F_{0.5}$ measure on the simpler ISSIA-football dataset is less than 0.1.

Action description comparison

To justify the choice of the GFD from an experimental point of view, we compare it with two alternatives: (i) Polar-HOG (PHOG) (ii) Raw SSM intensity values. The first descriptor is composed by a stacked set of normalized histogram of the orientation of the gradients computed on a polar grid disposed around the analysed point. It has been included in the comparison since it has been successfully applied to analyse SSMs for video alignment and action recognition (e.g. [DPL09, JDLP11]). The second descriptor is a baseline representation obtained by unfolding raw intensity values of the part of the SSM included in the support of the descriptor.

We have used the validation set to select both the description and the radius of the support, obtaining the results in Figure 7.7. The results suggest the GFD is the feature that fulfils better our requirements: *(i)* it has the highest precision of the set *(ii)* its results are stable w.r.t. variation of the radius of the support *(iii)* the $F_{0.5}$ measure on all the range tested is better than results obtained with the other two description using their optimal values. In Table 7.3 we report detailed

Dataset	Descriptor	TP	FP	mean $F_{0.5}$
APIDIS-basketball	Raw SSM	0.43	0.05	0.75
	PHOG	0.67	0.05	0.87
	GFD	**0.81**	**0**	**0.95**
ISSIA-football	Raw SSM	0.46	**0**	0.78
	PHOG	0.71	0.74	0.90
	GFD	**0.95**	**0**	**0.98**

Table 7.3: Results computed on the validation set using the adjacency matrix A with all the descriptions described in the text. The validation procedure selects as optimal the GFD. TP: True Positive, FN: False Negatives

Dataset	Descriptor	TP	FP	mean $F_{0.5}$
APIDIS-basketball	Raw SSM	0.76	0.18	0.8
	PHOG	**0.98**	**0**	**0.99**
	GFD	**0.98**	**0**	**0.99**
ISSIA-football	Raw SSM	0.94	**0**	0.98
	PHOG	0.94	0.03	0.96
	GFD	**1**	**0**	**1**

Table 7.4: Results of the algorithm with the orthogonalization of the adjacency matrix A^I on the validation set. TP: True Positive, FN: False Negatives

results on the validation set showing that the GFD achieves the best results both in precision and recall.

Matches computation

In this section we compare the raw adjacency matrix A with its orthogonalized version A^I. As a first experiment we compare the performances of the three action descriptors, for a variable time window size both using the criterion of Eq. 7.8 on the matrix A and on the orthogonalized matrix A^I. As it has been described in Sec. 7.3.3, the value of σ used to compute the matrix A^I is chosen automatically looking to the matrix A. Figure 7.7 shows the comparison, highlighting that the accuracy with the orthogonalized matrix is consistently better with all the combinations of descriptor and size of the time window T.

In Table 7.4 we show the results computed on the validation set using A^I. It is interesting to notice that the orthogonalization successfully recovers some of the performance gaps (observed in Tab. 7.3) between the GFD and of the PHOG. Using the orthogonalization the performances on the simple ISSIA-football sequence using the PHOG or the GFD are very similar. A higher performance gap can be observed on the more complex APIDIS-basketball sequence between

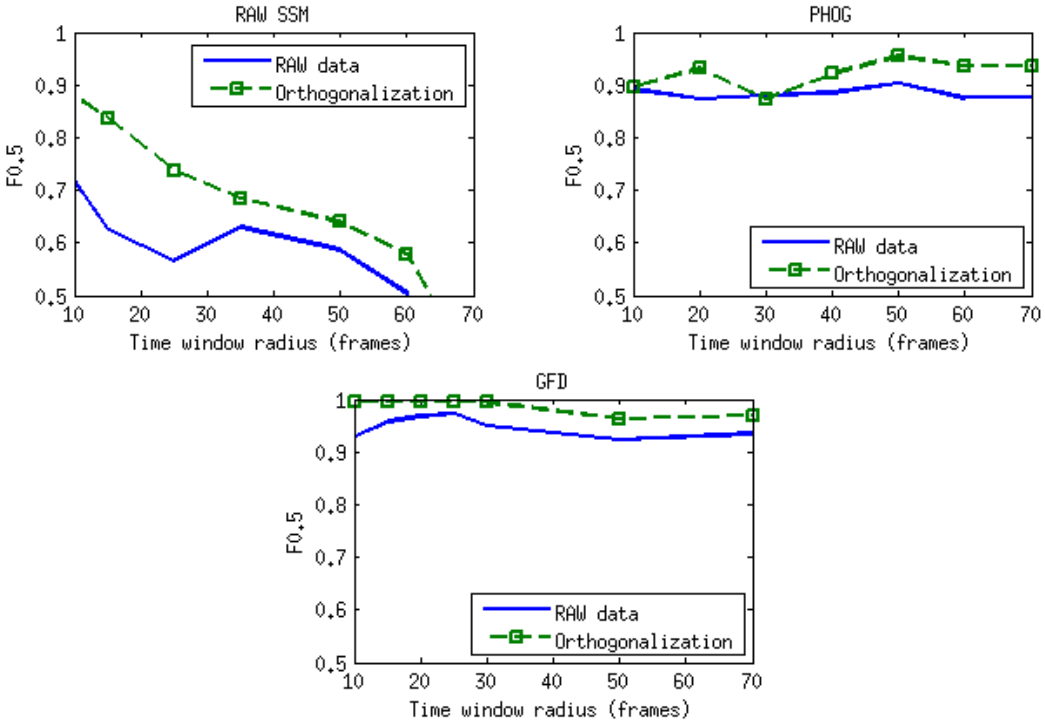


Figure 7.7: Plot of the $F_{0.5}$ measure on the validation set for different choices of size of the time window for all the descriptions tested. The GFD obtains the best results both in accuracy and stability.

Descriptor	A	Lap A^I	Lor A^I	Gauss A^I	No-Weights A^I
Raw SSM	0.71	0.84	0.86	**0.88**	0.81
PHOG	0.9	0.91	0.9	**0.95**	0.83
GFD	0.97	0.99	0.99	**0.99**	0.98

Table 7.5: $F_{0.5}$ measure computed on the validation set with different weighting functions of the similarity matrix A (Lap: Laplacian, Lor: Lorentzian, Gauss: Gaussian). The best results are obtained by weighting with a Gaussian the matrix A before its orthogonalization.

the latter two descriptors and the raw SSM values.

The results shown in this section confirm the effectiveness of the orthogonalization to raise the recall rate. The tests on the orthogonalized adjacency matrix also confirm the good performances of the GFD w.r.t. the other descriptors tested and underline the excellent stability w.r.t. variations of its parameters, that lead to a low risk of overfitting.

Non-linear weighting of the adjacency matrix

In this section we show the results of the tests to compare the Gaussian, Lorentzian, Laplacian weighting of the matrix A and the method to choose automatically σ. In table 7.5 we show the results computed on the validation set with all the configurations taken into account. All the parameters but σ are chosen to maximize the $F_{0.5}$ measure.

These tests show clearly that the Gaussian weighting is an important step to be done before the orthogonalization of the matrix A^w, since not only it improves the results w.r.t. the original matrix A, but the computation of the orthogonalization on the raw A can degrade the accuracy. Probably this is due to the fact that the weighted matrix, being "sparser" than the original one, it is closer to the matrix M that we want to approximate, making the closest orthonormal matrix (w.r.t. the Frobenius norm) a more reasonable approximation of the correct solution. As opposite to [CH03, DIOV06], while the application of the Lorentzian and of the Laplacian weighting function improves the performances w.r.t. the non weighted data, in our case the Gaussian weighting gives the best results.

7.4.3 Overall performances

Tab. 7.6 shows the results of the full algorithm on the test set using the settings that have been chosen on the validation set ($K = 45$, $R = 15$, $w_o = 0.8$, $T_s = 25$frames/1 second). We have not made any choice regarding the use of the orthogonalization (Eq. 7.7): we leave to the user the choice between a higher precision and a lower recall or the opposite. The expected overlap with the FOV should be considered as well, since as it is explained in Sec. 7.8 the orthogonalization

Dataset	Precision	Recall	mean $F_{0.5}$
APIDIS-basketball	0.97(1.00)	0.95 (0.81)	0.97 (0.95)
ISSIA-football	1.00 (1.00)	1.00 (0.94)	1.00 (0.99)
In-house outdoor	0.87(0.92)	0.85 (0.61)	0.87 (0.83)

Table 7.6: Results of our algorithm ($F_{0.5}$) applied on the three test sets. We show the results of the algorithm in two configurations, in brackets the values obtained using the matrix A, the others values have been obtained with the orthogonalized matrix A^I.

Dataset	Precision	Recall	mean $F_{0.5}$
APIDIS-basketball	1	0.95	0.99
ISSIA-football	1	1	1
In-house outdoor	0.92	0.85	0.9

Table 7.7: Results ($F_{0.5}$) obtained on the test set by considering the matrix A^I and using the optimal threshold w.r.t. the $F_{0.5}$ measure on the confidence value to select the accepted matches. The gap of performance w.r.t. the results shown in Table 7.6 shows that the confidence values are able to order effectively the correct matches w.r.t. the wrong ones.

of A could lead to poor results with a small overlap (see Sec. 7.4.3). Tab. 7.6 shows the results on the test sets of both the configurations.

The SNR confidence measure proposed in Sec. 7.3.3 to sort the matches according to their probability of being correct has proved to be effective: in Tab. 7.7 are shown the best results that can be obtained starting from the orthogonalized solution of Tab. 7.6 and filtering the results by thresholding the accepted matches with the optimal threshold w.r.t. the $F_{0.5}$ measure. It is interesting to note that we can consistently achieve the same recall of the orthogonalized algorithm with a comparable or better precision of both the configurations, proving that the SNR measure is effective in ordering the selected matched depending on the confidence on the solution.

In Figure 7.8 we show an example of error that is due to three factors: one of the cameras observes only one of the two objects at a time, the other both of them at the same time; a strong occlusion on one camera penalize the correct association; one of the objects is visible in one camera only for three seconds.

All the errors on the in-house outdoor dataset show at least two of the problems listed.

To check the behaviour of the algorithm w.r.t. variations in the point of view, we have analysed the correlation between the angle of the optical axis of the considered POVs (approximated to multiples of 15 degrees) and the $F_{0.5}$ measure. The result is a correlation coefficient that is between -0.1 and 0.2 depending on the configurations used, suggesting a weak dependence on the relative viewpoints.

Figure 7.8: An error on the in-house outdoor dataset. Camera B observes only one person and both person are walking. Those two factors, combined with the occlusion that is seen only by one camera, lead the algorithm to an error.

Effectiveness in the presence of camera motion. An important feature of the proposed algorithm is its ability cope with moving cameras.

