UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA APLICADA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUSTAVO FÜHR

Pedestrian Tracking and Collective Behavior Recognition

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dr. Cláudio Rosito Jung

Porto Alegre
April 2017

CIP – CATALOGING-IN-PUBLICATION

Führ, Gustavo

Pedestrian Tracking and Collective Behavior Recognition / Gustavo Führ. – Porto Alegre: PPGC da UFRGS, 2017.

101 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2017. Advisor: Cláudio Rosito Jung.

1. Pedestrian tracking. 2. People detection. 3. Collective behavior. 4. Group activity. 5. Self-calibration. 6. Surveillance systems. I. Jung, Cláudio Rosito. II. Título.

ABSTRACT

Collective behavior detection and pedestrian tracking present many applications, specially in surveillance systems. In this dissertation, we proposed a complete pipeline for achieving robust tracking and collective behavior recognition based on calibrated static cameras.

To remove the necessity of manual calibration, we first present a fully automatic self-calibration system that explores pedestrian detection results and background removal at non-consecutive frames in order to calibrate a static camera using a non-linear cost function. We also propose the use of camera calibration to generate geometrically coherent candidates for pedestrian detection. Our approach aims to reduce the scale range typically used in sliding-window techniques, which leads to less feature extractions and decreased number of false positives.

Then, we propose a multi-target pedestrian tracking algorithm using a calibrated static camera. The tracking approach explores color histograms to track patches of each target. Obtained displacement vectors are combined with the expected motion of pedestrians in the world coordinate system. The proposed tracker also incorporates pedestrian detector results to improve the system's accuracy and its ability to recover from failure.

Finally, we propose a two-layered approach for collective behavior recognition based on Random Forests classifiers. In the first level, we use inter-personal distances and relative speeds computed in the world coordinate system to classify asymmetrical pair interactions. Those interactions are combined with group shape dynamics and mean velocity to recognize the collective behavior. We devise a set of experiments to attest the quality of our approaches using publicly available datasets. Results have shown to be competitive against state-of-the-art techniques, and particularly of good generalization across different databases.

Keywords: Pedestrian tracking. people detection. collective behavior. group activity. self-calibration. surveillance systems.

Rastreamento de Pedestres e Análise de Comportamento Coletivo

RESUMO

A análise de comportamento coletivo e rastreamento de pedestres apresentam diversas aplicações, especialmente em sistemas de vigilância inteligente. Neste trabalho é proposta uma solução compreensiva com objetivo de atingir rastreamento de pedestre e reconhecimento de atividade coletiva de maneira robusta baseada na utilização de câmeras calibradas.

Primeiramente, com o objetivo de remover a necessidade de calibração manual, nós apresentamos um método de calibração automática que explora detectores de pedestres e remoção de fundo para calibragem baseada em otimização não-linear. Adicionalmente, nós propomos a utilização da matriz de calibração para gerar candidatos coerentes com a geometria de cena em detectores de pedestres. Nossa abordagem tem como objetivo diminuir o intervalo de escalas comumente utilizado em detectores baseados em janelas deslizantes, gerando um número menor de extrações de atributos e reduzindo o número de falsos positivos na detecção.

Em seguida, nós propomos um método de rastreamento de múltiplos pedestres utilizando câmeras calibradas. Nossa abordagem explora histogramas de cor para rastrear os pequenas regiões (*patches*) de cada alvo. Os vetores de deslocamento obtidos através do pareamento de atributos de aparência são combinados com um vetor obtido através de um preditor de movimento em coordenadas de mundo. Adicionalmente, nós incluímos informações originárias de detectores de pedestres para aumentar a acurácia do sistema e sua habilidade de recuperação a falhas.

Por fim, nós propomos uma abordagem hierárquica de duas camadas para o problema de reconhecimento de atividade coletiva baseada no uso de classificadores *Random Forests*. No primeiro nível da técnica proposta, nós utilizamos distâncias entre pares de pessoas e suas respectivas velocidades relativas para classificar interações de pares. Estas interações são combinadas com a dinâmica do formato do grupo observado (e sua respectiva velocidade) para o reconhecimento de atividades coletivas. Os experimentos realizados neste trabalho demonstram a qualidade de nossas abordagens em sequências de vídeos disponíveis publicamente. Nossos resultados mostram serem competitivos quando comparados com técnicas do estado da arte e, particularmente, apresentam uma boa generalização entre diferentes cenários de captura de vídeo.

Palavras-chave: rastreamento de pedestres, detecção de pessoas, análise de comportamento, calibração de câmera vigilância de vídeo.

LIST OF ABBREVIATIONS AND ACRONYMS

PCA	Principal Component Analysis
WCS	World Coordinate System
ICS	Image Coordinate System
SVM	Support Vector Machine
WVMF	Weighted Vector Median Filter
HOG	Histograms of Oriented Gradients
PTZ	Pan-tilt-zoom
CCTV	Closed-Circuit TeleVision
RANSAC	Random Sample Consensus
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
HMM	Hidden Markov Model
GUI	Graphical User Interface
PID	Personal Interaction Descriptors
CBD	Collective Behavior Descriptors
KDE	Kernel Density Estimation
PDF	Probability Density Function
FPPI	False Positive per Image
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision

LIST OF FIGURES

LIST OF TABLES

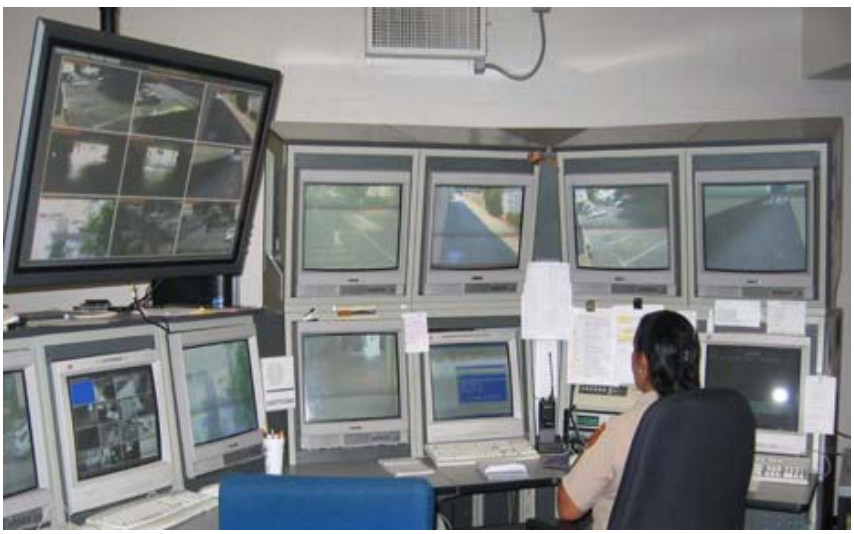
CONTENTS

1 INTRODUCTION

1.1 Motivation

In recent decades, the rapid increase in the number of cameras distributed in both outdoor and indoor environments has prompted a necessity for processing an enormous quantity of data in a manner that is both automatic and swift. The traditional protocol in video surveillance, which consists of showing a mosaic of streams on a screen monitored by a human operator (see Fig. 1.1), has been proved to be ineffective. A studied performed by the National Institute of Justice of the United States (GREEN, 2005) indicated the attention of an operator in such systems significantly drops after only 20 minutes of watching and evaluating a security video sequence. Therefore, it is no surprising that the problem of obtaining high-level information from the streams of (an ever increasing number of) cameras in public spaces has raised the interest of the research community in the last few decades. In the context of video surveillance, there are several possible applications that could benefit from computer vision algorithms. The scope of such applications is actually very broad (HAERING; VENETIANER; LIPTON, 2008): many focus on physical security and law enforcement (LIPTON, 2005), such as in applications that aim to detect intruders (LIM; TANG; CHAN, 2014), abandoned objects (FERRYMAN et al., 2013), people loitering (ARROYO et al., 2015), among others; additionally, there are applications concerning traffic control (XIA et al., 2016), video synopsis (RAV-ACHA; PRITCH; PELEG, 2006; LEE; GRAUMAN, 2015) and even retail analytics (DENMAN et al., 2012)– often performed by counting people and recognizing gender, customer behavior and facial expressions.

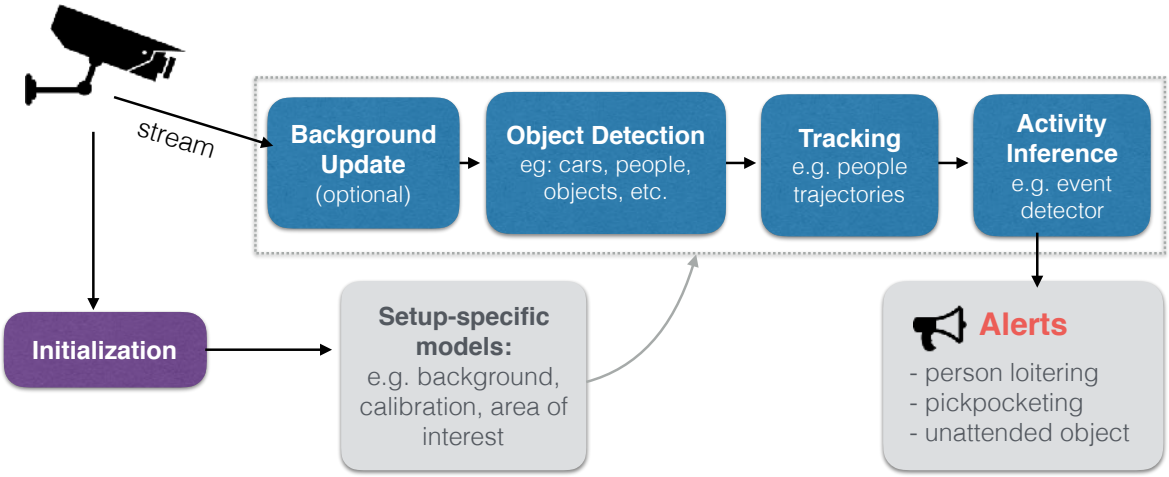
Figure 1.1 – Classical surveillance systems are inherently flawed. Often, the operator is overwhelmed with data.



Source: (SEBE et al., 2003)

Although there are many possible applications in the context of video surveillance, the typical pipeline involves a set of common problems, as highlighted in Figure 1.2. Often, surveillance systems will present an initialization step to estimate/compute a set of parameters that will be fixed along the execution of the system, such as camera parameters (assuming static cameras with no zoom), areas of interest in the scene, an initial background model, etc. – these can be computed automatically or be inputted manually by an operator. At runtime, the majority of the systems employs background segmentation on the frame and/or object detection to detect the entities that are important in the context of the application (e.g. people, vehicles, etc.). Tracking the relevant objects detected in the previous step is also a common task, aiming to obtain information about their dynamics. Activity inference (or some type of high level semantic recognition) often uses classification or direct trajectory analysis to infer events involving the entities of interest, such as loitering detection, fighting, objects invading restricted zones and unusual events, to name a few. Finally, alerts and statistics can be created using collected data.

Figure 1.2 – A traditional pipeline of intelligent surveillance applications.



Source: Author

1.2 Problem description and dissertation goals

As mentioned before, the scope of problems tackled in video surveillance applications is very broad. They might involve single or multi-camera setups, fixed or moving cameras, focus on different types of objects (such as pedestrians, bicycles, vehicles, etc.), and finally extract and analyze information at different "levels", ranging from direct analysis of tracked trajectories to high-level actions that involve human body parts detection, such as punching or kicking.

The main focus of this dissertation is to detect and recognize collective behaviors of

people in a video sequence captured by a single static camera. To reach this final goal, the dissertation also tackles two important problems also present in surveillance pipelines: camera calibration and multiple pedestrian tracking.

In fact, the problem of detecting group activities in a video sequence has attracted the attention of the computer vision community in the past years, yet it remains mostly an open problem. Besides video surveillance, there are many other applications that could benefit from a robust solution to this problem, such as traffic monitoring and video indexing by semantic context, to name a few. In most of the solutions proposed in the literature, a pedestrian tracker is used first to extract the trajectories, which are then analyzed and classified into different sets that usually correspond to group activities or event classes.

In this dissertation, we are mainly interested in the analysis of group behavior in surveillance scenarios by exploring the ground plane trajectories obtained from a sequence. We propose two types of descriptors that capture pair-wise and collective information from trajectories using a hierarchical strategy. To infer the interaction of each pair of subjects, we extract their relative distance and speed within a temporal window. Instead of using image coordinates, we use ground plane positions in order to have real-world metric measurements and the ability to generalize between different camera setups. Furthermore, the use of real-world distance allows us to infer the "level" of interaction between a pair of agents using known psycho-social studies. For example, Hall (HALL, 1973) introduced the concept of proxemics, which is a personal space (spatial region) that each person tends to preserve. The radii of these regions depend on the kind of relationship between the agent and the neighbor, which would be very hard to estimate based solely on image coordinates. Once the pair-wise features are extracted, they are fed to a classifier in a subsequent step to classify each pair-wise interaction. In a second stage, we compute an histogram from these interactions and augment this data with the group shape dynamics, mean velocities and current positions of all the people involved in the interaction. The classification is divided in two layers and Random Forests are used to combine the different kinds of information.

The use of ground plane coordinates of pedestrian tracks requires a pedestrian tracking algorithm and also a calibrated camera, so that mapping from image to plane coordinates is possible. Although there are many camera calibration approaches that explore calibration patterns (ZHANG, 2000; TERAMOTO; XU, 2002), the development of self-calibration methods that explore real world information without manual intervention has shown to be very important in practical surveillance applications (WANG, 2013). For instance, when using pan-tilt-zoom (PTZ) cameras, it is common for a human operator to change the camera settings aiming to monitor a specific point of interest. In that case, manual re-calibration would be impractical, and

automatic self-calibration could be used.

This work also tackles the problem of self-calibrating a single static camera by exploring information about pedestrians present in the sequence. First, a detector is used to extract pedestrian bounding boxes, which are fed to a "people poles" extractor that employs background segmentation and PCA (Principal Component Analysis) to extract the vertical orientation and height of the pedestrians. These poles are then used to estimate an initial calibration that is improved using a non-linear optimization procedure.

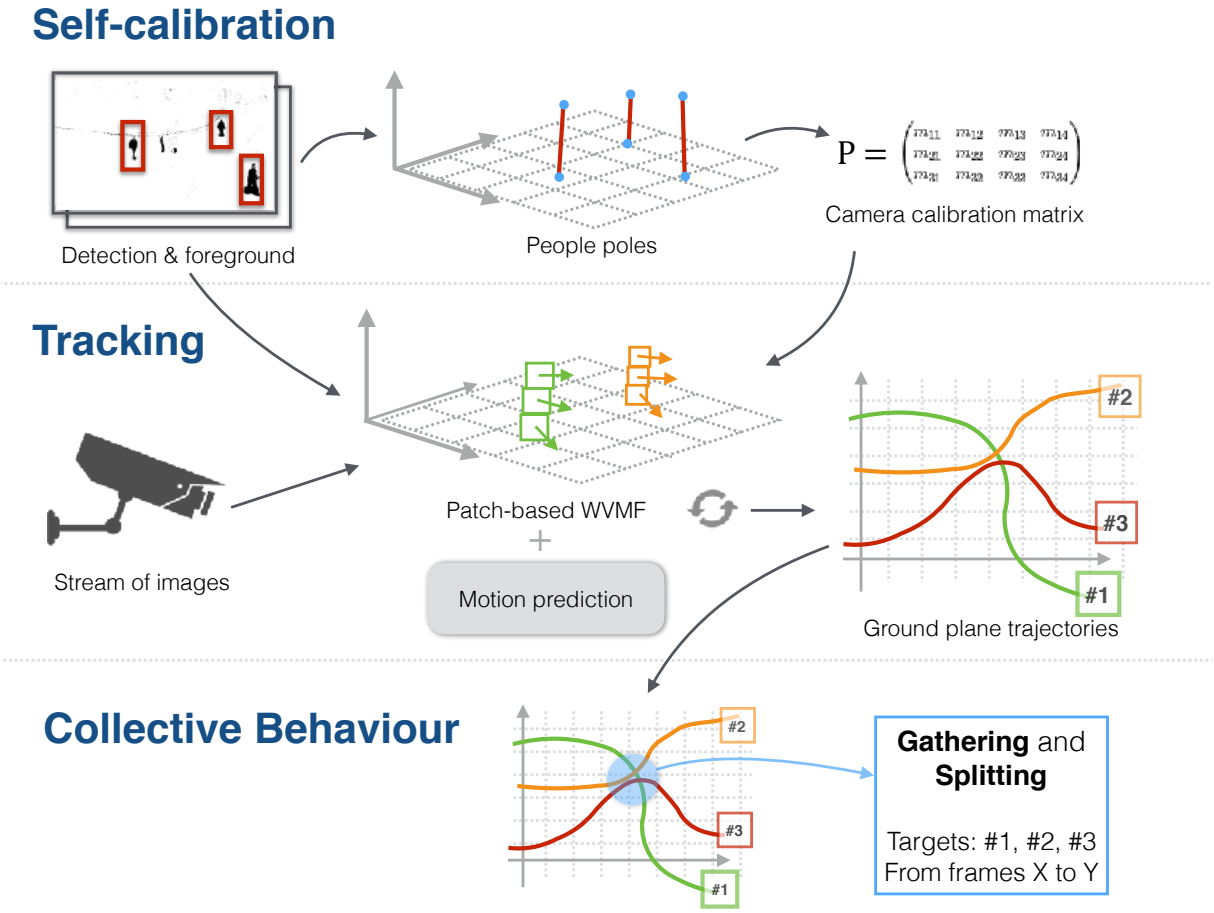
Finally, this dissertation also tackles the problem of multiple pedestrian tracking. A robust tracking system should be able to cope with a high number of occlusions, different camera setups, image noise, etc. It is also very important to detect the pedestrians as soon as they enter the scene, initialize their trackers and terminate them when they leave. To handle the pedestrian tracking problem, we developed an iterative approach that explores a calibrated camera and the expected ground-plane motion of a standing pedestrian to reduce the algorithm complexity and achieve near-real time performance while still being a causal approach (i.e. no latency). We mix information from a pedestrian detector, a patch-based template matching tracker and motion prediction in a framework that combine displacement vectors robustly in the world coordinate frame. The use of calibration is again used to reduce the search area for the targets displacement and their scale variations.

Figure 1.3 summarizes the main components of the proposed pipeline. At the initialization step, a person detector and a background segmentation method serve as input for the extraction of the so-called people poles. These poles are assumed to be vertically parallel in the world and can used to generate the camera calibration matrix provided by a two-stage fully automatic self-calibration procedure. Pedestrian detectors and foreground masks are also used in the tracking phase. To increase performance and accuracy, we reconstruct 2D patch-based image displacements in the World Coordinate System (WCS) using the camera calibration. This set of vectors is augmented using motion prediction and detected pedestrians nearby the current target location to better handle occlusions and target hijacking. Finally, the ground-plane locations of tracked pedestrians are used to identify pair-wise interactions, and then classify the behavior of a group of individuals during the video sequence.

1.3 Dissertation contributions

The main goal of this work is to identify collective behaviors in a video sequence acquired by a single static camera. The main contributions of this dissertation are related to the three

Figure 1.3 – The main components of our proposed pipeline. In this dissertation, we show contributions in all these three problem.



Source: Author

problems described in Section 1.2, which are related to the three main steps in a collective behavior recognition system as shown in Figure 1.3, namely:

- Development of a fully automatic self-calibration system based on background segmentation, people detection and a non-linear optimization phase. The method is shown to correctly estimate the camera projection matrix. We also proposed the use of calibration to the problems of people tracking and detection (FÜHR; JUNG, 2015; FÜHR; JUNG; PAULA, 2016).
- Development of a robust multi-target pedestrian tracker by exploring displacement vectors in the WCS, which are possible to obtain when calibrated cameras are used. We show experimentally that our tracker is both fast and accurate without the drawback of non-causality presented by some modern alternatives based on tracking-by-detection (FÜHR; JUNG, 2012; FÜHR; JUNG, 2014).
- Development of a collective behavior recognition method using a two-stage Random

Forests classification scheme, which is able to extract pair-wise interactions and then collective behaviors. The proposed method uses very compact descriptors based on WCS metrics, and our experiments indicate global accuracy comparable to or better than stat-of-the-art competitors, being also able to generalize surprisingly well across different datasets.

2 RELATED WORK

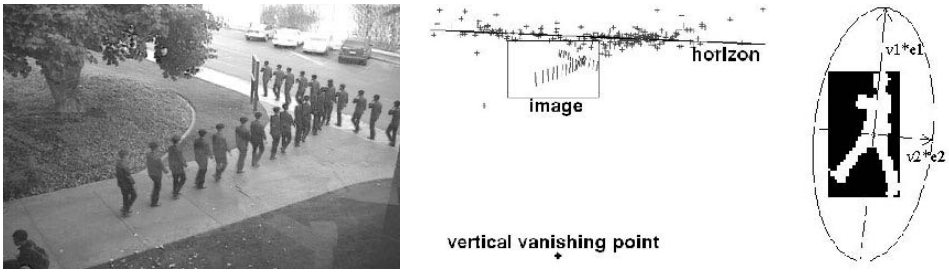
As it was discussed in the previous chapter, our main contributions are in three different yet closely related problems. Therefore, our review of the specialized literature is divided based on the modules illustrated in Figure 1.3. First, Section 2.1 covers some methods related to camera self-calibration. In Section 2.2, we review the state-of-the-art in pedestrian tracking. Finally, Section 2.3 discusses some recent works on collective behavior recognition.

2.1 Self-calibration

Camera calibration is a widely studied problem in computer vision, and most existing approaches rely on a set of calibration patterns (TSAI, 1987; ZHANG, 2000; DOUXCHAMPS; CHIHARA, 2009). Self-calibration approaches, on the other hand, explore features that can be extracted from image objects in a given context, such as buildings (KIM; KWEON, 2009), roads (KANHERE; BIRCHFIELD, 2010), or the geometry of tennis courts (YU et al., 2009). In particular, this brief review focuses on approaches that explore the expected pose or motion of pedestrians for camera self-calibration. Some of the foundations for later work on self-calibration using people appearing in the scene was introduced by Cipolla et al. (CIPOLLA; DRUMMOND; ROBERTSON, 1999), who used three vanishing points computed from lines extracted from architectural scenes. Also, it is worthwhile to mention the study of Zhang on camera calibration using one-dimensional objects (ZHANG, 2004).

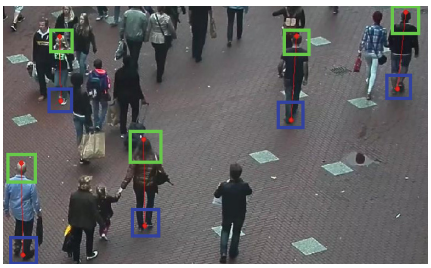
The work of Lv. et al. (LV; ZHAO; NEVATIA, 2002b; LV; ZHAO; NEVATIA, 2006) proposed to calibrate a camera using a single pedestrian that it is observed at several locations in the scene. The feet and head points of the pedestrian are first extracted using background segmentation. Then, the method analyzes the eigenvalues of the covariance matrix computed from the foreground pixel positions, and their ratio is computed for several walking cycles of the same subject. These values are temporally arranged to form a series whose minimum is achieved when the person legs cross. At these minima, the lines from feet and head are stored to compute the horizon line and the vertical vanishing point of the image. These measurements serve as input for the calibration computation. Some of the method's steps are illustrated in Figure 2.1(a). Their method, however, does not handle noisy measurements very well and assumes a single person scenario. Kusakunniran and colleagues (KUSAKUNNIRAN; LI; ZHANG, 2009) use the same approach for feet and head extraction, yet aim at computing the projection matrix directly, without extracting focal length, optical center, etc.