We test our method in the case of moving cameras on the in-house outdoor dataset, by combining all the cameras with the moving one (camera 2) the mean $F_{0.5}$ is 0.73 using the orthogonalization and 0.79 without. While those values are lower than the mean values showed in Table 7.6, this deviation is mainly due to one single sequence: the coupling of camera 2 with camera 5 shows weak performances of due to the poor overlap with the FOV of the two (as tested in Sec. 7.4.3). Consistently with the other tests regarding the robustness to weak overlaps, the experiments have shown a decrease in the performance of the A^I w.r.t. A and a general, but more limited, drop of the $F_{0.5}$ index in the latter configuration. Note that while the overlap of the FOVs is small for most of the video, when it is most overlapped the common FOV it is in the far field of camera 5, where the resolution and the artefacts of the compression degrade the image quality (flickering and moiré). If we remove camera 5 the mean $F_{0.5}$ index of camera 2 raises to 0.87 (with orthogonalization) and 0.90 using the plain matrices, showing that the algorithm can work with moving cameras without incurring in penalties on the performances.

FOV with a small overlap. We now discuss the effect of observing two FOVs that are only marginally overlapped is that, in the same instant, may observe two different sets of objects. The analysis is divided in two cases: 1) (Figure 7.9) one set is completely contained in the second one, and the latter includes some more objects; 2) (Figure 7.10) there is a subset of common objects, and each camera observes some objects that are not available in the other view.

The reason of those two experiments is that, in the first case, the main assumption of having a solution that is a full rank orthonormal association matrix (as M) is still correct, while in the second one it is not more valid, since the rank is equal to the number of the common objects. All the experiments have been repeated sampling differently the actors 1000 times, in the case of the first experiments we have repeated them swapping the roles of the videos.

In Figure 7.9 we show the results in terms of $F_{0.5}$ varying the relative amount of common objects in the two views. As expected with sets that are barely intersected the solution computed on the matrix A outperforms the solution computed with its orthogonalization. While the recall is consistently higher with the matrix A^I, its precision is lower in all the cases, being equal only when both the algorithms reach the maximum value. Since the $F_{0.5}$ measure decreases less than 0.05 even with an intersection between the sets of 10%, the performances in this case are satisfactory.

We show the results obtained with the second set of tests in Figure 7.10. As expected this noise has a stronger effect on the algorithm, since it makes false the implicit assumption of a one-to-one correspondence. The experiments show that the orthogonalized solution is weak w.r.t. this kind of noise, while the approach based directly on the matrix A is more robust. This confirms that, as it has been pointed out in Carcassoni et al. [CH03] in a different context, this class of methods is not a good choice to manage strongly different field of view. At the same time, since

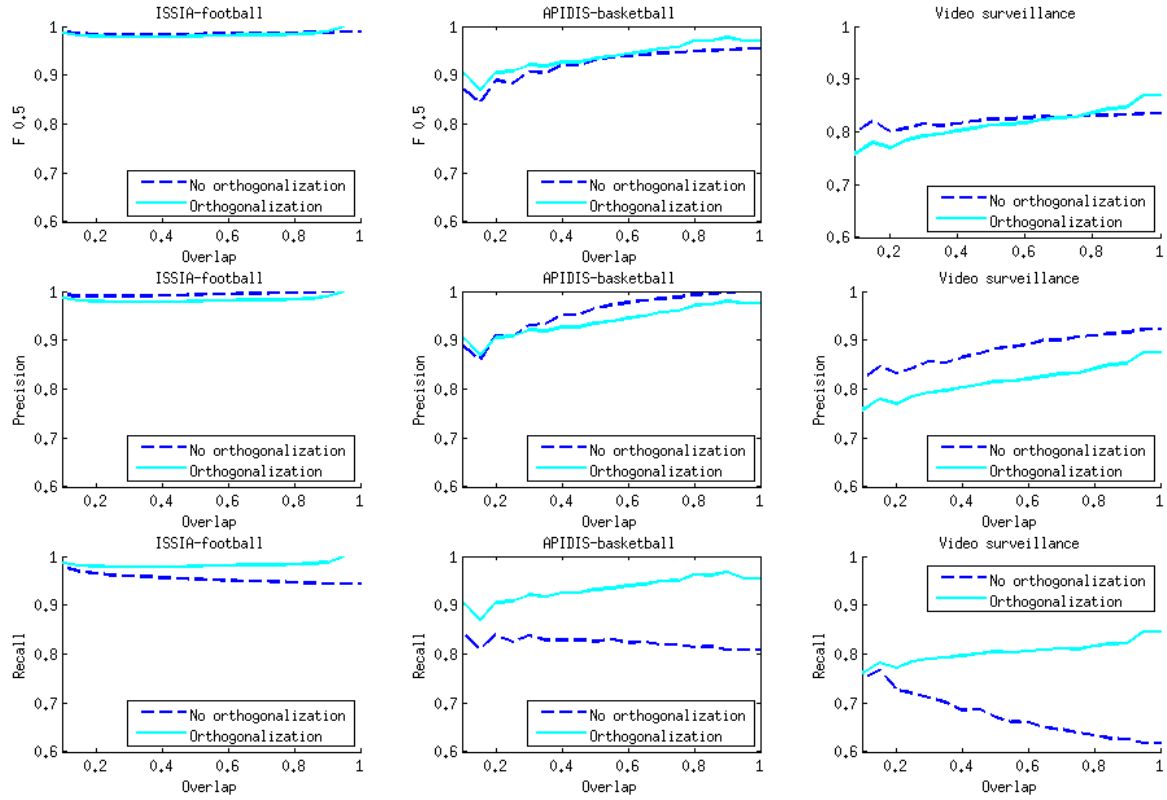


Figure 7.9: $F_{0.5}$, precision and recall of the algorithm with sets of objects of different cardinality extracted from the test set. The smaller set is totally included in the bigger one, the proportion of the intersection is reported on the x-axis. It is apparent how the orthogonalization forces more matches leading to a higher recall, while loosing some precision.

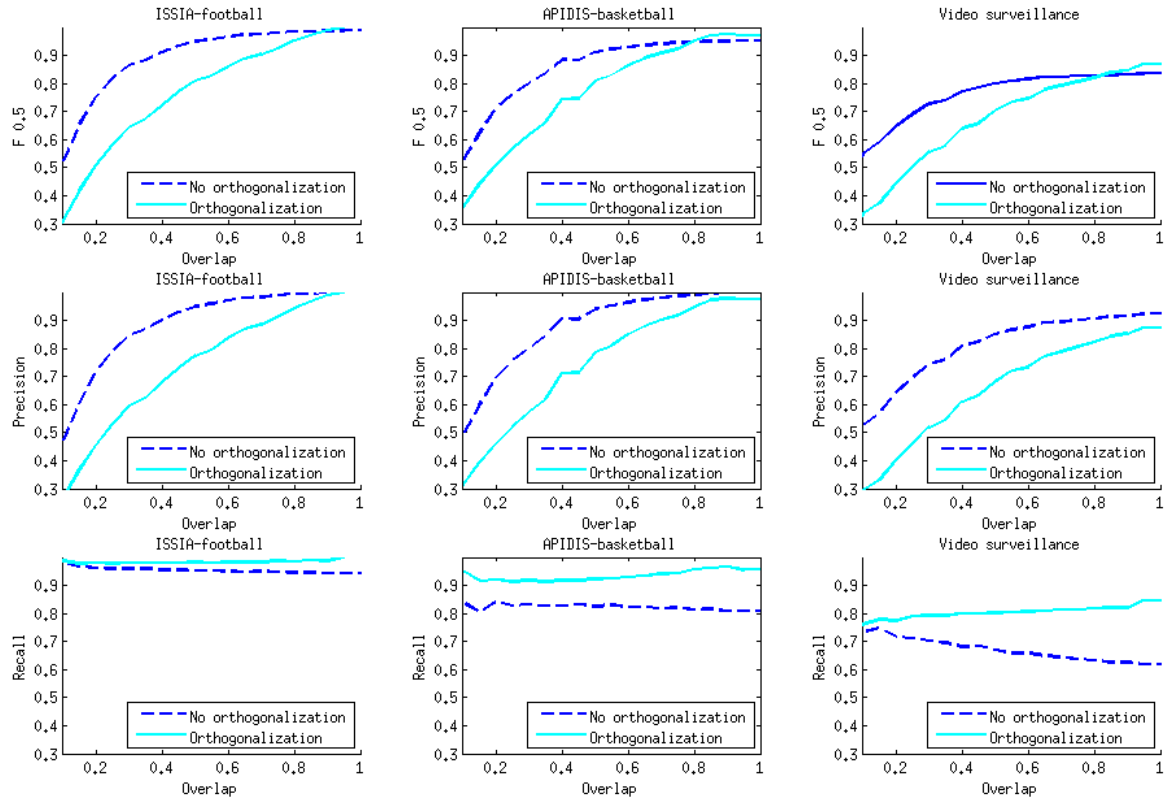


Figure 7.10: $F_{0.5}$, precision and recall of the algorithm with sets of objects of different cardinality extracted from the test set such that each couple of videos containing only a subset of common objects. The two sets considered have a decreasing intersection reported on the x-axis, the remaining part in both the sets is obtained randomly sampling two non overlapping sets of objects. In this case the orthogonalization of the similarity matrix does not produce any benefit.

with an overlap of the 40% of objects both the precision and the recall on the ISSIA-football and on the APIDIS-basketball datasets are above 0.9 if computed on the matrix A, we have the proof that the input representation is strong enough to have reasonably good performances even in very difficult scenarios (60% of the objects are never seen on both the cameras).

Effect of short tracks. To analyse how the algorithm behaves with short observations we have prepared virtual videos containing tracks extracted by the original videos cropped to have all the same length. We have considered all the tracks longer than 400 frames and we have recombined them into a set of virtual videos containing 10 random objects whose behaviour is extracted from a random subsequence of the associated original track. The coherence with the other experiments is maintained since each track is combined in each virtual video only with tracks from the same video.

As final result each experiment compares two videos containing a set of ten people that are acting

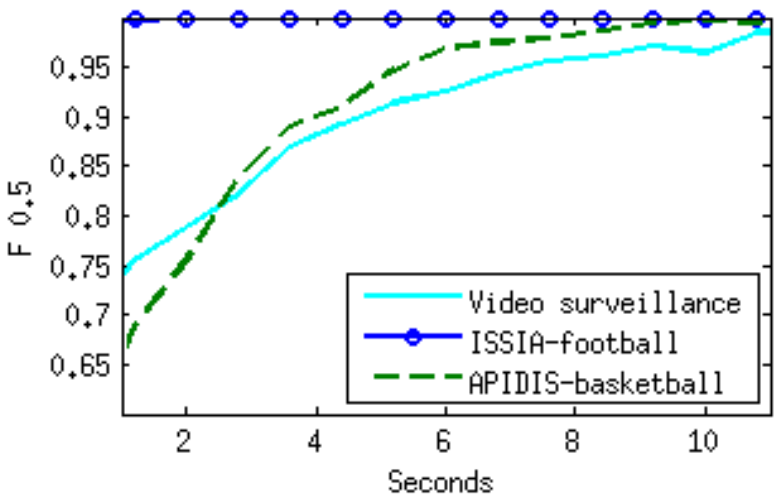


Figure 7.11: Performances of the algorithm w.r.t. the temporal extent of the observations on the three datasets of the test set.

exactly as they were doing in different instants of the original videos but in different instants.