Figure 2.1 – Main components of selected self-calibration techniques found in the literature. (a) The seminal work of Lv. et al (LV; ZHAO; NEVATIA, 2002a) uses pedestrians as vertical poles and estimates the calibration using the horizon line and vanishing points. (b) Hoiem and colleagues (HOIEM; EFROS; HEBERT, 2008) aim to estimate the different surfaces of the scene to improve detections. (c) (BROUWERS et al., 2016) uses head and feet detectors to estimate pedestrian orientation for self-calibration.



(a) Source: (LV; ZHAO; NEVATIA, 2002a)

(b) Source: (HOIEM; EFROS; HEBERT, 2008)



(c) Source: (BROUWERS et al., 2016)

Krahnstoever and Mendonça (KRAHNSTOEVER; MENDONçA, 2005) proposed a self-calibration method that uses information of pedestrians in terms of foot-to-head homologies. To perform calibration, these measurements are fed to a Bayesian filter, which is also used to model error and outliers. However, their algorithm requires prior knowledge about the calibration parameters. The work proposed in (JUNEJO; FOROOSH, 2006) also employs the idea of homologies to solve auto-calibration. Their approach is a linear one, and is somewhat similar to the work of Lv. et al. (LV; ZHAO; NEVATIA, 2006) due to the fact that the authors also use the horizon line and vertical vanishing points to compute the projection matrix. Also, temporal consistency and a single person setup are requirements of the method. A follow-up was proposed in (JUNEJO; FOROOSH, 2007), where the calibration step was integrated with path modeling and surveillance (although the calibrated camera was not explored for pedestrian detection/tracking).

Micusík and Pajdla (MICUSIK; PAJDLA, 2010) presented an approach for camera self-calibration by extracting silhouettes and formulating the calibration of internal and external camera parameters as a Quadratic Eigenvalue Problem, using the estimated camera parameters to improve silhouette extraction. Despite the good results shown in (MICUSIK; PAJDLA, 2010),

tests were performed in controlled environments, and the initial silhouette extraction may fail in practical surveillance scenarios. Zhang and colleagues (ZHANG et al., 2013) presented an approach for camera calibration by also exploring vehicle motion in traffic scenarios. However, their method requires the camera height to be known a priori.

The work of Liu and colleagues (LIU; COLLINS; LIU, 2013) proposes a self-calibration method for multi-view scenarios. First, the approach tries to fit ellipses around foreground blobs to extract the principal orientation of each person in a scene. This serves to compute robustly the vertical vanishing point – the focal length is extracted by testing a set of hypothesis and minimizing an error function based on the blobs 3D heights. This process is performed for each camera first individually and then an optimization is performed to define a global world coordinate system. Also focusing in multi-view scenarios, the method of Guan et al (GUAN et al., 2016) proposes the reconstruction of head and feet positions in 3D w.r.t. a local camera coordinate system. The extrinsic parameters are extracted by solving a linear system of equation using least squares. The information from different cameras is then combined (pairwise) to generate the extrinsic calibration, which is further refined using an optimization via Gradient Descent aimed at minimizing the reprojection error over all parameters. Despite the simplicity of the method, the authors focus their experiments in more controlled scenarios.

More recently, Brouwers et al. (BROUWERS et al., 2016) proposed an approach that also uses head and feet locations (as illustrated in Figure 2.1(c)) to extract vanishing points and the horizon line. Instead of using foreground masks, they train Histograms of Oriented Gradients (HOG) detectors for feet and head localization which are performed separately. Then, head and feet of same people are matched by shifting the head detection downwards and computing an overlap with the possible feet detection: if the overlap is too small, the pair is rejected. The lines from feet to head are used to estimate the full calibration matrix based on the generic work of (ORGHIDAN et al., 2012), which uses vanishing points to calibrate cameras.

The method of Huang et al. (HUANG et al., 2016), similarly to previous works, also focus on extracting three vanishing points to calibrate the camera. Yet, the features used in their work consist the left and right foot locations of a person walking in a straight line. By combining these points in order to generate lines (similar to what is proposed in (LV; ZHAO; NEVATIA, 2006)), the method is able to extract the vanishing points and perform calibration. Their formulation also address the problem of when the left and right foot positions are co-linear, but it assumes that the foot are very prominent in the sequence such they can be individually located.

As it will be clear in the next chapters of this dissertation, we aim to use the calibration

at different stages of our pipeline. Since it is important to recover the real-world coordinates from tracker trajectories, we must get a full understanding of the camera position and orientation w.r.t. the ground plane. However, there are some methods that do not require a full calibration matrix, but instead are interested in understanding some of the 3D structure of the scene and/or camera viewpoint. One example is the method of Hoiem and collaborators (HOIEM; EFROS; HEBERT, 2008), which consists of an approach for estimating the viewpoint (horizon line and camera height) in a single image, as illustrated in Figure 2.1(b). For that purpose, they modeled the viewpoint parameters in a probabilistic manner, and explored the relationships of different objects detected in the scene (e.g. vehicles and pedestrians) to compute maximum likelihood estimates. Additionally, they also used viewpoint and geometry cues to improve object detection.

Chakraborty et al. (CHAKRABORTY; CHENG; JAVED, 2013) build on top of (HOIEM; EFROS; HEBERT, 2008) and attempt to recognize interactions between people in a single image using geometric information. Face detections are used as input and the goal of their calibration procedure is to estimate the ground plane coordinates of each detected face (taking the assumption that they lay on the ground plane). To achieve this goal, they first assume that all faces have the same height and look for outliers using RANSAC. These outliers are fed to an error correction system that updates the model to explain those faces and retrieve their ground plane disposition. Taylor and Mai (TAYLOR; MAI, 2013) proposed a method to estimate the pixels corresponding to the floor by examining the movement of targets in the scene. With a similar goal, the method of Fouhey et al. (FOUHEY et al., 2014) uses appearance and the detection of people actions to determine if a surface of the scene is one in which people sit or walk around.

2.2 Multi-object tracking

Object tracking is an active research topic in the computer vision community, and many approaches were proposed in the last decades. However, the problem is still open, particularly when tracking multiple simultaneous objects. The available literature on the subject is extensive, and this dissertation will provide an overview of some works focused on monocular 2D pedestrian tracking. The reader can refer to (YILMAZ; JAVED; SHAH, 2006; LEPETIT; FUA, 2005) for a comprehensive review and taxonomy of general 2D and 3D tracking algorithms, or the survey paper by Enzweiler and Gavrila (ENZWEILER; GAVRILA, 2009) for pedestrian detection and tracking using monocular cameras.

Tracking multiple pedestrians simultaneously using a single camera is a challenging

task. To robustly solve this problem, one has to account for appearance changes, images noise, partial or total occlusions, among others. One of the first proposed systems, the W4 algorithm (HARITAOGLU; HARWOOD; DAVIS, 1998), employed background segmentation combined with shape and texture information to perform real time tracking in gray-scale video sequences. Fleuret et al. (FLEURET et al., 2008) also explored background segmentation coupled with appearance models built using color histograms.

A challenging aspect of pedestrian tracking is to maintain a good localization during and after an occlusion. A simple way to deal with partial occlusions is to consider the target object as a set of patches. The rationale behind this idea is that if some patches are occluded and tracked incorrectly, the remaining patches can provide a good estimate of the pose. The FragTrack algorithm (ADAM; RIVLIN; SHIMSHONI, 2006) divides the target region into multiple image fragments at initialization. For each fragment, a vote map is constructed using image histograms. Then, these maps are combined in a robust way so that the influence of outliers is reduced. Dihl et al. (DIHL; JUNG; BINS, 2011) also use the same idea for object tracking, but track each patch independently and combine these tracking results to estimate the location of the target. The use of multiple fragments has shown good results in generic tracking applications (ADAM; RIVLIN; SHIMSHONI, 2006; DIHL; JUNG; BINS, 2011), and also when tailored to pedestrian tracking (FÜHR; JUNG, 2012; FÜHR; JUNG, 2014).

In recent years, a different class of approaches based on tracking-by-detection has gained significant attention, also because they are usually more robust than traditional methods in the presence of occlusions. These methods are based on the continuous application of a detection algorithm in individual frames, and then performing the association of detection results across frames.

Benfold and Reid (BENFOLD; REID, 2011) use Histograms of Oriented Gradients (HoGs) (DALAL; TRIGGS, 2005) and Kanade-Lucas-Tomasi (KLT) tracking to detect people and estimate their motion between detections. To obtain the final trajectories, a Markov-Chain Monte-Carlo data association is applied within a temporal window. Pirsiavash et al. (PIRSIAVASH; RAMANAN; FOWLKES, 2011) proposed a method that first detects all the pedestrians in the sequence and then uses dynamic programming to associate the detections into trajectories, as illustrated in Figure 2.2(b). Methods that performed data association globally (PIRSIAVASH; RAMANAN; FOWLKES, 2011) or within a sliding-window (FAGOT-BOUQUET et al., 2016; BENFOLD; REID, 2011) perform generally well, since looking at future frames can reduce uncertainty at current and past times. Yet, this comes with a price: the latency caused by the use of future observations in the estimation of the current state, i.e. they are not causal. Some new

approaches tried to tackle this inherent problem. The work of Choi (CHOI, 2015) aim to keep the benefits of a causal approach yet fixing past association errors. His work also introduces the use of optical flow association (within a temporal window) following the reasoning that such feature can provide a good cue in cases appearance fails – the motivation example is similar cars being tracked with different dynamics.

In the context of this dissertation, the non-causal characteristic is undesired because many surveillance applications require online tracking. Despite that, class-specific detectors have been proved to be very powerful and are continuing to increase in accuracy and performance in recent years (GEIGER; LENZ; URTASUN, 2012; ZHANG et al., 2016). Thus, it makes sense to use pedestrian detection as an additional cue within an online framework.

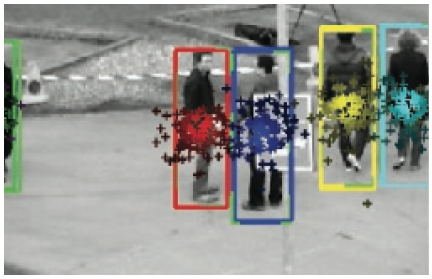
Breitenstein and colleagues (BREITENSTEIN et al., 2011) presented a multi-person online tracking algorithm in an incremental manner: they use class-specific information to detect pedestrians, and also target-specific information to discriminate each pedestrian. Data association across time is performed using a particle filter (Figure 2.2(a)), using position and velocity to build the state vector. The tracker proposed by Liu et al. (LIU et al., 2015) is also based on particle filters. Their main contribution is the addition of a model that simulates velocities and destination for a set of pedestrians.

With the goal of tracking generic objects, Kalal et al. (KALAL; MATAS; MIKOLAJCZYK, 2010; KALAL; MIKOLAJCZYK; MATAS, 2012) presented an approach that combines detection, learning and tracking. A tracker is used to follow the target in time, while the detector localizes all appearances that have been observed in the past and corrects the tracker if necessary. The learning phase estimates detection errors and updates the detector to avoid future mistakes.

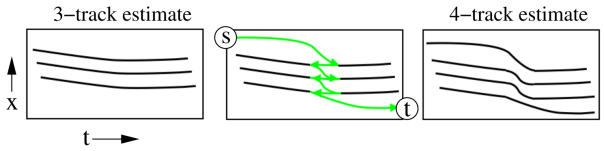
The knowledge of camera information is also useful in pedestrian tracking. Choi and colleagues (CHOI; SAVARESE, 2010) proposed a multi-target tracking model to identify the trajectories of multiple objects in 3D based on an initial estimate of the camera parameters. Such 3D trajectories are estimated by measuring their projections onto the 2D image plane, which represent the observation variables, and then jointly searching the most plausible explanation for both camera and all the existing target states in using the projection provided by the camera model. However, their method assumes that the camera is parallel to the ground plane, being more useful for mobile robots than surveillance systems that rely on static cameras.