We have considered a time description length between 10 and 400 frames, at steps of 10 frames, repeating the experiments 100 times with different people observed in different time intervals.

In Figure 7.11 we show how the required observation length strictly depends on the complexity of the video. In the simplest cases with a few seconds it is already possible to reach $F_{0.5} > 0.99$, while in the most difficult scenario the accuracy start to be reasonable only after a few seconds. This is not only due to the need of collecting enough discriminative actions, but also to the occlusions that are common in the sequences tested and whose negative effects may dominate the comparisons in case of short observations.

Since by considering only relatively long observations the in-house outdoor dataset reaches almost the same performances of the other two, we can suppose that the weaker performances on our dataset do not depend only on the less distinctive observed behaviours, but mainly to the short length of tracks.

In Figure 7.12 we show the effect of applying the SNR measure to sort the matches. The aim of this experiment is to show how, also in cases where the matching algorithm is not able to obtain good performance, the ordering power of the SNR is not affected. The tests are carried out on the three datasets and show the performances for different observation lengths (left column) and for a different overlap in the field of view (right column). The gain obtained with a selection of matches with higher SNR is apparent.

Non homogeneous frame rates. One of the requirements of the algorithm is to be able to work with videos with different frame rates. In addition to the tests on the in-house outdoor dataset,

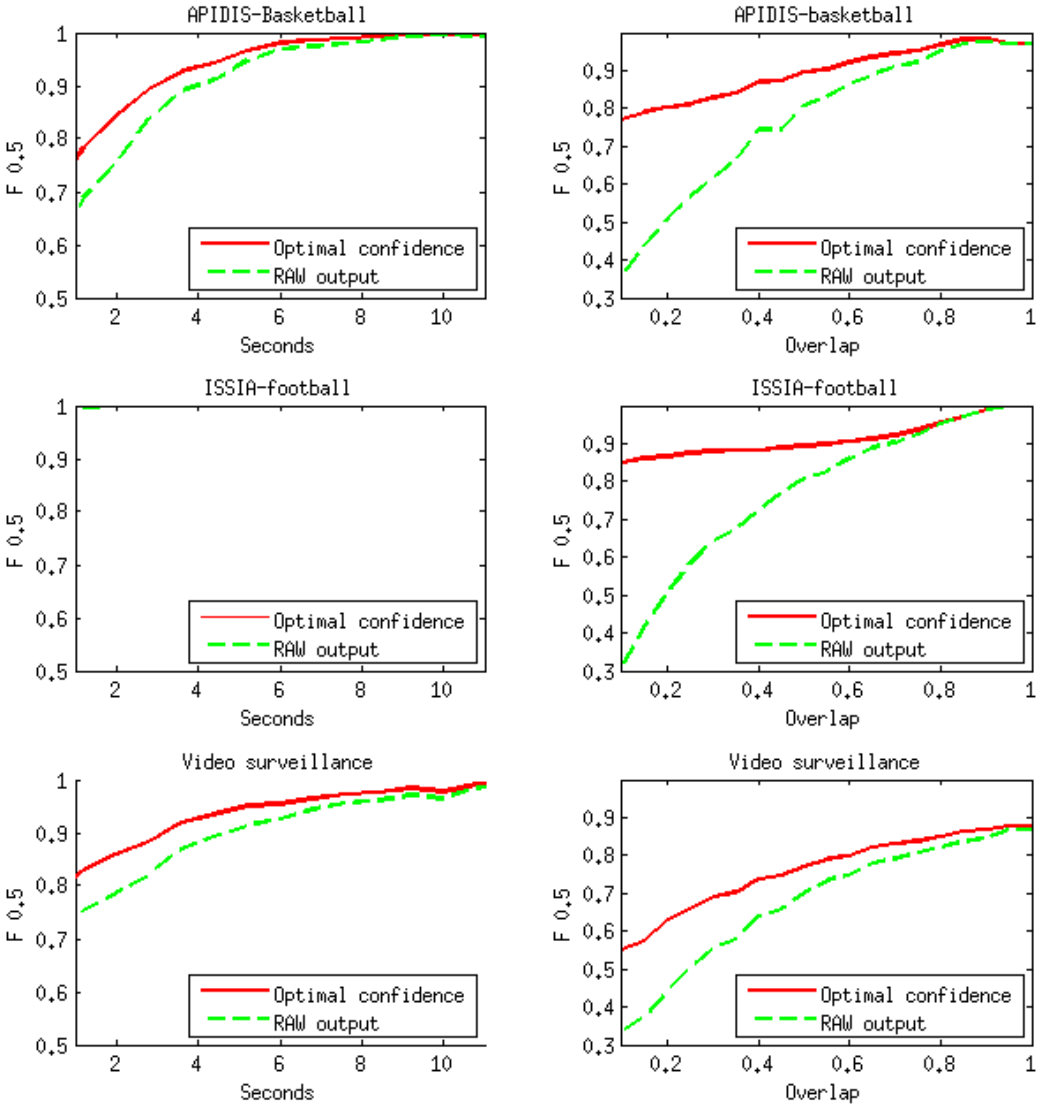


Figure 7.12: Effectiveness of the SNR-based sorting. Left column: $F_{0.5}$ measure for different observation lengths. Right column: $F_{0.5}$ measure for different FOV overlaps. The results obtained by using only matches with a high SNR are consistently better.

FPS	ISSIA-football	APIDIS-basketball	In-house outdoor
Orig. (50-24)	1	0.97	0.87
13	1	0.94	0.82
6	0.95	0.8	0.59

Table 7.8: Performances of the algorithm by re-sampling the original videos to obtain videos with different frame rates.

that has been acquired with different frame rates, we have repeated the tests on the remaining two datasets by manually re-sampling the frame rate of the videos to check if the results depend on the frame rates rather than the content of videos.

We have repeated the experiments with a frame rate ranging between 50 and 6. The results (see Table 7.8) show that the algorithm can work effectively in the range between 50 and 13 fps with a variation of $F_{0.5}$ bounded by 0.05. A further decrease in the frame rate affects significantly the performance of the algorithm, and with only 6 fps the $F_{0.5}$ decrease in the worst case of more than 0.25.

While a good sampling of the actions in the videos is required, the good results in the range (50fps-13fps) prove a good robustness w.r.t. different sampling frequencies of the actions. Moreover better accuracy may be obtained selecting the parameters of the algorithm specifically for low frame rates.

Robustness to track noise. To study the robustness w.r.t. noise in the input tracking data we have added a Gaussian noise on the position of the bounding boxes of each object. The noise is independent for any object in any frame and it is added in both the axis using a unimodal Gaussian distribution of mean 0 as model. For each bounding box we have extracted two values from the Gaussian and we have added them to the position multiplied for the size of the bounding box. In Figure 7.13 we show two examples of the noise applied with different values of σ.

In Figure 7.14 it is shown the mean over 50 experiments of the $F_{0.5}$ measure with different values of noise. The results show a good robustness up to $\sigma = 0.1$, and drop faster after that point. Normalizing the meaning of the noise to a bounding box of height 128 pixels that we use to compute the HOG representation, a noise with $\sigma = 0.1$ means that the target is randomly displaced between two consecutive frames of more of the 10% of its size with probability 0.2. Moreover the input tracks are not noise free, adding further noise.

7.5 Comparison with other approaches

In this section we compare the proposed approach based on the analysis of the dynamics of the appearance of people with two different alternatives: (i) a colour description (ii) a geometry

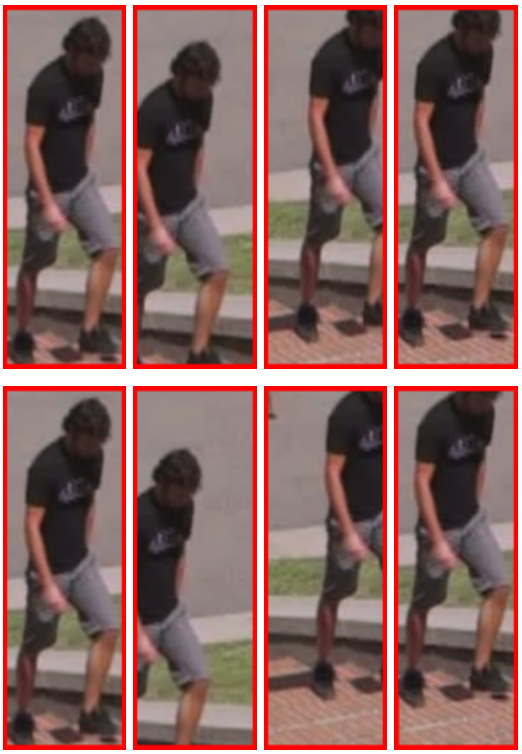


Figure 7.13: Bounding boxes affected by noise. The first line shows a sequence extracted from a Gaussian with $\sigma = 0.07$, the second one with $\sigma = 0.2$. Given the standard deviation, there is a 20% of probability of extracting a displacement bigger than the maximum shown.

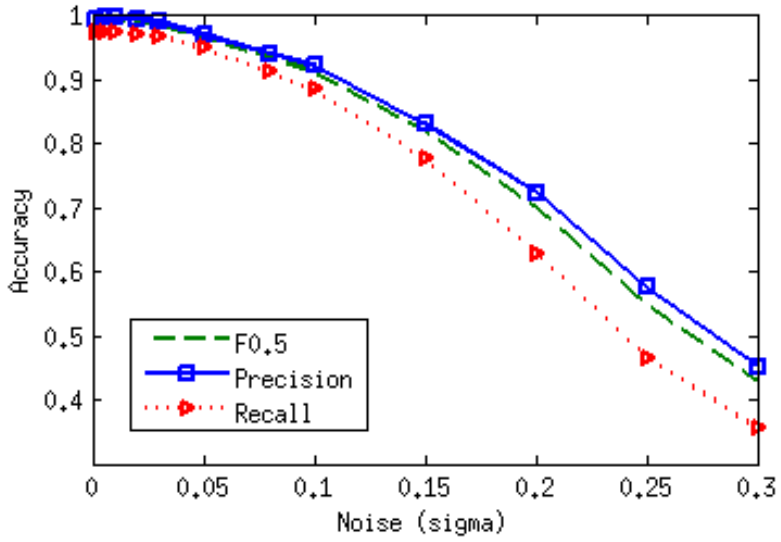


Figure 7.14: Performance of the algorithm adding noise to the positions of the objects on each frame. The noise is reported as the σ of the Gaussian from which it is extracted the percentage of the amount of displacement in both the directions w.r.t. the size of the bounding box.

based description.

7.5.1 Colour based approach

The colour description that we have selected is the one proposed in Bird et al. [BMPI05] for re-identification. In our implementation, to remove the background we have used a change detection algorithm based on a running average, whose parameters have been chosen by visually inspecting the foreground masks.

While Bird et al. [BMPI05] may not be aligned with the updates in the state of the re-identification (a more recent approach is for example [BVC11]), this descriptor encodes more information other than the colour of the clothes as for example the pose and position of the arms. The pose can be considered noise for a re-identification algorithms as their objective is to match the same person in different instants, but it is a source of valuable information for us as we compare people in the same instant.

Moreover, since Bird et al. [BMPI05] aim to solve a different task, we did not follow the same procedure of training a classifier for each observed person (as we do not expect to see the same person twice in the same camera). We compare two people using a frame by frame comparison (Eq. 7.4), using the validation set to select w (the best accuracy on the validation set is achieved with $w = 1$).

7.5.2 Geometry based approach

Another possible solution is to consider the geometry of the system to associate the people whose tracks are geometrically more compatible.

We have considered to map each bounding box between the images both using homographies and fundamental matrices. The algorithm computes, for each possible assignment, the distance between the feet on one camera (the center of the lower side of the bounding box) and the corresponding feet on the other camera. In the case of the fundamental matrix the distance is computed between each head and the associated epipolar line. A couple of tracks match iff the criterion 7.8 applied on the matrix containing the mean distances between the tracks holds.

Since in the experiments we consider non calibrated systems, we have estimated the geometry using the sequence of positions of feet and head of a training track. A more complex and possibly precise solution is given in Calderara et al. [CCP08], that however, requires more data or a collaborative training. Moreover Calderara et al. [CCP08] propose a more complex method to deal with groups of people, that are out of the scope of this study.

7.5.3 Results

Table 7.9 reports the results of the comparison on the test sets. In the case of the colour based approach we show the results with and without the orthogonalization of A, for the geometry based method the mean and the median value obtained considering different tracks as training.

As it can be expected the results of the colour based method are low on the sport sequences, however, on the APIDIS dataset it seems to benefit of the changes of poses and using the orthogonalization it achieves good results. The performance drop between the static and the moving set can be attributed to the absence of foreground segmentation on the latter set, that makes the representation prone to noise from different backgrounds.

Interestingly the colour based method does not suffer of the same performance drop of our method on the couple 2-5 of the in-house outdoor dataset, as, by removing it, the F measure is 0.78 (instead our method achieve 0.87). We suppose that this can be due to the fact that the simpler colour descriptor still work in the far field of camera 5, where the HOG and SSM descriptions are affected by the the poor resolution and flickering of images.

In general the dynamic-based approach shows to be more precise both with and without the orthogonalization in all the settings, even if, as it has been observed in other context (see Sec. 7.4.2), the orthogonalization reduces the gap between the two.

The geometry based approach obtains perfect results in most of the tests on APIDIS and ISSIA dataset (the median $F_{0.5}$ is 1), however despite their length not all the tracks are sufficient to obtain a good calibration (hence the mean result is lower). As it can be expected in the in-

Method	APIDIS	ISSIA	In-house outdoor static	In-house outdoor moving
Proposed	0.97(0.95)	1(0.99)	0.95(0.86)	0.73(0.79)
Colour	0.9(0.62)	0.54(0.5)	0.89 (0.75)	0.79 (0.59)
Homography	0.95-1	0.91-1	0.55-0.54	-
Fundamental matrix	0.95-1	0.82-0.99	0.63-0.74	-

Table 7.9: Comparison of the proposed behaviour-based algorithm with a colour-based and a geometry-based method. For behaviour and colour-based method are reported the $F_{0.5}$ with orthogonalization and without it (it brackets). For the geometry-based method are reported the mean and the median value of each set of experiments varying the training set.

house outdoor dataset the results based on the homography are not reliable since the world is not planar, however, also the calibration based on the computation of the fundamental matrix does not achieve good results.

The latter result is probably due to the input sequence that is too short and not sufficient to compute a reliable calibration and better results may be achieved using a user provided fundamental matrix. However this result suggests that with a geometry based method it is needed a collaborative training or some user interaction to achieve good results on static cameras and clearly it is even more challenging to work with moving cameras.

7.6 Discussion

In this chapter we have presented a method based on the dynamic of the appearance of people (behaviour) to match people between multiple cameras able to work in a constraint free setting.

The method is based on the same idea of exploiting the SSM to compute a view invariant representation of the action of a person [JDLP11] already used in the Chapter 6 for video alignment. Such approach based on a high level representation, allows to have an algorithm that is independent of the geometry and does not require any training.

The advantage w.r.t. geometry-based methods is the lack of any requirements on the calibration of the system, allowing the cameras to move freely. Moreover our method does not consider the static appearance and can associate people dressed similarly or starting from gray level images.

In Sec. 7.4 we have proposed a deep experimental analysis to test the accuracy and the limitations of the algorithm, that have proved to be precise in discriminating people behaving similarly (e.g. Figure 7.2), to the price of requiring a few seconds of video to compute a reliable solution.

A journal paper describing the proposed approach is in preparation [ZOCed].

Chapter 8

Conclusions

The goal of this thesis was to study computer vision algorithms for video-surveillance able to manage single and multi-camera systems and to extract information from them. In the case of a single camera, our work considered motion analysis to detect and track moving objects, people detection and people counting to deal with different levels of crowd in the scene. On multiple views we addressed the problem of coordinating multiple streams from the geometrical and the temporal viewpoint and of associating the same object (person) between multiple views.

The contribution of this thesis to the literature are several.

The first one [ZO11] is the study and the validation of a framework based on structural regularized feature selection to build an image description for objects detection. Taking into account the structure of the full detection pipeline and the structure of the features, the method enriches the description with the dual objective of containing the computational cost while increasing the accuracy. The method has proved to be effective on the variable-size HOG description, while the performances on the covariance matrices description suggest that it may be best suited for a cascade of classifiers architecture. Finally we designed the combination of the framework with multi-scale descriptors (we describe a multiscale variation of the variable-size HOG), that may fulfil the objective of increasing the performances of the algorithm without affecting the computational cost.

The second contribution [ZON13] is a simple method for real-time people counting that, despite its low computational cost, has proved to achieve results comparable with the state of the art algorithms in different scenarios. To develop the method, starting from a basic geometrical approach, we have systematically analysed the source of errors, modifying the algorithm to take them into account without giving away the generality of a model free approach. Our method has been successfully tested during a live demo for the project SINTESIS [1] and has been used in real-time on

[1] SINTESIS Sistema INTEgrato per la Sicurezza ad Intelligenza diStribuita (integrated system for distributed security) SIIT. Funding: MIUR - L297

a video surveillance camera installed in our department. Even if in this case we constrained the method to a static calibrated camera and to a planar world, it can be configured simply since it does not have a complex training procedure that would requires to collect big amount of data, it has a reduced computational and it has proved to be robust w.r.t. weak calibrations.

The third contribution [ZCO13] is a method for video alignment whose key idea is to consider high level information on the appearance of the dynamic of multiple objects to obtain a constraint free algorithm. Given a set of tracks and object association information, the proposed method can work in complex real-world scenarios without assumptions or requirements on the geometry, on the similarity of the points of view and on the structure of the warping functions. The price is that we shift some of the complexity of the problem from the alignment algorithm to the detection, tracking and association of people. However most of the algorithms in the state of the art rely on the robustness of a tracking algorithm and hence are comparable under this aspect.

The fourth contribution [ZOCed] is a matching algorithm that adopts a high level SSM-based description of the behaviour of people (similar to the one used for video alignment) to associate identities between pairs of cameras. The contribution is twofold in fact, both the proposed algorithm and the general idea of the approach are novel as, to our knowledge, there is no method in the literature that exploits behaviour analysis to match people between views. The method that we propose has proved to be robust and accurate in a broad range of scenarios being able to distinguish between people with very similar behaviours with high accuracy. Thanks to the high level approach it does not bundle any assumption on the scene or on the camera pose and motion. As opposite to other approaches based on the static appearance, it is able to distinguish between people with similar appearance and it supports grey level images (as it could be in the case of night videos). Moreover, since it does not require any geometrical information, it supports moving cameras whose geometry may be difficult to compute on a frame basis in a general scenario.

Our study focused only on some of the challenges when dealing with multi-camera surveillance systems. Each contribution aimed to cope with the complexity of the whole problem by solving a specific task. Each method can be analysed singularly or as a building block of a more complex system that, given in input multiple streams computes high level information on the observed scene.

Wide range future works include the study of high level information to model the dynamic of activities in the scene. A straightforward extension of our work may involve the use of the SSM-based descriptions that we adopted to associate identities between views, for analysing behaviours. We verified in our work that SSMs perform very accurately when used to describe a specific instance of a behaviour and in fact their representative power allowed us to associate very accurately people in different views. It will be interesting to investigate instead their capability in describing from a more global standpoint classes of behaviours. This will also require the design of an appropriate similarity measure or kernel function to compare descriptors, as well as the study of (possibly unsupervised) learning methods to model classes of coherent events.

If the occupancy of the scene is relevant, the high-level analysis can take place after the people counting we developed in this thesis. Once that a group has been observed, with an estimate of the people belonging to it we can apply the people detector and understand how people are located each other. In other words, we will aim at learning people interactions to model groups behaviours, that can be better described if the context (in this case nearby people) is considered (e.g. [CSS11]). Part of the current work is about the development of methods to describe and learn collective activity. As a first step, we are studying methods to classify people poses (making use of HOG descriptors), a problem only partially considered and that provides challenges due to the smooth variations among groups of coherent poses. It represents a mid-level information then used to build a graph-based description of the group. Each node in the graph (a person) is labelled with an appropriate representation, while edges denote relations among people. We are currently investigating the use of such graphs, as well as the design of appropriate similarity functions (see [LH05, NAMF12]).

It is worth noting that the use of multi-camera systems will be crucial to acquire more data and to have more robust and less ambiguous estimates. We plan to extend the techniques to coordinate multi-camera systems in unconstrained environments (e.g. the alignment and the people association problem in this thesis) to the challenging case of groups of people, where the availability of observations from multiple view can make the difference.

Appendix A

Benchmarks

A.1 Images datasets

INRIA pedestrian

The INRIA pedestrian dataset (*http://pascal.inrialpes.fr/data/human/*) is composed by 1774 images of high resolution pedestrians (taller than 100 pixels) and a set of 1671 images with no people.

The dataset is divided in a training set counting 1208 person (mirrored to 2416) and 1218 negative images, while the test set contains 566 positive images (mirrored) and 453 negatives. The size of the positive samples is 128x64 plus a variable border useful to avoid border effects on the description extraction process. However, whereas it is not available the images are padded copying the last row/column of pixels (see Figure A.1)

Caltech pedestrian

Caltech pedestrian dataset *http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/* is a carefully annotated dataset that contains, other than the bounding boxes on the image, also annotation on the occlusion state.

The dataset has been acquired from a camera with resolution 640x480 pixels mounted on a moving vehicle and reports more than 350000 annotations. However, since the dataset comes from videos, each pedestrian appears multiple times and the number of unique people is 2300. The median height of the bounding boxes is 48 pixels.