More recently, some papers have been proposed to tackle crowded scenes using the concept of tracklets, i.e. small temporal adjacent associations of a target that are used to create longer and more stable trajectories. Bae and Yoon (BAE; YOON, 2014) proposed a method based on estimates of tracklet confidences and online appearance learning stage. The confidence

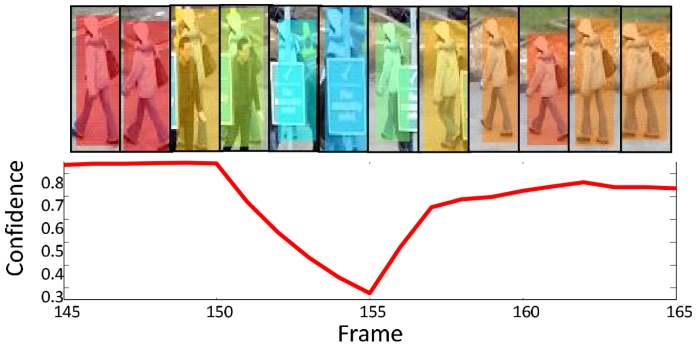
Figure 2.2 – Main components of difference state-of-the-art methods. (a) Breitenstein et al. (BREITENSTEIN et al., 2011) use particle filtering to track subjects. (b) Pirsiavash et al. (PIRSIAVASH; RAMANAN; FOWLKES, 2011) assemble tracklets into trajectories with a global optimization algorithm. (c) Bae and Yoon(BAE; YOON, 2014) rely on the confidence of tracklets to perform global or local associations.



(a) Source: (BREITENSTEIN et al., 2011)



(b) Source: (PIRSIAVASH; RAMANAN; FOWLKES, 2011)



(c) Source: (BAE; YOON, 2014)

of a tracklet depends on three different aspects: i) how much the subject appears occluded in the scene; ii) how well a nearby detection can be associated with a tracklet; and iii) the length of a tracklet. The effect of an occlusion to the tracklet confidence can be observed in Figure 2.2(c). The tracklets are associated based on its confidence: a global association is performed if a tracklet presents a low confidence value, while a local association (with a detection) is applied otherwise. Another method based on global optimization was proposed by Zhang et al. (ZHANG et al., 2015). Their method works by first associating temporal adjacent detections into tracklets using pairwise Markov Random Fields. The algorithm iterates between the optimization of trajectories and tracklets until the system converges. At each iteration, longer tracklets are created. Their approach show improvement upon existing non-causal approaches. Wang et al. (WANG et al., 2016) proposed the use of motion and appearance in order to learn a discriminative metric to associate tracklets. While appearance is modeled using color and shape information, motion dynamics similarity assumes that the targets do not change significantly their dynamics between tracklets. The method learns weights for each affinity model at tracking-time and refines tracklets using these data. Experimental results do not show significant improvement over the state-of-

the-art, yet the authors show that tracking improves when both appearance and motion are used, suggesting that both are important cues for tracking.

In analogy to what occurred to many other fields in Computer Vision, Convolutional Neural Networks (CNNs) based methods have appeared in the last few years for general object tracking. Bertinetto and colleagues (BERTINETTO et al., 2016) trained a fully convolutional network to learn a function of similarity that is used for tracking. At runtime, a set of search locations is fed to the network to predict the current 2D pose probability map of the target in the current image. The model used to describe the appearance of the target is not updated trough time and a search window is defined at different scales. Held et al. (HELD; THRUN; SAVARESE, 2016) proposed a similar architecture but use as input the current and previous frames cropped around the previous location. The output of the network is the estimated bounding box for the current frame. Since their network is quite simple, they achieve very fast performance. However, the system cannot handle well occlusions or fast movements.

Several recent methods propose the use of learning in multi-object tracking performed in both offline (XIANG; ALAHI; SAVARESE, 2015) and online fashions (BAE; YOON, 2014). The method of Xiang et al. (XIANG; ALAHI; SAVARESE, 2015) proposes the use of Markov Decision Processes (MDP) in a reinforcement learning scheme to make decision about the people trajectories. The MDP models the lifetime of an object and has the states of active, tracked, lost or inactive. The policy that will be learned in training will constitute the data association from detections to trajectories. Experiments show that despite heavily relying on these learned policies, the method is still able to perform reasonably well in cross dataset scenarios.

It is also worth mentioning here the work done in past years from the MOT Challenge group, most recently by Milan and colleagues (MILAN et al., 2016), who set a comprehensive benchmark for pedestrian tracking. This follows previous work of Bernardin et al. (BERNARDIN; STIEFELHAGEN, 2008), who proposed a metric protocol for multiple-object tracking that since became very popular (and it is used in this dissertation): the CLEAR MOT metrics. A very recent work that performed very well in the MOT Challenge is the method of Sadeghian et al. (SADEGHIAN ALEXANDRE ALAHI, 2017), which combines multiple cues to perform tracking: motion, appearance and interaction cues are combined inside a Recurrent Neural Network. As it seems to be a trend in the last couple of years, the appearance features are extracted using previously proposed CNNs. Unfortunately, many of the challenge videos are not from surveillance scenarios and some are captured by moving cameras in which calibration is not possible. Nonetheless, the sequences used for the tracking experiments of this dissertation (Towncentre e PETS S2.L1) are included in the MOT Challenge dataset.

2.3 Collective behavior recognition

In recent years, there has been increasing interest in inferring semantic information about the relation and interaction among people in a video sequence. There are different terms that are being used to describe the topics that are related to this fairly new area of research, such as group activity recognition, collective activity recognition and collective behavior recognition. In this dissertation, we chose the latter term because we are mainly interested in two aspects of the human behavior analysis: the activities that are being observed in the video and the social relation between pedestrians. It is also important to clarify that we focus our research on sparse surveillance scenarios, where people are individually identified, as opposed to crowd analysis. In the remaining of this section, we highlight some of the relevant works on collective behavior recognition. A more comprehensive study on human behavior recognition can be found in the survey of Borges et. al (BORGES; CONCI; CAVALLARO, 2013).

One of the first works about human interactions was proposed by Oliver et al. (OLIVER; ROSARIO; PENTLAND, 2000). They developed a framework that applies Kalman filter in order to track the objects' locations, shape, color and velocity. This information, together with the spatial relationship to nearby objects, is used to describe people's motion and interaction. The data is temporally arranged in streams that are used to obtain the collective behavior observed in the sequence. Extensions on the well know Hidden Markov Models (HMM) are presented in order to cope with multiple agents and state variables that interact with each other. Their algorithm is able to detect behaviors such as *meet and continue together*, *meet and split* and a person *following another*. Taj and Cavallaro (TAJ; CAVALLARO, 2010) also use HMMs to detect events and activities involving persons and objects, as well as groups of people (which they call in their paper as Interaction Event Recognition, IER). For the latter, features such as relative distances, direction and speed are used for training. The activities tackled in their work are people *approaching*, *meeting* and *walking together*.

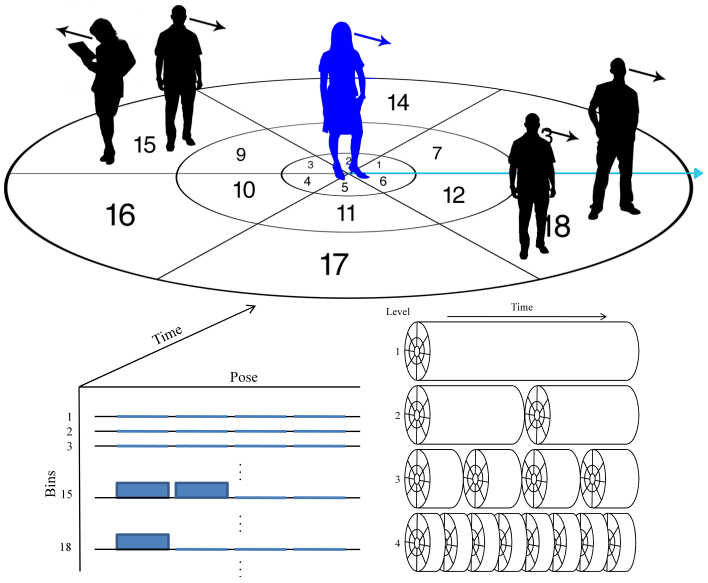
Ge et al. (GE; COLLINS; RUBACK, 2012) proposed a method to discover pedestrian groups in a video sequence. First, they combine pedestrian detector, particle filter for tracking and a data association scheme to merge people tracklets into trajectories. These trajectories are projected to the ground plane using an homography and an hierarchical clustering approach is used to identify and merge/split small groups of people. Their algorithm is inspired by the sociological work of McPhail and Wohlstein (MCPHAIL; WOHLSTEIN, 1982), which proposed an objective scheme to detect groups of pedestrians. This detection can be decomposed in three subsequent tests performed in groups of two people. More specifically, for a pair to be

considered a group the subjects should: 1) be closer than 2 meters and not separated by another subject; 2) have a difference in their speeds that is lower than 0.15 meters per second and finally 3) have a difference in direction smaller than 3 degrees. Feng and Banhu (FENG; BHANU, 2015) proposed a method to identify groups and their interactions using what they called an evolving tracklet interaction network (ETIN). Tracklets are considered as nodes in a graph and the edges represent the relation between different persons in the scene. The weight of these relations is measured by the weighted sum of aggregated positional, velocity and directional distances. Social groups are identified by maximizing the modularity of the network created using the tracklets that appear in a given period of time (snapshot).

Choi et al. (CHOI; SHAHID; SAVARESE, 2009) proposed a method for collective activity classification that first detects the people in the scene and extract their poses using a Support Vector Machine (SVM) classifier that defines the human poses as front, left, right and back with respect to their orientation from the camera. The subjects are tracked using Kalman filters and their trajectories are represented by descriptors which the authors called Spatial-Temporal Local (STL) descriptors. Figure 2.3 shows an illustration of this kind of feature. These are simply histograms of the number and orientation of people within a radius area around a specific subject along time. These features are used for classification — once again, an SVM classifier is used for this task. The method is able to detect events such as people queuing in line, talking and crossing a street. Extracting the subjects direction can be helpful to understand the role of an individual in an activity – for instance, it is possible to differentiate waiting to cross a street or queuing if the people still directions are extracted. However, their approach to obtain these orientations is highly dependent on the camera views used in the training and test sets. An extension of the STL descriptor was presented by Chang and colleagues (CHANG; ZHENG; ZHANG, 2015), which use motion features together with the STL descriptor to learn pairwise relations for different collective activities.

The work presented in (CHU et al., 2012) uses only the information based on trajectories to classify group activities. They introduce the idea of heat-map features in which a trajectory is modeled as a series of heat sources laying on a grid of non-overlapping patches that represents the scene. To avoid the loss of temporal information, a decay function is used in such a way that the beginning of the trajectory has less thermal energy than the current position. Also, to account for noise in the trajectories, a thermal diffusion is applied around the initial heat sources. An important problem with such a direct map from the trajectories is the same activity can lead to different heat maps because of the variety of angles and lengths in people's path. To account for that, a key-point on the peaks of such heat maps is extracted and used to align the features.

Figure 2.3 – Spatial temporal local (STL) descriptor.



Source: (CHOI; SHAHID; SAVARESE, 2009)

Finally, the classification is performed by surface fitting of a previously trained and a current heat map surface. Their method, yet simple, is able to detect activities like *turn*, *follow*, *overtake*, etc.

Also based on trajectories alone, the work of Huis et al. (HUIS et al., 2014) attempts to recognize events such as *pickpocketing* based on the sequence of actions such as *walk*, *meet*, *split*, *loiter*, etc. Features such as speed, distance, direction and angles between tracks are used to detect the actions according to predefined rules – e.g. a person is considered to be loitering if his/her trajectory has a speed smaller than 3km/h for longer than 4 seconds. Once the events related to pickpocketing are identified, an alert can be raised in Closed-Circuit TeleVision (CCTV) systems. Bouma et al. (BOUMA et al., 2014) extended this work by employing more flexible features, such as the number of nearby persons, speed and distance after a split and orientation changes. These serve as input to a classifier (Fisher linear discriminant classifier) that can recognize pickpocketing situations in a real scenario.