Figure A.1: Images from the training set of the INRIA pedestrian dataset. Each image in the set appears mirrored and, if there is not enough border, the image is padded.

Figure A.2: Sample frames from the Caltech pedestrian dataset.

Chapter	APIDIS-basketball	ISSIA-football	In-house outdoor	In-house indoor	PETS09	PETS07
Object detection					X	
Crowd estimation				X	X	X
Video synchronization	X	X				
Objects association	X	X	X			

Table A.1: Datasets used for the evaluation of each algorithm proposed in the thesis.

A.2 Videos datasets

APIDIS-basketball

The APIDIS basketball dataset (*http://www.apidis.org/Dataset/*) is composed by 7 videos acquired from different cameras framing a basketball match. The length of each video is one minute (1500 frames) and contains up to 12 people that are annotated with position and size of their bounding boxes.

In this thesis we have considered only four cameras (see Figure A.3) selected to frame an overlapping field of view and removing the two cameras with fish eye lenses. Each camera has different orientation and position, with a maximum angle between the optical axis of roughly 90 degrees.

ISSIA-football

The ISSIA-football dataset (*http://www.issia.cnr.it/htdocs%20nuovo/progetti/bari/soccerdataset.html*) is composed by 6 videos acquired from 6 cameras disposed in couples that frame the same field of view from opposite viewpoint. As it can be observed in Figure A.4 the videos are mirrored horizontally.

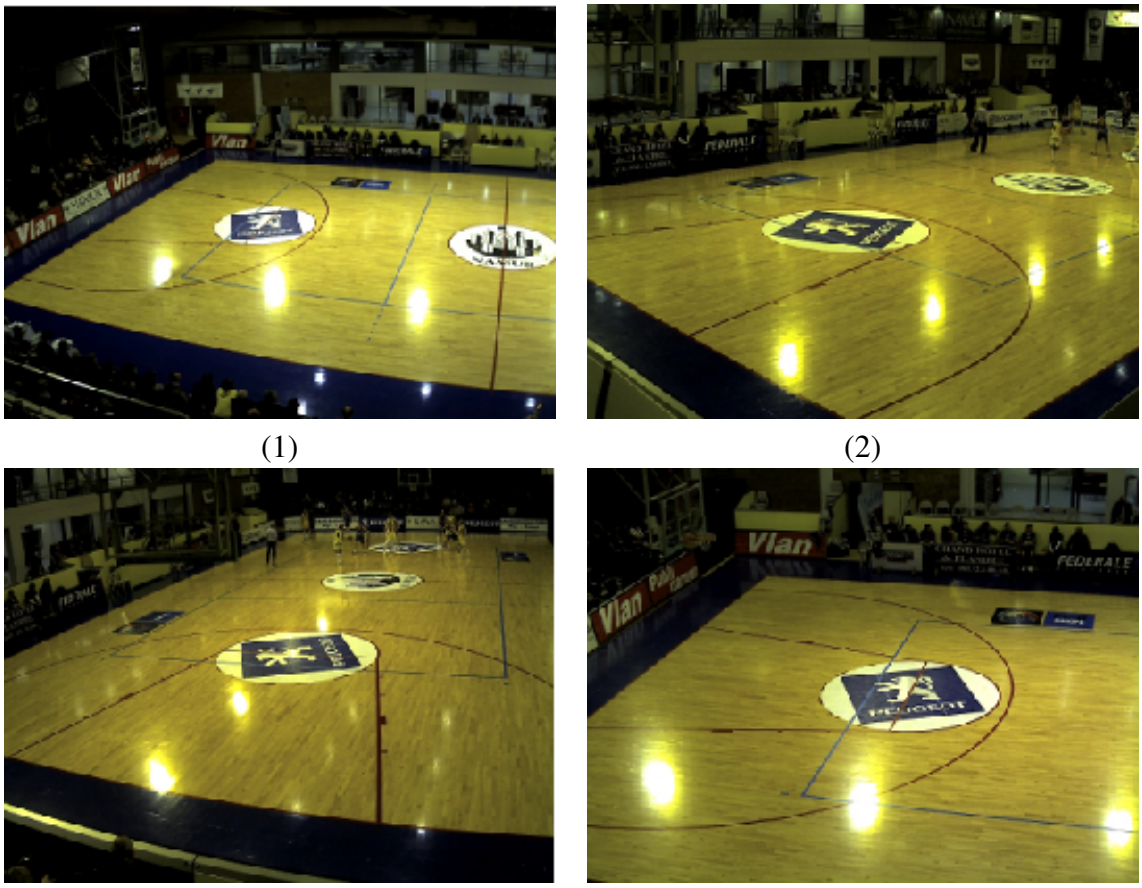
Each video is two minutes long and the associated ground truth contains position and size of each bounding box for all the players appearing from the frame 401.

PETS 2009

This dataset has been released during the Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (*http://www.cvg.rdg.ac.uk/PETS2009/a.html*) and has been used to evaluate people counting and tracking algorithms.

The videos have been acquired at 7 frame per second. The sequences contain both small groups and single pedestrians and groups bigger than 40 people.

In this thesis we have used the view 1 (see Figure A.5) subsets S1.L1 from the original data, that are the parts that the organizers of PETS 2009 have selected as benchmark for the evaluation of



(1) (2)

(4) (7)

Figure A.3: Points of view of the four considered cameras of the APIDIS-basketball dataset. The labels refer to the full dataset.

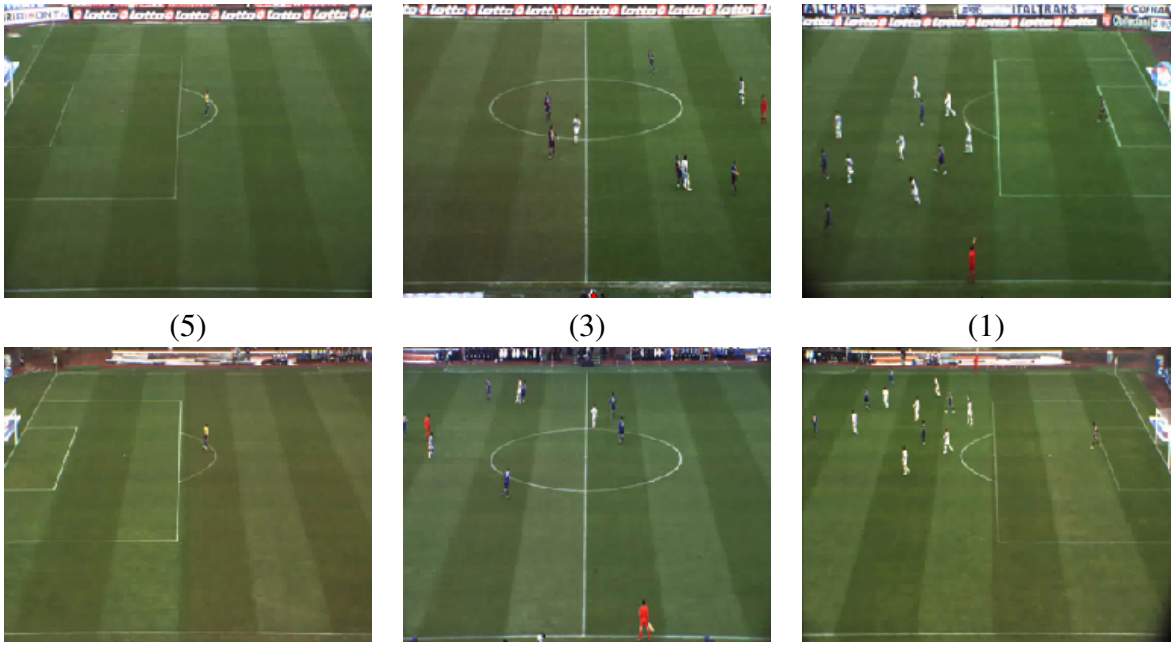


Figure A.4: Points of view of the cameras of the ISSIA-football dataset.

Figure A.5: Example frames from view 1 from the benchmark data of PETS 2009.

Figure A.6: View 3 from the benchmark data of PETS 2007.

people counting algorithms. Moreover we have used some videos from S1.L2/S0 that have been tested in literature.

The people counting annotations are not provided by the organizers and therefore we have manually annotated each frame of the sequences used with the number of people in the scene. The full geometrical calibration is provided by the organizers.

PETS 2007

This dataset have been proposed in the Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (*http://www.cvg.rdg.ac.uk/PETS2007/data.html*) to test algorithms to detect suspicious behaviours (loitering, unattended luggage and untended luggage removal).

We have used the view 3 of the third dataset S00 to count people. The video is three minutes long (25 frame per seconds) and contains up to 10 people. The scenario is mostly planar, however a portion of the frame contains a stair. The full calibration has been provided by the organizers, instead we have annotated the number of people for each frame.

Camera	Resolution	FPS	#Annotated	Motion
(1) Analog camcorder	720x576	25p	3000	static
(2) Apple iPhone 4s	1280x720	30p	3600	free
(3) Sony RX100	1920x1080	50i	6000	static
(4) Ixus 300 hs	1280x720	30p	3600	static
(5) Nikon D90	1280x720	24p	2880	static

Table A.2: List of cameras and settings used to acquire the in-house outdoor dataset.

In-house outdoor

The in-house outdoor dataset (which will be soon available at *http://slipguru.disi.unige.it*) is composed by videos acquired from five cameras one of which moving. This dataset has been acquired with the aim of framing typical video surveillance scenario on a non planar world, with hand held moving cameras and partially overlapped FOVs.

Each camera has different resolution and frame rate settings (see Table A.2). Figure A.8 shows the field of view of each camera, whereas Figure A.7 reports their position and the size of the observed area.

The dataset is composed by five videos of two minutes, each frame shows up to 6 actors whose track length range between three seconds and one minute. Each track of each person has been manually annotated.

In-house indoor

The in-house indoor dataset is a set of videos acquired from a video surveillance system [1] with a wide angle lens placed inside our department.

The offline dataset is composed by 90 minutes of videos framing uncontrolled scenes an is sparsely annotated with the number of people. A subset of 15 minutes of videos reports the number of people inside the red polygon of Figure A.9 for each frame.

The videos in the fully annotated sequences contains up to 12 people, the extended dataset contains a sequence with a crowd with more than 40 people. Due to privacy issues the dataset cannot be published.

Some of the proposed algorithms have been integrated in the system to evaluate their ability of processing online information streams.

[1] www.imavis.com

Figure A.7: Approximative points of view and optical axis of the cameras used to acquire our in-house outdoor dataset. The optical axis of camera 2 is noted with two dotted line as the camera moves during the video (map data Google©).