Cheng et al. (CHENG et al., 2014) represented the problem of group activity recognition using a three-layered approach that gathers information about the individuals performing the actions, the possible pairs between two people and small groups. The motion features that are extracted are also defined in these layers. First, each trajectory is modeled using Gaussian processes. Additional information, such as location change since the beginning of the sequence, the average velocity and the velocity ratio are also added as features. Context information is extracted by comparing both the location and velocity of an individual with relation to the others. They also propose the use of the participants shape to describe an activity. These so-called action

style features are Histograms of Oriented Gradients (HOG) of the people in the group, which is shown to discriminate correctly between different actions performed by individuals such as *standing* and *fighting*. Similar to the work of Jacques et al. (JR et al., 2007), a geometric shape is used to analyze the group formation. In order to do this, the Delaunay triangulation algorithm is applied to the polygon that connects the people in the scene. Finally, these descriptors are used for training and classification. First, the descriptors are clustered using K-means to generate visual words and then SVM is used to classify samples.

The work of (LI; CHELLAPPA; ZHOU, 2013) proposed a descriptor strategy for motion information of group activities that is compact and discriminative. First, the method computes a tensor that relates each subject being tracked with the others at all the time steps. To fill this tensor, the relation between two subjects at two possibly distinct times can be computed as the Euclidean distance between their two centers of mass or the inner product of the velocity vectors – indeed, the authors show in the paper how to fuse different tensors computed using different features. A tensor reduction is used to compute a more compact and view-stable features. The final descriptors are used for classification through a probabilistic framework that is described in the manifold of these reduced features.

More recently, there is a number of approaches modeling the problem of group activity using recurrent neural networks (RNN). The approach of Deng et al. (DENG et al., 2016) explores two main ideas. First, that the context of other people in the scene can remove ambiguity in the inference of an individual action. Second, that recognizing which subjects are interacting (or not) with each other is an important property to perform collective activity recognition. Following this, a Convolutional Neural Network (CNN) whose input is the frame window of a person detection is used to classify individual actions. In addition, the whole frame is given as input to a different CNN that tries to classify the group activity from a single image. A graphical model and an RNN is used on top of these classifiers to refine these results based on relations among entities.

IBrahim et al (IBRAHIM et al., 2016) proposed a 2-layer hierarchical model based on long short-term memory models (LSTM) – a type of RNN architecture. In the first stage, an LSTM is used to represent temporally the action of each person. Again, a CNN is used to extract features inside each bounding box, which will serve as input to the model. The second layer is responsible for modeling the temporal dynamics of the group activity. It achieves that purpose by analyzing the individual actions and temporal changes of person actions as a whole – a pooling layer is used to aggregate relevant information from the first level. The presented results did not achieve state-of-the-art performance for small datasets, yet the method seems promising in larger

ones, such as the volleyball dataset proposed by the authors themselves.

Despite the fact that RNNs can accommodate complex inference models, is not entirely clear at this point how well they generalize across different scenarios and if the training set needs to be as large as in many other RNN applications.

2.4 Conclusion of the chapter

As exposed in the Introduction (Section 1), our ultimate goal is to understand what is happening in the scene observed by a single camera and how people are interacting. The methods covered in this chapter present different challenges that arise in a surveillance environment. One critical aspect of it is that tracking (and subsequently, collective behavior analysis) should present small latency times. Methods based on global optimization along the temporal axis could be impractical in many real applications if they rely heavily on future frames. For example, in case of detecting events such as pickpocketing or loitering, it is important that the system alert the supervisor as soon as possible so he/she can act on them.

Another important aspect that is not addressed extensively in the literature is the loss of generality when trajectories are recovered in the image plane rather than the world. This becomes evident when analyzing people position and orientation to determine which activities are being observed by a camera. For some methods, once the trajectory is recovered in image coordinates, it is difficult to extend this information to complete different scenarios due to the variety of camera setups encountered in the wild. Therefore, the use of actual distances (in the world coordinate system) is useful for behavior analysis. Moreover, because some relevant activities are rather non-usual, the training should be able to build a model that is discriminative using only a small number of training samples.

Due to this challenges, we propose the use of calibration in the tracking stage, which can improve tracking results and also provide discriminative features for event recognition based on world coordinates. To alleviate the limitation of using calibrated cameras, we propose an automatic self-calibration scheme to be applied prior to tracking. Additionally, our tracking method is causal such as can be applied in online scenarios with near real-time performance. In the next chapter, we cover in details our approaches to self-calibration, tracking and collective behavior.

3 PROPOSED METHODS

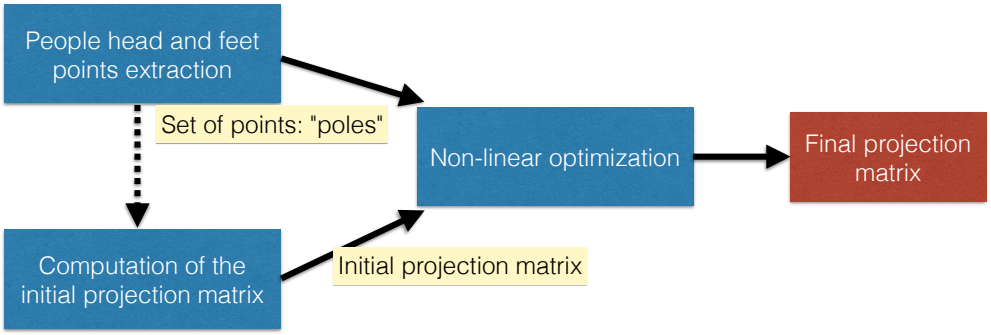
In this dissertation, instead of approaching each isolated problem, we devise contributions in three main areas that are inter-related: camera self-calibration, pedestrian tracking and collective behavior recognition. These steps have the ultimate goal of extracting interactions and group behaviors from the set of people observed by a static surveillance camera. Additionally, one strong contribution of the dissertation is to show how calibration can be used in the typical video surveillance pipeline, enabling the methods of pedestrian tracking and people detectors to be more accurate and faster. Also important is the proposed use of calibration to extract relevant inter-personal distances for obtaining interactions (and subsequently, group events) based on known psycho-social distances, being also more suited to a wider range of camera setups.

We first propose a fully automatic self-calibration method that only uses information extracted from people in the scene to calibrate static cameras. The method is described in 3.1, together with applications in pedestrian detection, tracking and scene geometry understanding. The multi-pedestrian tracker is described in Section 3.2. Our approach relies on multiple image cues such as patch-based matching, motion prediction and association of pedestrian detection. We combine the ensemble of displacement estimates for each target in the WCS using the camera projection matrix. Also using calibration, we discuss how to simplify scale estimation and reduce the target search region between two adjacent frames. Finally, we describe our proposal for collective behavior recognition in Section 3.3. The method is composed of two classifiers in sequence: the first is responsible for estimating non-symmetrical interactions between pairs of pedestrians, while the second aims at classifying group activities appearing in the sequence. Our lightweight interaction descriptors are built by analyzing the distances between a pair of targets inside a temporal window through histograms, using bins that are defined according to psychosocial work by Hall (HALL, 1973). Our collective behavior descriptor consists of a histograms of pair-wise interactions, a factor representing the group shape dynamics and its mean velocity.

3.1 Self-calibration

Instead of using artificial patterns to calibrate cameras, self-calibration aims to use natural structures from the scene to infer camera parameters. The seminal work of Lv et al. (LV; ZHAO; NEVATIA, 2002a) in this area proposed the use of pedestrians to calibrate a static camera. In their work, different detections of the same person are assumed to be vertical poles that are parallel in

Figure 3.1 – Overview of the self-calibration technique proposed in this work



Source: Author

the WCS. By analyzing the vanishing points and horizon line extracted from combining these poles projections, it is possible to estimate the camera projection matrix. Our method is based upon this idea of using poles to calibrate a static surveillance camera, but further refined to reduce the error of re-projected poles. Our approach is composed of three stages, as illustrated in Figure 3.1.

We assume in this dissertation that multiple people can be fully observed (from feet to head) from a single static camera and that the ground is in fact planar, which is reasonable for most surveillance scenarios. Additionally, we differ from methods such as such as (HUANG et al., 2016; GUAN et al., 2016), in which the problem of co-linearity of feet and head points (in 3D) is explicitly addressed. In surveillance scenarios, where multiple pedestrians are typically present, that the assumption that the points will not be co-linear should not be a major concern. In the proposed formulation, we also assume that the heights of people observed in the scene averages to an specific value. However, this value can be changed in order to accommodate unusual distributions. Finally, we assume the classical pinhole camera model without radial distortion, so that the final goal is to estimate the 3×4 projection matrix P.

The first stage of the proposed algorithm, described in Section 3.1.1, estimates the line segment along the body of each detected pedestrian connecting the head to the feet from a given set of frames. In the remainder of the text, we refer these segments as *people poles*, which represent the height and the orientation of each detected pedestrian. The second step uses the endpoints of these extracted lines, and it is responsible for computing a rough approximation of the camera matrix using a modification of (LV; ZHAO; NEVATIA, 2002b)[1], as described in Section 3.1.2. The final phase consists in a non-linear optimization that aims to minimize the re-projection error of the poles in the image. This optimization can greatly increase the quality

[1]Please notice that other methods can provide a first calibration estimate for our non-linear optimization phase and thus can be applied in our pipeline

of the self-calibration, as demonstrated in the set of experiments described in Section 4.1.

3.1.1 Extracting people poles from a set of images

In order to extract the people poles, pedestrian detection and background segmentation are applied to each frame. More specifically, the pedestrian detector proposed in (DOLLÁR; BELONGIE; PERONA, 2010; DOLLÁR et al., 2009) is applied to each frame[2] and, for each resulting bounding box, the foreground within this region is extracted using a background segmentation approach (BARNICH; DROOGENBROECK, 2011). The foreground is first used to eliminate false positives occurred during the detection phase by computing the ratio between the number of foreground pixels and the total number of pixels inside the bounding boxes – the detection is rejected if this ratio is below a given threshold t_a, which possibly indicates a false positive detection.

To compute the orientation of the poles, we propose a technique similar to the method used by Lv et al. (LV; ZHAO; NEVATIA, 2002b). The coordinates of all foreground pixels inside a detection bounding box is fed to the Principal Component Analysis (PCA) procedure, which returns the two main axes of variance of the points. The major axis of variance is assumed to be the vertical orientation of the person inside the bounding box. To detect the feet and head points, we initially compute the line inside the bounding box that passes through the centroid of the foreground points and has orientation given by the major axis. Along this line, the first and last foreground points are extracted and assumed to represent the feet and head endpoints, respectively.

The ratio between the highest and the lowest eigenvalue provided by the PCA is also computed for each pole. This value is the ratio of variances along the minor and major axes of the underlying ellipse characterized by the covariance matrix. If this value is above a threshold q_s the pole is rejected because it may represent that the pedestrian has its legs apart or the background segmentation was not well computed in that region. This in turn can result in noisy pole estimates, which could compromise the calibration procedure. The pole extraction procedure is illustrated in Figure 3.2.

This procedure also reduces the number of extracted poles by keeping what we expect to be the best ones. However, we use a simple threshold and do not compute pedestrian walking cycles to estimate when the legs are the closest to each other, as in (LV; ZHAO; NEVATIA, 2002b). The reasoning behind this was that the original method of Lv et al. (LV; ZHAO; NEVATIA,

[2]Other detectors could be used instead. See (ENZWEILER; GAVRILA, 2009) for a comprehensive list.

2002b) was intended to calibrate a camera that only observes a single pedestrian throughout a sequence of frames, which may be a limitation for real scenarios (such as surveillance). Our approach does not present such constraint and, furthermore, does not require a set of temporally adjacent frames as (LV; ZHAO; NEVATIA, 2002b), used to extract walking cycles. Instead, we use the poles related to different pedestrians (and possibly obtained at sparsely sampled frames in time) to obtain an initial estimate of the projection matrix, which is refined later.

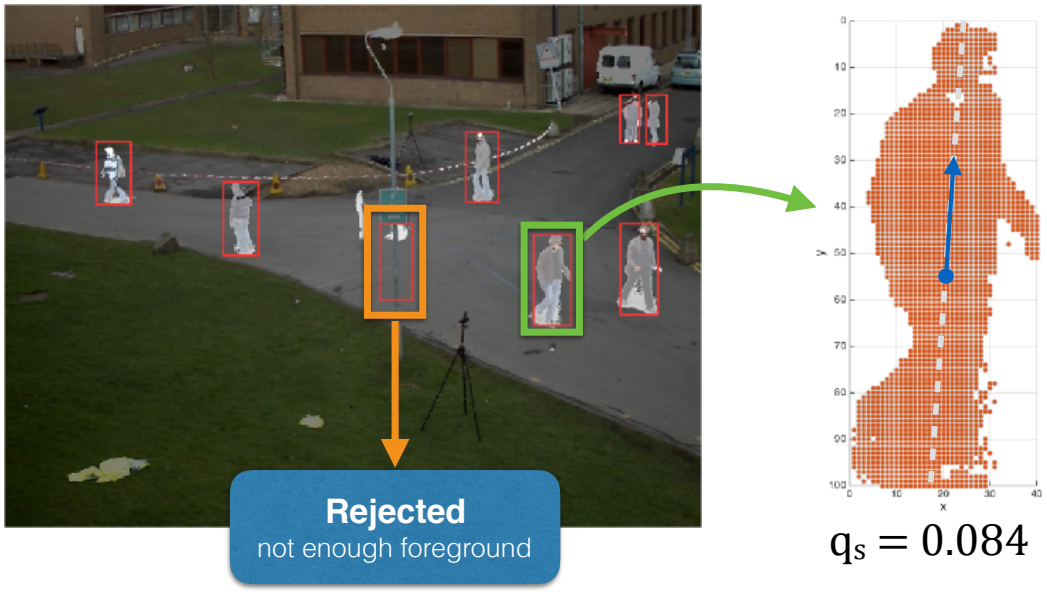
3.1.2 Projection matrix initialization

As we present in Section 3.1.3, our non-linear optimization has two requirements as input: the set of poles extracted in the image and an initial projection matrix. The former was addressed in the previous section and the later can be provided by different (linear or non-linear) self-calibration methods (LV; ZHAO; NEVATIA, 2002b; LV; ZHAO; NEVATIA, 2006; JUNEJO; FOROOSH, 2006). We propose a method based on the extraction of the vertical vanishing point and horizon line of the image that follows the line of (LV; ZHAO; NEVATIA, 2002b).

There are different methods for extracting the horizon line of an image (LV; ZHAO; NEVATIA, 2002b; HOIEM; EFROS; HEBERT, 2008). One approach is to create, for each pair of poles, two lines connecting the head and feet points. Then, the intersection point of these two lines should lie in the horizon line, as previously shown in Figure 2.1(a). In our algorithm, each pole is paired to all others poles that lie in a distance greater than a threshold d_p, since nearby poles tend to generate noisy vanishing points. Each pair of poles contributes to a point, and RANSAC (FISCHLER; BOLLES, 1981) is used to fit the line that corresponds to the horizon. It is important to point out that the theoretical background of the method requires that all people have the same height (LV; ZHAO; NEVATIA, 2002b; LV; ZHAO; NEVATIA, 2006), which can be imposed by using only the poles related to a single pedestrian. In fact, we carried out experiments using a robust pedestrian tracker (BREITENSTEIN et al., 2011) to link poles related to the same person, and the results were not better than using the poles of all people at the same time. The reason is that the errors in estimating the pole orientation and endpoints are larger than the actual difference of different people heights, so that we decided not to employ a tracker and assumed an average height.

The vertical vanishing point encodes how the orientation of pedestrians changes for different locations of the scene. This point corresponds to the intersection of the lines related to the poles of any pair of pedestrians. As described for the horizon line procedure, we choose all possible pairs of poles whose distances are greater than d_p. Outlier points can appear if a pair of

Figure 3.2 – The extraction of people poles. In (a), a detection is rejected due to insufficient foreground and, for a different pedestrian, the major axis is obtained trough PCA computed from the mask pixels locations. (b) shows the poles extracted lay on top of the original image.



(a)

(b)

Source: Author

poles is nearly parallel[3] or if the orientation was poorly estimated due to a number of factors, such as occlusions or noisy background segmentation. Therefore, the x and y coordinates of the final vanishing point is taken to be the median value in each dimensions of all intersection

[3] Actually, depending on the measure applied, one can think that all lines are nearly parallel. However, when computing vanishing points from it, a cluster near the truth vanishing point will appear, yet some vanishing points will be far from it due to poor pole extractions.

points. Once the vanishing point and the horizon line are extracted, it is possible to extract a projection matrix $\tilde{P}$ if a given height is assumed for the set of pedestrians observed. See (LV; ZHAO; NEVATIA, 2002b) for more details on how to accomplish this task.

3.1.3 Non-linear optimization

Since the computation of the initial matrix $\tilde{P}$ is based on the vertical vanishing point and horizon line (as described in the previous subsection), a small error in the extraction of people orientations can lead to rather large errors in the projection matrix. In particular, lower resolution video sequences are more error-prone, since they tend to present larger errors in the pole estimation due to the small size of pedestrians. Indeed, we observed in our experiments that our initial method, while it generally provides a good ground plane calibration, it fails to correctly estimate the vertical axis Z (see Fig. 3.4(a)). Therefore, given $\tilde{P} = \begin{bmatrix} \tilde{\boldsymbol{p}}_1 & \tilde{\boldsymbol{p}}_2 & \tilde{\boldsymbol{p}}_3 & \tilde{\boldsymbol{p}}_4 \end{bmatrix}$, we want to find a refined camera matrix $P = \begin{bmatrix} \tilde{\boldsymbol{p}}_1 & \tilde{\boldsymbol{p}}_2 & \boldsymbol{p}_3 & \tilde{\boldsymbol{p}}_4 \end{bmatrix}$ that inherits the good ground plane homography achieved by $\tilde{P}$, but improves the projection of vertical poles. For that purpose, we propose to obtain the 3×1 vector $\boldsymbol{p}_3$ by minimizing a distance function between the people poles predicted using the camera projection and the poles extracted from the image.

Each extracted pole p_e^i is characterized by its two 2D endpoints: the feet region point $\boldsymbol{f}_e^i$ and the head point $\boldsymbol{h}_e^i$. We assume that all the extracted feet points $\boldsymbol{f}_e^i$ lie on the ground plane, i.e. $Z = 0$. Assuming also that pedestrians are standing (as in most pedestrian detectors), the projection of a vertical pole with a given height (e.g. the average height Z_{avg} of a person) at the location of detected pedestrians should approximately coincide with the corresponding extracted pole p_e^i.

To perform this projection, the feet point $\boldsymbol{f}_p^i$ of the predicted pole is set to be the same point as the extracted one, i.e. $\boldsymbol{f}_p^i = \boldsymbol{f}_e^i$. The projected head point $\boldsymbol{h}_p^i$ is obtained by mapping $\boldsymbol{f}_p^i$ to world coordinates using the ground plane homography matrix $H = \begin{bmatrix} \tilde{\boldsymbol{p}}_1 & \tilde{\boldsymbol{p}}_2 & \tilde{\boldsymbol{p}}_4 \end{bmatrix}$, adding Z_{avg} to the height component and then projecting back to the image using P, leading to

$$w\hat{\boldsymbol{h}}_p^i = \hat{\boldsymbol{f}}_p^i + Z_{avg}\boldsymbol{i}\boldsymbol{h}_3^T\hat{\boldsymbol{f}}_p^i\boldsymbol{p}_3, \tag{3.1}$$

where $\hat{\boldsymbol{u}}$ denotes the homogeneous coordinates of a 2D vector $\boldsymbol{u}$, w is the scale factor, and $\boldsymbol{i}\boldsymbol{h}_3^T$ is the third row of H^{-1}.

Clearly, one key issue is the definition of the distance measure between predicted and extracted pole that will guide the optimization problem. In several applications, it is important

to keep coherence in the projection of the Z axis. For instance, some pedestrian trackers (e.g. (FÜHR; JUNG, 2014)) explore the constant height of a person to improve their results; also, an accurate projection of the pedestrian height can be used to reduce the search space and discard false positive in pedestrian detectors based on sliding windows, as we show latter in this dissertation. Furthermore, accuracy in the Z orientation is also crucial for augmented/mixed reality applications.

The proposed error measure for a given extracted pole p_e^j with respect to its projected counterpart pole p_p^j obtained with projection matrix P is given by

$$C_j(P) = \alpha d_a\left(p_e^j, p_p^j\right) + (1-\alpha)\left\|\boldsymbol{h}_p^j - \boldsymbol{h}_e^j\right\|, \tag{3.2}$$

where d_a is a function that computes the angular difference (in radians) between two poles, given by

$$d_a\left(p_e, p_p\right) = \arccos\left(\frac{\boldsymbol{q_e}\cdot\boldsymbol{q_p}}{\|\boldsymbol{q_p}\|\|\boldsymbol{q_p}\|}\right), \tag{3.3}$$

where $\boldsymbol{q_e}$ and $\boldsymbol{q_p}$ are the vectors representing the poles p_e and p_q (respectively) centered at the origin $[0, 0]$. The parameter $\alpha \in [0,\ 1]$ in Eq. (3.2) is the balancing weight used to control the influence of each term in the optimization process. It is important to remember that the two errors are given in different units – one is a distance in pixels and the other is an angle in radians. Depending on the application that uses the calibration a different α should be chosen. For instance, if the low angular errors are preferred, larger values of α should be used. Conversely, smaller values should be used if the pixel-wise euclidean distance is to be minimized.

The total cost function $C(P)$ is then defined as

$$C(P) = C(\boldsymbol{p}_3) = \sum_{p_e^j \in S_{in}} C_j(P), \tag{3.4}$$

which is the summation of individual pole errors within a subset S_{in} of the poles extracted in the first stage. More precisely, if $S = \{p_e^1, p_e^2, \ldots p_e^n\}$ denotes the set of the n originally extracted poles, then $S_{in} \subset S$ is the subset composed by all the inliers with respect to the cost function $C_j(\tilde{P})$, where $\tilde{P}$ is the initial estimate of the projection matrix as described in Section 3.1.2. To obtain these inliers, the third quartile of C_j for all poles, denoted by $Q3$, is extracted along with the interquartile range, IQR. Then, a given pole p_e^j is considered an inlier (i.e. is assigned to set S_{in}) if $C_j(\tilde{P}) < Q3 + 1.5IQR$ (HAN; KAMBER, 2001).

Another issue related to the proposed cost function is that the re-projection error tends to be smaller in image regions containing a higher density of pedestrians, since $C(P)$ is the

summation of individual pedestrian re-projection errors (and denser regions implicitly carry more weight). Although one could introduce a weighing factor to penalize the influence of denser regions, we decided for a sub-sampling approach that aims to balance the density of poles along the ground plane, and also reduces the computational cost (since the number of calculations in $C(P)$ is reduced).

For the selection of samples, we divide the ground plane in squares of equal size and project the corresponding grid onto the image using the homography matrix H. Within projected squares containing at least one pole, we compute the average number of poles n_p per square. Then, in squares for which the number of poles is above n_p, we randomly select a subset of n_p poles, also removing poles that lie sufficiently close to an existing pole (a minimum distance threshold of 5cm was imposed). An example of our sampling procedure is shown in Fig. 3.3. As it can be observed, the procedure provides a more uniform distribution of the poles in the scene (right).

Figure 3.3 – Sampling of the person poles using the ground plane homography. Left: original set of poles. Right: sampled set of poles.