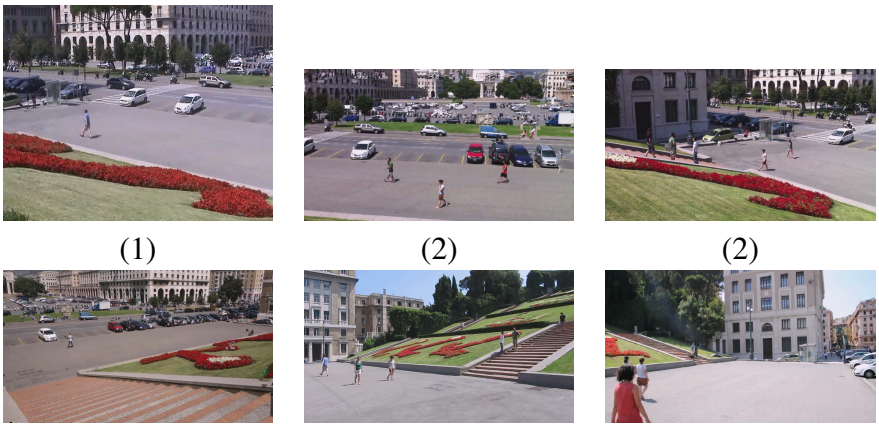


Figure A.8: Example of frames of each camera of the in-house outdoor dataset to show their fields of view. The two images of camera (2) show an example of the change in the FOV during the video.

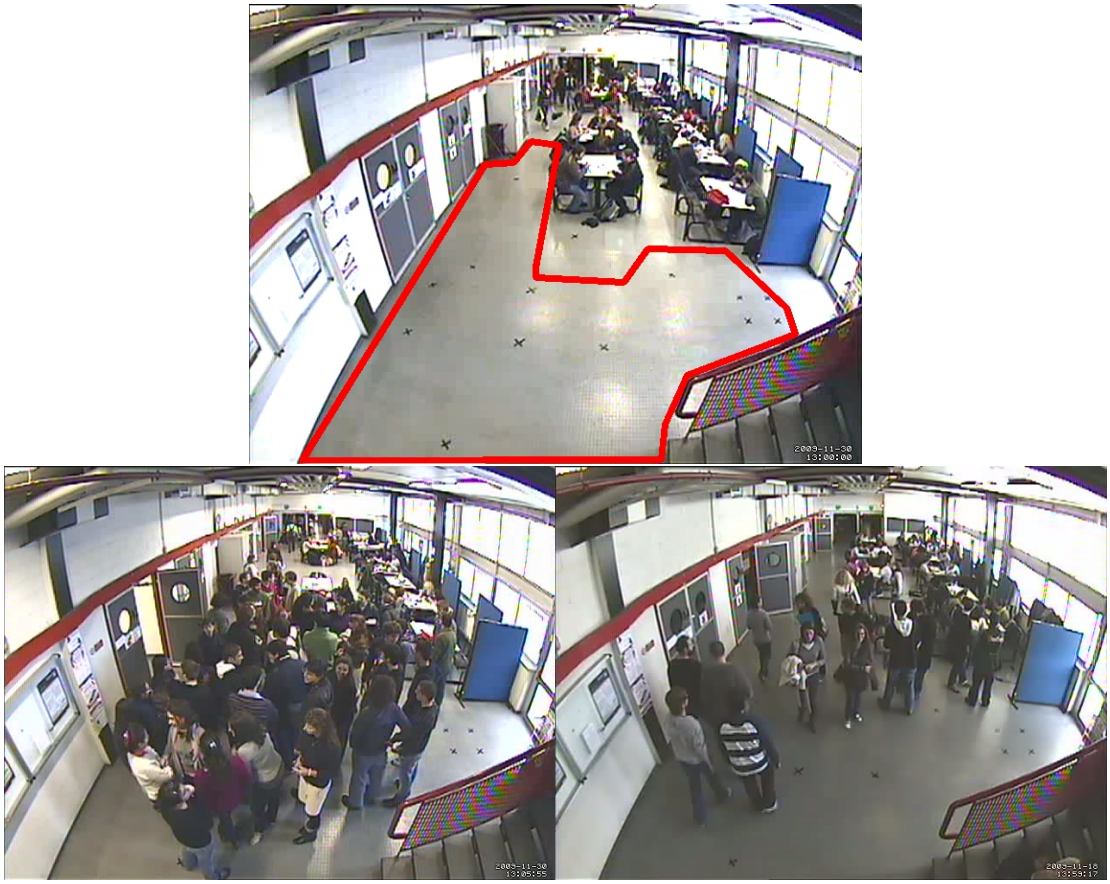


Figure A.9: Frames from the in-house indoor dataset. The red polygon defines the area used to count people.

Bibliography

[AAK71] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range photogrammetry*, volume 1, page 18, 1971.

[AJBV09] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE, 2009.

[ASAM09] A. Albiol, M.J. Silla, A. Albiol, and J.M. Mossi. Video analysis using corner motion statistics. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 31–38, 2009.

[BBS01] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–355. IEEE, 2001.

[BCPM11] L. Bazzani, M. Cristani, A. Perina, and V. Murino. Multiple-shot person re-identification by chromatic and epitomic analyses. *Pattern Recognition Letters*, 2011.

[BCS07] M. Bozzoli, L. Cinque, and E. Sangineto. A statistical method for people counting in crowded environments. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 506–511. IEEE, 2007.

[BD01] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[BE02] J. Black and T. Ellis. Multi-camera image measurement and correspondence. *Measurement*, 32(1):61–71, 2002.

[Bev06] Alessandro Bevilacqua. A novel shadow detection algorithm for real time visual surveillance applications. In *ICIAR (2)*, pages 906–917, 2006.

[BHH11] Sebastian Brutzer, Benjamin Hoferlin, and Gunther Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1937–1944. IEEE, 2011.

[Bla86] S.S. Blackman. Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1, 1986.

[BLK12] O. Barinova, V. Lempitsky, and P. Kholi. On detection of multiple object instances using hough transforms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1773–1784, 2012.

[BMPI05] N.D. Bird, O. Masoud, N.P. Papanikolopoulos, and A. Isaacs. Detection of loitering individuals in public transportation areas. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):167–177, 2005.

[Bre97] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 568–574. IEEE, 1997.

[Bro71] D.C. Brown. Close-range camera calibration. *Photogrammetric engineering*, 37(8):855–866, 1971.

[BTVG06] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.

[BVC11] D. Baltieri, R. Vezzani, and R. Cucchiara. Sarc3d: a new 3d body model for people tracking and re-identification. *Image Analysis and Processing–ICIAP 2011*, pages 197–206, 2011.

[BVC12] D. Baltieri, R. Vezzani, and R. Cucchiara. People orientation recognition by mixtures of wrapped distributions on random trees. *Computer Vision–ECCV 2012*, pages 270–283, 2012.

[CCC06] T.H. Chen, T.Y. Chen, and Z.X. Chen. An intelligent people-flow counting method for passing through a gate. In *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, pages 1–6. IEEE, 2006.

[CCCL06] H.S. Chen, H.T. Chen, Y.W. Chen, and S.Y. Lee. Human action recognition using star skeleton. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 171–178. ACM, 2006.

[CCL04] F. Chang, C.J. Chen, and C.J. Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004.

[CCP08] S. Calderara, R. Cucchiara, and A. Prati. Bayesian-competitive consistent labeling for people surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):354–360, 2008.

[CFB09] S. Choudri, J.M. Ferryman, and A. Badii. Robust background model for pixel based people counting using a single uncalibrated camera. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE, 2009.

[CFPV10] D. Conte, P. Foggia, G. Percannella, and M. Vento. A method based on the indirect approach for counting people in crowded scenes. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 111–118. IEEE, 2010.

[CGO00] T.H. Chang, S. Gong, and E.J. Ong. Tracking multiple people under occlusion using multiple cameras. In *Proc. British Machine Vision Conf*, pages 566–576, 2000.

[CGP+01] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 334–339. IEEE, 2001.

[CH03] Marco Carcassoni and Edwin R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1):193 – 204, 2003.

[CHBY06] P. Chandon, J. Hutchinson, E. Bradlow, and S. Young. Measuring the value of point-of-purchase marketing with commercial eye-tracking data. *INSEAD Business School Research Paper*, 22(2007), 2006.

[CHH06] H. Celik, A. Hanjalic, and E.A. Hendriks. Towards a robust solution to people counting. In *Image Processing, 2006 IEEE International Conference on*, pages 2401–2404. IEEE, 2006.

[CI02] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE TPAMI*, 2002.

[CKY09] S. Choi, T. Kim, and W. Yu. Performance evaluation of ransac family. In *Proceedings of the British Machine Vision Conference*, 2009.

[CLK+00] R.T. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, et al. *A system for video surveillance and monitoring*, volume 102. Carnegie Mellon University, the Robotics Institute, 2000.

[CLV08] A.B. Chan, Z.S.J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[CM02] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[CMV09] A.B. Chan, M. Morrow, and N. Vasconcelos. Analysis of crowded scenes using holistic properties. In *11th IEEE Intl. Workshop on Performance Evaluation of Tracking and Surveillance (PETS'09)*, 2009.

[CPT03] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–721. IEEE, 2003.

[CRM00] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

[CSI06] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *IJCV*, 68(1):53–64, 2006.

[CSS11] Wongun Choi, Khuram Shahid, and Silvio Savarese. Learning context for collective activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3273–3280. IEEE, 2011.

[CT07] E. Candes and T. Tao. The Dantzig selector: statistical estimation when $p$is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[CTTC07] L. Chen, J. Tao, Y.P. Tan, and K.L. Chan. People counting using iterative mean-shift fitting with symmetry measure. In *Information, Communications & Signal Processing, 2007 6th International Conference on*, pages 1–4. IEEE, 2007.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CV12] A.B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *Image Processing, IEEE Transactions on*, 21(4):2160–2177, 2012.

[CWX+10] X. Cao, L. Wu, J. Xiao, H. Foroosh, J. Zhu, and X. Li. Video synchronization and its application to object transfer. *Image and Vision Computing*, 28(1):92–100, 2010.

[DBL11] *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*. IEEE, 2011.

[DF12] C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. *Computer Vision–ECCV 2012*, pages 301–311, 2012.

[DIOV06] E. Delponte, F. Isgrò, F. Odone, and A. Verri. Svd-matching using sift features. *Graphical models*, 68(5):415–431, 2006.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[DMOA09] A. Destrero, C. De Mol, F. Odone, and A.Verri. A sparsity-enforcing method for learning face features. *IEEE Transactions on Image Processing*, 18(1), 2009.

[DPG+10] C. Donatello, F. Pasquale, P. Gennaro, T. Francesco, and V. Mario. A method for counting moving people in video surveillance videos. *EURASIP Journal on Advances in Signal Processing*, 2010, 2010.

[DPL09] E. Dexter, P. Pérez, and I. Laptev. Multi-view synchronization of human actions and dynamic scenes. In *BMVC*, 2009.

[DRCB05] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[DT01] S.L. Dockstader and A.M. Tekalp. Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE*, 89(10):1441–1455, 2001.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, volume 1, pages 886–893. Ieee, 2005.

[DTPB09] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *British machine vision conference*, pages 1–11, 2009.

[DTS06] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *Computer Vision–ECCV 2006*, pages 428–441, 2006.

[DWSP09a] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[DWSP09b] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009.

[DZL06] C. Dai, Y. Zheng, and X. Li. Accurate video alignment using phase correlation. *Signal Processing Letters, IEEE*, 13(12):737–740, 2006.

[EB11] G. Evangelidis and C. Bauckhage. Efficient and robust alignment of unsynchronized video sequences. *Pattern Recognition*, pages 286–295, 2011.

[EDSL11] G.D. Evangelidis, F. Diego, J. Serrat, and A.M. López. Slice matching for accurate spatio-temporal alignment. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1941–1946. IEEE, 2011.

[EF10] A. Ellis and J. Ferryman. Pets2010 and pets2009 evaluation of results using individual ground truthed single views. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 135–142. IEEE, 2010.

[EGVR11] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination for the lasso. *Journal of Machine Learning Research.*, 2011.

[ELVG07] A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[EPP00] T. Evgeniou, M. Pontil, and T. Poggio. Regularization Networks and Support Vector Machines. *Adv. Comput. Math.*, 13(1):1–50, 2000.

[ESK+12] A. Elhayek, C. Stoll, K. Kim, H. Seidel, and C. Theobalt. Feature-based multi-video synchronization with subframe accuracy. *Pattern Recognition*, pages 266–275, 2012.

[EW99] G.A. Einicke and L.B. White. Robust extended kalman filtering. *Signal Processing, IEEE Transactions on*, 47(9):2596–2599, 1999.

[Fai75] W. Faig. Calibration of close-range photogrammetric systems: mathematical formulation. *Photogrammetric engineering and remote sensing*, 1975.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FBP+10] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2360–2367. IEEE, 2010.

[FGM10] P.F. Felzenszwalb, R.B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010.

[FGMR10] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[FMC12] A. Fusiello, Vittorio Murino, and Rita Cucchiara, editors. *Computer Vision - ECCV 2012. Workshops and Demonstrations - Florence, Italy, October 7-13, 2012, Proceedings, Part II*, volume 7584 of *Lecture Notes in Computer Science*. Springer, 2012.

[FP01] H. Farid and A.C. Popescu. Blind removal of lens distortion. *JOSA A*, 18(9):2072–2078, 2001.

[FP06] J.M. Frahm and M. Pollefeys. Ransac for (quasi-) degenerate data (qdegsac). In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 453–460. IEEE, 2006.

[FSM+09] D. Fehr, R. Sivalingam, V. Morellas, N. Papanikolopoulos, O. Lotfallah, and Y. Park. Counting people in groups. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 152–157. IEEE, 2009.

[FT08] A. Farhadi and M. Tabrizi. Learning to recognize activities from the wrong view point. *Computer Vision–ECCV 2008*, pages 154–166, 2008.

[GBS+07] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247–2253, 2007.

[GBTTO12] G. Garcia-Bunster, M. Torres-Torriti, and C. Oberli. Crowded pedestrian counting at bus stops from perspective transformations of foreground areas. *Computer Vision, IET*, 6(4):296–305, 2012.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[GPC12] Giovanni Gualdi, Andrea Prati, and Rita Cucchiara. Multistage particle windows for fast and accurate object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1589–1604, 2012.

[Har97] R.I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997.

[HC03] D. Huang and T.W.S. Chow. A people-counting system using a hybrid rbf neural network. *Neural processing letters*, 18(2):97–113, 2003.

[HCSW09] L. He, Y. Chao, K. Suzuki, and K. Wu. Fast connected-component labeling. *Pattern Recognition*, 42(9):1977–1987, 2009.

[HEH06] D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2137–2144. IEEE, 2006.

[HHD99] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV*, volume 99, pages 256–261, 1999.

[HKM+98] K. Hashimoto, C. Kawaguchi, S. Matsueda, K. Morinaka, and N. Yoshiike. People-counting system using multisensing application. *Sensors and Actuators A: Physical*, 66(1):50–55, 1998.

[HP11] Y.L. Hou and G.K.H. Pang. People counting and human detection in a challenging situation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(1):24–33, 2011.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[HSD73] R.M. Haralick, K. Shanmugam, and I.H. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973.

[HTFF01] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference, and prediction, 2001.

[HZ00] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[IRP94] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[JB08] C.T. Johnston and D.G. Bailey. Fpga implementation of a single pass connected components algorithm. In *Electronic Design, Test and Applications, 2008. DELTA 2008. 4th IEEE International Symposium on*, pages 228–231. IEEE, 2008.

[JDLP11] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *IEEE TPAMI*, 33(1):172–185, 2011.

[JF91] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, 24(12):1167–1186, 1991.

[JJ08] K. Jeong and C. Jaynes. Object matching in disjoint cameras using a color transfer approach. *Machine Vision and Applications*, 19(5):443–455, 2008.

[Kar90] K.P. Karmann. Moving object recognition using an adaptive background memory. *Proc. Time Varying Image Processing*, 1990.

[KB01] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, volume 25, pages 1–5, 2001.

[KCCK02] J.W. Kim, K.S. Choi, B.D. Choi, and S.J. Ko. Real-time vision-based people counting system for the security door. In *International Technical Conference on Circuits/Systems Computers and Communications*, pages 1416–1419, 2002.

[KCHD05] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185, 2005.

[KCM03] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–267. IEEE, 2003.

[KGT05] D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. In *British Machine Vision Conf*. Citeseer, 2005.

[KH11] Reinhard Koch and Fay Huang, editors. *Computer Vision - ACCV 2010 Workshops - ACCV 2010 International Workshops, Queenstown, New Zealand, November 8-9, 2010, Revised Selected Papers, Part II*, volume 6469 of *Lecture Notes in Computer Science*. Springer, 2011.

[Koh01] T. Kohonen. *Self-organizing maps*, volume 30. Springer Verlag, 2001.

[KP99] J.K. Kim and H.W. Park. Statistical textural features for detection of microcalcifications in digitized mammograms. *Medical Imaging, IEEE Transactions on*, 18(3):231–238, 1999.

[KRJ+08] P. Kilambi, E. Ribnick, A.J. Joshi, O. Masoud, and N. Papanikolopoulos. Estimating pedestrian counts in groups. *Computer Vision and Image Understanding*, 110(1):43–59, 2008.

[KS03] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1355–1360, 2003.

[KZP08] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Human activity recognition using a dynamic texture based method. In *British Machine Vision Conference*, volume 1, pages 1–10, 2008.

[Lap05a] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[Lap05b] I. Laptev. On space-time interest points. *IJCV*, 64(2/3), 2005.

[LBH08] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[LBRF10] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Sparse distance learning for object recognition combining rgb and depth information. In *Workshop on Advanced Reasoning with Depth Cameras*, 2010.

[LC10] R. Li and R. Chellappa. Aligning spatio-temporal signals on a special manifold. *Computer Vision–ECCV 2010*, pages 547–560, 2010.

[LH87] HC Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pages 61–62, 1987.

[LH05] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1482–1489. IEEE, 2005.

[Lin93] T. Lindeberg. *Scale-space theory in computer vision*. Springer, 1993.

[Lin98] T. Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[LKZI12] X. Lin, V. Kitanovski, Q. Zhang, and E. Izquierdo. Enhanced multi-view dancing videos synchronisation. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2012 13th International Workshop on*, pages 1–4. IEEE, 2012.

[LL06] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *SCVMA 2004.*, volume 3667, page 91. Springer-Verlag New York Inc, 2006.

[LLC09] H.H. Lin, T.L. Liu, and J.H. Chuang. Learning a scene background model via classification. *Signal Processing, IEEE Transactions on*, 57(5):1641–1654, 2009.

[LLF05] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 775–781. IEEE, 2005.

[LM10] C. Lu and M. Mandal. Efficient temporal alignment of video sequences using unbiased bidirectional dynamic time warping. *Journal of Electronic Imaging*, 19(4):0501, 2010.

[LM11] C. Lu and M. Mandal. An efficient technique for motion-based view-variant video sequences synchronization. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[LMSR08] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[LNM+12] P. Lanza, N. Noceti, C. Maddaleno, A. Toma, L. Zini, and F. Odone. A vision-based navigation facility for planetary entry descent landing. In Fusiello et al. [FMC12], pages 546–555.

[LOV09] L.Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML*, 2009.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[LPF04] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–244. IEEE, 2004.

[LSS05] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005.

[LTR+05] X. Liu, PH Tu, J. Rittscher, A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 306–311. IEEE, 2005.

[Mäe03] T. Mäenpää. The local binary pattern approach to texture analysis: Extenxions and applications, 2003.

[MAT10] D. Merad, K.E. Aziz, and N. Thome. Fast people counting using head detection from skeleton graph. In *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 151–156. IEEE Computer Society, 2010.

[MBM09] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2009.

[MD01] A. Mittal and L. Davis. Unified multi-camera detection and tracking using region-matching. In *Multi-Object Tracking, 2001. Proceedings. 2001 IEEE Workshop on*, pages 3–10. IEEE, 2001.

[MG06] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11), 2006.

[MLHT04] R. Ma, L. Li, W. Huang, and Q. Tian. On pixel count based crowd density estimation for visual surveillance. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 170–173. IEEE, 2004.

[MMH06] K. Morioka, X. Mao, and H. Hashimoto. Global color model based object matching in the multi-camera environment. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2644–2649. IEEE, 2006.

[MMTV09] C. De Mol, S. Mosci, M. Traskine, and A. Verri. A regularized method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, 16(5), 2009.

[MRS+10] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa. Solving structured sparsity regularization with proximal methods. In *ECML*, 2010.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[MV07] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA, 2007.

[MVCL97] AN Marana, SA Velastin, LF Costa, and RA Lotufo. Estimation of crowd density using image processing. In *Image Processing for Security Applications (Digest No.: 1997/074), IEE Colloquium on*, pages 11–1. IET, 1997.

[MVCL98] AN Marana, SA Velastin, L.F. Costa, and RA Lotufo. Automatic estimation of crowd density using texture. *Safety Science*, 28(3):165–175, 1998.

[NAMF12] Rebagliati Nicola, Solé-Ribalta Albert, Pelillo Marcello, and Serratosa Francesc. Computing the graph edit distance using dominant sets. In *Pattern Recognition International Conference on*. IEEE, 2012.

[NW70] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[OF97] B. A. Olshausen and D. J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 1997.

[OPH94] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585. IEEE, 1994.

[PBF10] Enrico Puppo, A. Brogni, and Leila De Floriani, editors. *Eurographics Italian Chapter Conference 2010, Genova, Italy, 2010*. Eurographics, 2010.

[PCSK10] F.L.C. Pádua, R.L. Carceroni, G.A.M.R. Santos, and K.N. Kutulakos. Linear sequence-to-sequence alignment. *IEEE TPAMI*, 32(2):304–320, 2010.

[PES10] M. Patzold, R.H. Evangelio, and T. Sikora. Counting people in crowded environments by fusion of shape and motion information. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 157–164. IEEE, 2010.

[PGX+08] B. Prosser, S. Gong, T. Xiang, et al. Multi-camera matching under illumination change over time. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, 2008.