Source: Author

It is also important to point out that our cost function $C(P) = C(\boldsymbol{p}_3)$ given by Eq. (3.4) involves a 3-DOF variable $\boldsymbol{p}_3$, since the optimized projection matrix P inherits the ground plane homography from the initial estimate $\tilde{P}$. For the sake of comparison, we also performed a full optimization of P (i.e., all the 11-DOF elements of P) using $\tilde{P}$ just as an initial approximation. However, we noticed in our experiments that solving for the full projection matrix P often results in very poor calibration matrices, probably due to local minima of the cost function. The resulting projection matrix, while it produces small errors according to Eq. (3.4), tends to corrupt the ground plane homography. Fig. 3.4 shows the ground plane grid and the projection of vertical poles for the initial calibration matrix $\tilde{P}$, and the refined matrices P' (optimizing all elements) and P (optimizing only the third row, which is the proposed approach). As it can be observed, $\tilde{P}$

does a good job at the ground plane, but the quality of the orientation degrades (particularly for the poles on the left). Matrix P' produces projected poles very similar to the estimated ones, but ground plane homography was completely degraded. Finally, the projection matrix P optimizing only $\boldsymbol{p}_3$ mantains the good ground plane homography of $\tilde{P}$, at the same time reducing height and orientation errors of the projected poles.

Figure 3.4 – Non-linear self-calibration. Red poles were predicted by the projection matrix, and the blue ones were extracted from the video sequence. (a) The calibration resulted from the first stage ($\tilde{P}$) . (b) Non-linear optimization in all the projection matrix elements (P') (c) Non-linear optimization in the third column of the matrix (P).

(a) (b) (c)

Source: Author

Finally, different non-linear optimization methods can be used to produce the final projection matrix. Indeed, two methods, namely the Simplex method (LAGARIAS et al., 1998) and the Levenberg-Marquardt-Fletcher (FLETCHER, 1971), were employed with successful results. See the experiments (Section 4.1) for details.

3.1.4 Applications for calibrated cameras

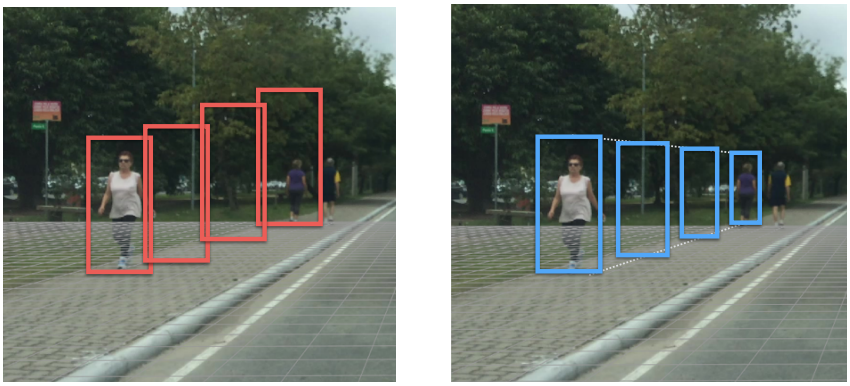
In this section, we present how the proposed self-calibration scheme can improve pedestrian detection within current state-of-the-art frameworks. We also discuss a simple yet useful application of simulating the placement of cameras in a surveillance scenario. It is worth noticing that our multi-target tracker also benefits from calibrated camera, but the method is presented separately in Section 3.2.

3.1.4.1 Improving pedestrian detectors

A common approach for pedestrian detection based on sliding windows is to classify image patches at several different scales, which is typically done using pyramids of images, pyramids of classifiers, or a combination of both (DOLLÁR; BELONGIE; PERONA, 2010).

However, there is only a small range of scales that relate to the dimensions of a pedestrian for a given pixel location, as illustrated in Fig. 3.5. More precisely, Fig. 3.5(a) shows a fixed-size scanning window, which is plausible for the nearest woman in the picture, but too large at the other locations illustrated. Fig. 3.5(b) shows geometrically-aware windows, for which the size depends on the ground-plane location related to the camera and a fixed pedestrian height.

Figure 3.5 – Example of detection windows with (a) fixed size, which lead to implausible pedestrian heights at some points, and (b) adaptive size, depending on the ground plane location.



(a) (b)

Source: Author

Using camera calibration to improve detectors has been tackled before in the literature. Notably by Hoiem and colleagues (HOIEM; EFROS; HEBERT, 2008) which explored a simplified camera model (knowledge of the horizon line) and local object geometry to improve the performance of object detectors. In this work, we rely on a better camera model (full calibration) and drop the local geometry constraint.

For a given bounding box B, let E_B denote some kind of pedestrian image-based evidence computed on B (e.g HOG or Haar-like features), and let Z_B denote its height in the WCS computed using the known camera parameters, assuming that the base of the bounding box is on the ground plane[4]. Following a Bayesian classifier, a pedestrian is detected when

$$P(ped)p(E_B, Z_B|ped) > P(\neg ped)p(E_B, Z_B|\neg ped), \tag{3.5}$$

where $p(E_B, Z_B|ped)$ and $p(E_B, Z_B|\neg ped)$ are the joint PDFs of E_B, Z_B for the pedestrian and non-pedestrian classes, and $P(ped)$ and $P(\neg ped)$ are the corresponding *a priori* probabilities. Assuming that Z_B and E_B are independent and that $p(Z_B|\neg ped)$ follows a uniform distribution,

[4]Notice that this assumption is different than saying that the pedestrian feet lay on the ground plane. Moreover, some detectors were trained with significant padding around people. However, such padding can be easily compensated by growing the bounding box in the image plane by a fixed factor.

inequation (3.5) reduces to

$$\frac{p(E_B|ped)}{p(E_B|\neg ped)} p(Z_B|ped) > T, \tag{3.6}$$

where T is a fixed threshold.

Considering that the score $R(E_B)$ of any "baseline" pedestrian detector can be used to approximate the likelihood ratio $p(E_B|ped)/p(E_B|\neg ped)$ (disregarding normalization issues) and that $p(Z_B|ped)$ follows a normal distribution with mean Z_{avg} and variance σ^2, the proposed detector is given by

$$S(B) = R(E_B) \exp\left[-\frac{(Z_B - Z_{avg})^2}{2\sigma^2}\right] > T_S, \tag{3.7}$$

where the acceptance threshold T_S is inherited from the baseline detector $R(E_B)$. Due to the fast decay of the normal distribution, just a few bounding boxes B with WCS heights in the range $[Z_{avg} - k\sigma, Z_{avg} + k\sigma]$ are needed in practice for each location.

For detection methods that rely on image pyramids, a classifier is trained with a pre-defined pedestrian model size, typically a rectangular region with height z_{model}. In traditional sliding-window methods, the model is kept constant and the image is re-scaled to capture pedestrians at different scales: upsampling is required to detect pedestrians smaller than the model, and downsampling for pedestrians larger than the model. In general, just downsampling is applied, so that the smallest detectable pedestrian in the scene is roughly the height of z_{model}. Given a maximum pedestrian height $Z_{max} = Z_{avg} + k\sigma$ (in the WCS), our method only creates candidates in which the height of the corresponding bounding box height is larger than a fraction of the height of the model bounding box z_{model}. This fraction depends on the height range of pedestrians in WCS and its value can be employed in order to limit the number of levels of the pyramid.

The largest pedestrian in the image should dictate the smallest resolution of the image pyramid. Since the pyramid is pre-computed in some methods to speed-up the process (as in (DOLLÁR; BELONGIE; PERONA, 2010)), the use of a calibrated camera can also define the smallest scale of the pyramid. Given a pedestrian with size z_{ped} (in the ICS), the ideal scale s in the pyramid should satisfy $2^{-s} z_{model} = z_{ped}$ (assuming that the scale factor is 2^{-s}). Hence, we scan all image pixels related to the ground plane and compute the projection of a pedestrian with the maximum allowed size Z_{max}, retrieving the height of the largest bounding box in the ICS, called z_{max}. Hence, the smallest scale in the pyramid is defined as $s = \log_2(z_{model}/z_{max})$.

Finally, our candidates are evaluated in this reduced pyramid using the test given by Inequation (3.7). Furthermore, as usual in sliding-window techniques, the final detection is achieved after performing non-maxima suppression to the outputs of $S(B)$.

To show the potential of our method, we used as the baseline pedestrian detector the method presented in (DOLLÁR; BELONGIE; PERONA, 2010), and our experimental results show that detection accuracy were increased using camera information. We also present results modifying the classical HOG+SVM detector by Dalal and Triggs (DALAL; TRIGGS, 2005). It is important to know that our idea to generate candidates could also be applied to different detection approaches. Even modern CNN detectors based on region proposals could profit from a geometry-specific creation of candidates. A clear example would be to use our candidate generation algorithm instead of selective search (UIJLINGS et al., 2013) in the pipeline of the R-CNN (GIRSHICK et al., 2016) detector – because we only analyze the geometry and not the image itself to generate bounding box proposals, our method should be much faster. Besides, other calibration methods than the one described in the previous sections could be used instead. In fact, we coupled our detection strategy with a different method of self-calibration, designed for use in on-board vehicular cameras (PAULA; JUNG; SILVEIRA, 2014; FÜHR; JUNG; PAULA, 2016) that explores the visible lane geometry.

3.1.4.2 Using calibration to place virtual cameras

As we saw in the previous section, coupling calibration with detection can greatly simplified this problem. Besides that, there are other types of information that can be directly inferred from the calibration which might be useful for some applications. We explore a small use case of simulating the placement of cameras in surveillance scenario. One key aspect when planning a surveillance system is to define the number and location of cameras to cover interesting regions of the environment. We propose a tool to accomplish that based on augmented reality. Given a calibrated camera that provides an overall view of the scene (e.g. a wide field-of-view passive camera (QURESHI; TERZOPOULOS, 2007)), we developed an application that allows the user to place a virtual camera (with known intrinsic parameters, so that different cameras can be emulated) at different locations of the scene with a few mouse clicks, and then preview the feed captured by the camera at different orientations, emulating a pant-tilt-zoom (PTZ) camera.

In the proposed system, the user initially clicks on a ground-plane pixel and then defines the camera height Z (in the WCS). The camera location in the ICS is obtained similarly to the poles extraction used in the calibration step (see Eq. (3.1)), generating a “virtual pole” in the real scene, on top of which the virtual camera will be placed.