[Pil97] M. Pilu. A direct method for stereo correspondence based on singular value decomposition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 261–266. IEEE, 1997.

[PKS09] R. Nevatia P. K. Sharma, C. Huang. Evaluation of people tracking, counting, and density estimation in crowded environments. In *11th IEEE Intl. Workshop on Performance Evaluation of Tracking and Surveillance (PETS'09)*, 2009.

[PM97] B. Prescott and GF McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1):39–47, 1997.

[PMTC03] A. Prati, I. Mikic, M.M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(7):918–923, 2003.

[Pop10] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[PP00] C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 2000.

[PRF10] Dennis Park, Deva Ramanan, and Charless Fowlkes. Multiresolution models for object detection. In *ECCV*, 2010.

[PSZ08] S. Paisitkriangkrai, C. Shen, and J. Zhang. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Trans. on Circuits and Sys for Video Technology*, 18(8), 2008.

[PWS07] T.V. Pham, M. Worring, and A.W.M. Smeulders. A multi-camera visual surveillance system for tracking of reoccurrences of people. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 164–169. IEEE, 2007.

[RAG04] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[RE95] P.L. Rosin and T. Ellis. Image difference threshold strategies and shadow detection. In *Proceedings of the 6th British Machine Vision Conference*, volume 1, pages 347–356. Citeseer, 1995.

[RGSSM03] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *IEEE ICCV*. IEEE, 2003.

[RNC06] H. Rahmalan, M.S. Nixon, and J.N. Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540–545. IET, 2006.

[RTK05] J. Rittscher, P.H. Tu, and N. Krahnstoever. Simultaneous estimation of segmentation and shape. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 486–493. IEEE, 2005.

[SB08] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[SBWS10] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski. Synchronization of multiple camera videos using audio-visual features. *IEEE TMM*, 12(1):79–92, 2010.

[SC78] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics Speech and Sig. Proc.*, 26(1):43–49, 1978.

[SDLÁ07] J. Serrat, F. Diego, F. Lumbreras, and J. Álvarez. Synchronization of video sequences from free-moving cameras. *Pattern Recognition and Img Analysis*, pages 620–627, 2007.

[Ser82] J. Serra. *Image analysis and mathematical morphology*. London.: Academic Press.[Review by Fensen, EB in: J. Microsc. 131 (1983) 258.] Cell size, Staining Microscopy Technique, Mathematics, General article Review article (PMBD, 185707888), 1982.

[SG99] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[SHS03] K. Suzuki, I. Horiba, and N. Sugie. Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1):1–23, 2003.

[SI07] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[SKHD09] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human detection using partial least squares analysis. In *ICCV*, 2009.

[SLBS06] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pages 70–70. IEEE, 2006.

[SLH91] G.L. Scott and H.C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 244(1309):21–26, 1991.

[SMB92] L.S. Shapiro and J. Michael Brady. Feature-based correspondence: an eigenvector approach. *Image and vision computing*, 10(5):283–288, 1992.

[SMS96] AJ Schofield, PA Mehta, and T.J. Stonham. A system for counting people in video images using neural networks to identify the background scene. *Pattern Recognition*, 29(8):1421–1428, 1996.

[ST94] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[ST03] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–259. IEEE, 2003.

[Ste97] G.P. Stein. Lens distortion calibration using point correspondences. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 602–608. IEEE, 1997.

[Ste99] G.P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *IEEE CVPR.*, volume 1. IEEE, 1999.

[SVG08] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require? In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[SWFS03] M. Seki, T. Wada, H. Fujiwara, and K. Sumi. Background subtraction based on cooccurrence of image variations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–65. IEEE, 2003.

[SWP09] G. Sen, L. Wei, and Y.H. Ping. Counting people in crowd open scene based on grey level dependence matrix. In *Information and Automation, 2009. ICIA'09. International Conference on*, pages 228–231. IEEE, 2009.

[SZ97] C. Schmid and A. Zisserman. Automatic line matching across views. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 666–671. IEEE, 1997.

[TBF+12] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R.J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2012.

[TC02] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. *Computer Vision—ECCV 2002*, pages 68–81, 2002.

[TC11] M. Taj and A. Cavallaro. Distributed and decentralized multi-camera tracking: a survey. *IEEE Sign Proc Magazine*, 28, 2011.

[Tel04] A. Telea. An image inpainting technique based on the fast marching method. *JOURNAL OF GRAPHICS TOOLS.*, 9(1):23–34, 2004.

[TH08] C. Thurau and V. Hlavác. Pose primitive based human action recognition in videos or still images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1), 1996.

[TK91] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ., 1991.

[TLW+07] J. Tang, D. Liang, N. Wang, et al. A laplacian spectral method for stereo correspondence. *Pattern recognition letters*, 28(12):1391–1399, 2007.

[Tor02] P.H.S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002.

[TPM08a] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1713–1727, 2008.

[TPM08b] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on PAMI*, 30(10), 2008.

[TR09] P.A. Tresadern and I.D. Reid. Video synchronization from human motion using rank constraints. *Computer Vision and Image Understanding*, 113(8):891–906, 2009.

[TS08] D. Tran and A. Sorokin. Human activity recognition with metric learning. *Computer Vision–ECCV 2008*, pages 548–561, 2008.

[Tsa87] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[TVG00] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *british Machine vision conference*, pages 412–425, 2000.

[TVG04] T. Tuytelaars and L. Van Gool. Synchronizing video sequences. In *IEEE CVPR*, volume 1, pages I–762. Ieee, 2004.

[TZ00] P.H.S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[TZW11] B. Tan, J. Zhang, and L. Wang. Semi-supervised elastic net for pedestrian counting. *Pattern Recognition*, 44(10):2297–2304, 2011.

[UI06] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. *Computer Vision–ECCV 2006*, pages 538–550, 2006.

[UOD06] M. Ushizaki, T. Okatani, and K. Deguchi. Video synchronization based on co-occurrence of appearance changes in video sequences. In *Proc. of IEEE ICPR*, 2006.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[VJS05] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[VTH06] S. Velipasalar, Y.L. Tian, and A. Hampapur. Automatic counting of interacting people by using a single uncalibrated camera. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1265–1268. IEEE, 2006.

[WB95] G. Welch and G. Bishop. An introduction to the kalman filter, 1995.

[WBR07] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7. IEEE, 2007.

[WCH92] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on pattern analysis and machine intelligence*, 14(10):965–980, 1992.

[WH99] Y. Wu and T. Huang. Vision-based gesture recognition: A review. *Gesture-based communication in human-computer interaction*, pages 103–115, 1999.

[WH06] H.F. Wang and E.R. Hancock. Correspondence matching using kernel principal components analysis and label consistency constraints. *Pattern recognition*, 39(6):1012–1025, 2006.

[WHK07] D. Wedge, D. Huynh, and P. Kovesi. Using space-time interest points for video sequence synchronization. In *IAPR*, 2007.

[WHY09] X. Wang, T.X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[WJY+11] Q. WEN, C. JIA, Y. YU, G. CHEN, Z. YU, and C. ZHOU. People number estimation in the crowded scenes using texture analysis based on gabor filter. *Journal of Computational Information Systems*, 7(11):3754–3763, 2011.

[WLLX06] X. Wu, G. Liang, K.K. Lee, and Y. Xu. Crowd density estimation using texture analysis and learning. In *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pages 214–219. IEEE, 2006.

[WMSS10] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1030–1037. IEEE, 2010.

[WOS09] K. Wu, E. Otoo, and K. Suzuki. Optimizing two-pass connected-component labeling algorithms. *Pattern Analysis & Applications*, 12(2):117–135, 2009.

[WS08] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM*, 2008.

[WVDM00] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

[WYZ10] F. Wu, Y. Yuan, and Y. Zhuang. Heterogeneous feature selection by group lasso with logistic regression. In *Proc Int Conf on Multimedia*, 2010.

[WZ06] L. Wolf and A. Zomet. Wide baseline matching between unsynchronized video sequences. *IJCV*, 68(1):43–52, 2006.

[XLH06] L. Xiaohua, S. Lansun, and L. Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3):299–308, 2006.

[YB98] Y. Yacoob and M.J. Black. Parameterized modeling and recognition of activities. In *Computer Vision, 1998. Sixth International Conference on*, pages 120–127. IEEE, 1998.

[YCSX08] S. Yu, X. Chen, W. Sun, and D. Xie. A robust method for detecting and counting people. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, pages 1545–1549. IEEE, 2008.

[YGBG03] D.B. Yang, H.H. González-Baños, and L.J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 122–129. IEEE, 2003.

[YKS08] P. Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[YL06] M. Yuan and Y. Lin. model selection and estimation in regression with grouped variables. *J.R.Stat.Soc Ser.B*, 68, 2006.

[YLY12] M. Yang, Y. Liu, and Z. You. Video synchronization based on events alignment. *Pattern Recognition Letters*, 2012.

[YST10] S. Yoshinaga, A. Shimada, and R. Taniguchi. Real-time people counting using blob descriptor. *Procedia-Social and Behavioral Sciences*, 2(1):143–152, 2010.

[YSZ+11] H. Yang, H. Su, S. Zheng, S. Wei, and Y. Fan. The large-scale crowd density estimation based on sparse spatiotemporal local binary pattern. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[YZ07] W. Ye and Z. Zhong. Robust people counting in crowded environment. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 1133–1137. IEEE, 2007.

[ZAYC06] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng. Fast human detection using a cascade of histogram of oriented gradients. In *CVPR*, 2006.

[ZCO13] L. Zini, A. Cavallaro, and F. Odone. Action-based multi-camera synchronization (submitted). *Journal on Emerging and Selected Topics in Circuits and Systems*, 2013.

[ZDFL95] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1-2):87–119, 1995.

[ZH05] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[Zha04] Z. Zhang. Camera calibration with one-dimensional objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):892–899, 2004.

[ZL02] D. Zhang and G. Lu. Shape-based image retrieval using generic fourier descriptor. *Signal Processing: Image Communication*, 17(10):825–848, 2002.

[ZM10] C. Zeng and H. Ma. Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2069–2072. IEEE, 2010.

[ZNO10] L. Zini, N. Noceti, and F. Odone. An adaptive video surveillance architecture for behavior analysis. In Puppo et al. [PBF10], pages 57–64.

[ZO11] L. Zini and F. Odone. Efficient pedestrian detection with group lasso. In *ICCV Workshops* [DBL11], pages 1777–1784.

[ZOCed] L. Zini, F. Odone, and A. Cavallaro. Multi-view matching of articulated objects. *Transactions on Circuits and Systems for Video Technology*, submitted.

[ZON13] L. Zini, F. Odone, and N. Noceti. Precise people counting in real-time (submitted). In *Image Processing, 2013 IEEE International Conference on*. IEEE, 2013.

[ZOV+10] L. Zini, F. Odone, A. Verri, P. Lanza, and A. Marcer. Relative pose estimation for planetary entry descent landing. In Koch and Huang [KH11], pages 255–264.