Given the location $\boldsymbol{X}_0 = (X, Y, Z)$ of the virtual camera, its resolution and intrinsic parameters encoded in a 3×3 matrix K, as well as the rotation matrix R (computed from the pitch, roll and yaw angles defined by the user), the projection matrix of the virtual camera is given by (HARTLEY; ZISSERMAN, 2000):

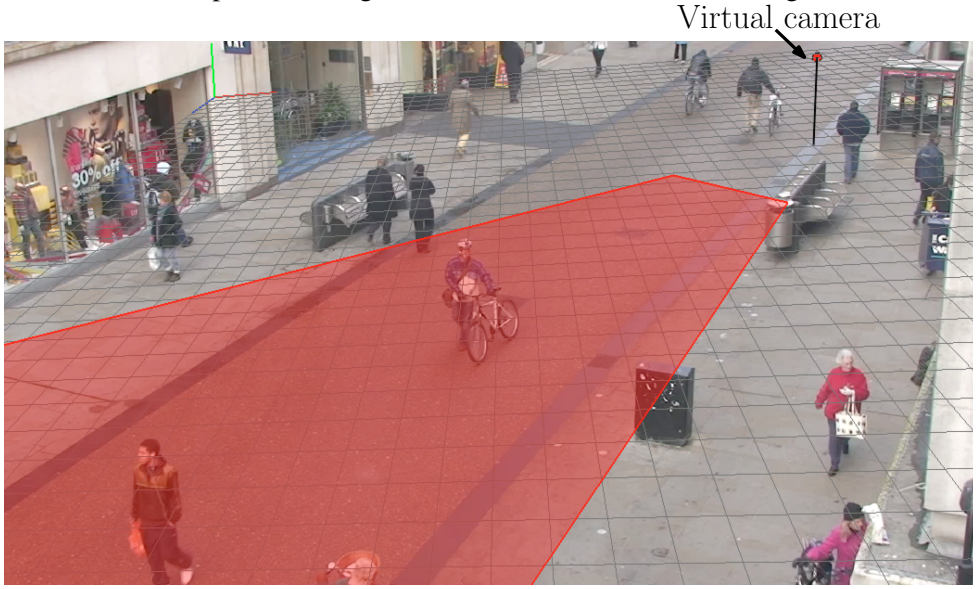
$$P_v = K \begin{bmatrix} R \mid -R\boldsymbol{X}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_1^v & \boldsymbol{p}_2^v & \boldsymbol{p}_3^v & \boldsymbol{p}_4^v \end{bmatrix}. \tag{3.8}$$

Hence, the homography matrix that maps the ground plane to the image plane of the virtual camera is given by $H_v = \begin{bmatrix} \boldsymbol{p}_1^v & \boldsymbol{p}_2^v & \boldsymbol{p}_4^v \end{bmatrix}$, so that any image pixel (in homogeneous coordinates) $\hat{\boldsymbol{u}}$ on the ground plane maps to another pixel (in homogeneous coordinates) $\hat{\boldsymbol{u}}_v$ of the virtual camera through

$$w\hat{\boldsymbol{u}}_v = H_v^{-1} H \hat{\boldsymbol{u}}. \tag{3.9}$$

Fig. 3.6 shows an example of virtual camera placement based on a frame of the TownCentre dataset (BENFOLD; REID, 2011), widely used for vision-based video surveillance. In this example, a virtual camera with focal length of $1000\ px$ and resolution of 640×480 was placed on the top right region of the scene, and the planar area viewed by the camera is highlighted in red. It is important to note that our system only provides the ground plane projection, so that occlusions due to non-planar objects in the field of view of the virtual camera are not handled. However, the proposed self-calibration approach could also be used to leverage interactive methods for image-based 3D reconstruction, such as (JIANG; TAN; CHEONG, 2009), allowing the inclusion of 3D objects as well in the camera preview.

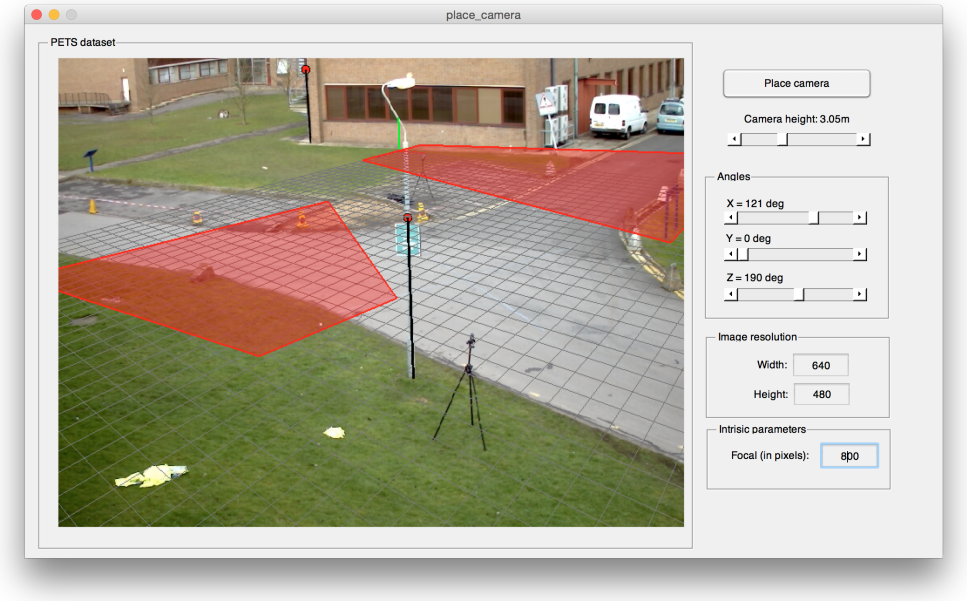
Figure 3.6 – Ground plane coverage (in red) of the virtual camera using our self-calibration.



Source: Author

Fig. 3.7 illustrates the Graphical User Interface (GUI) that we developed applied to PETS dataset. This example illustrates the placement of two virtual cameras at existing structures (one is placed on a light post, and the other on the wall of a building, indicated by blue arrows), as well as the corresponding planar views. The GUI allows to interactively change the focal length and camera rotation parameters, allowing to evaluate the coverage provided by PTZ cameras.

Figure 3.7 – GUI interface to help the user understand the best camera configuration for a desired coverage of the scene.



Source: Author

3.2 Multiple-person tracking

The goal of our multiple-person tracking is to extract the trajectories of multiple pedestrians observed by a single static surveillance camera. As stated in the Introduction (Sec. 1), we use camera calibration to ease the task of tracking, assuming that a camera projection matrix is provided. From that, we also take the assumption that the ground surface is planar and that the people heights are constant during the scene. Furthermore, we assume that people appearing in the scene have a vertical orientation in the world coordinate system, which is expected for walking pedestrians. As it would be presented later, we represent our targets by using patches that aligned in the WCS, which project to non-aligned patches in the image domain. However, as it is common in the literature, the adopted pedestrian detector (DOLLÁR et al., 2009) in our framework assumes that people appear mostly in upright positions. Tracking can represent the sole goal of a system and its final output can appear in different forms, depending on the

semantic level of information desired (e.g. 2D trajectories vs. 3D trajectories, rigid body vs. deformable parts). In this dissertation, we are interested in a method that will provide enough information for our collective behavior extraction algorithm. In this context, it is important that our tracker should be both causal, aiming to reduce delays on the whole system, and extract ground plane trajectories, so that the final goal (collective behavior detection) could be applied to a wider range of camera setups. With that in mind, we devised an online method that is able to combine different cues of information to track pedestrians, making use of camera calibration at several stages in the process.

The proposed approach consists of initially detecting the targets (pedestrians), and representing each target as a set of patches. The patches related to each pedestrian are then tracked individually, and their motion patterns are combined in a robust manner in the WCS using a Weighted Vector Median Filter (WVMF). We deliberately chose to not rely too much on appearance features, since different people can appear to have very similar appearance in a surveillance scenario – specially when the quality of the streams are of only moderate quality. A predicted motion vector and a pedestrian detector are also included in the tracking framework to improve accuracy and to better handle occlusions. The steps of the proposed method are detailed next.

3.2.1 Automatic initialization and patch creation

The first step of our approach is to initialize the tracks using a combination of pedestrian detection and background removal in an automatic procedure. For the detection step, although several algorithms may be used, we once again have chosen the detector proposed by Dollár et al. (DOLLÁR; BELONGIE; PERONA, 2010), which presents a good trade-off between accuracy and speed. After the bounding boxes of the pedestrians are found in the image by running this algorithm, the next step is to validate the detection results (i.e. to remove false positives) and to obtain a more accurate representation of each person, which will be used to create the multiple patches. For that purpose, we apply a background removal algorithm to extract the foreground blobs at each frame. Again, there are innumerous background removal algorithms proposed in the literature and we have chosen ViBE (BARNICH; DROOGENBROECK, 2011) due to its good performance for surveillance videos and its capability of adapting the background model at runtime.

For each detected pedestrian, we compute the percentage of foreground pixels inside the bounding box. If this value is below a threshold t_a, the detection is rejected as a false positive

– notice that this is the same scheme used in the proposed self-calibration method to remove incorrect detections. For the detections that were considered valid, the objective is to find a line segment along the body of each person that will be used to align the patches. To this end, several "body hypotheses" are created by considering line segments that originate from the lower portion of the bounding box (possible feet candidates), and that project to vertical line segments in the WCS (we assume that pedestrians are standing). Again, we tried to achieve a similar goal in the self-calibration method when we extracted the so-called people poles. However, here we already have the camera calibrated, so we make use of it to generate a better vertical alignment w.r.t. the target.

More precisely, we define a set of feet candidates $[u_f, v_f]^\mathsf{T}$ using the lower edge of the bounding box which is sampled to create these points. For each candidate, we compute the corresponding ground plane coordinates $[X_f, Y_f]^\mathsf{T}$ using the ground plane homography H, as follows:

$$w \begin{bmatrix} X_f \\ Y_f \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} u_f \\ v_f \\ 1 \end{bmatrix} . \tag{3.10}$$

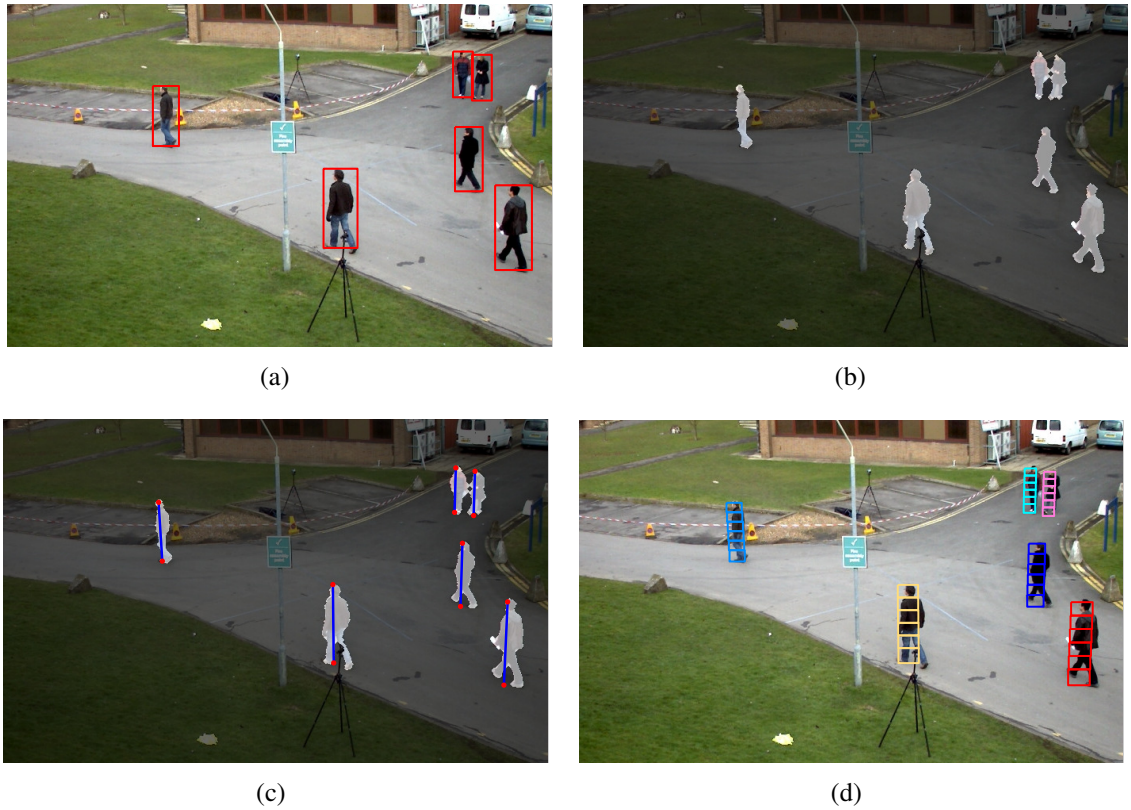
So, each feet candidate $[u_f, v_f]^\mathsf{T}$ point is used to generate a vertical 3D-line defined as $x = X_f$, $y = Y_f$ and $z > 0$. These lines are then projected into the image plane using the camera projection matrix. We define the best hypothesis as the one in which the projected line and the foreground are best aligned. In order to discover that, we count the number of foreground pixels that are both along the projected lines and inside the bounding box. To account for moving people, we only use the superior two thirds of these lines. The line with the highest number of foreground pixels is selected. Once the line along the body of a person is chosen, it is necessary to recover both of its end points, located in the person's feet and head points. Obviously, the feet point is the candidate $[u_f, v_f]^\mathsf{T}$ that generated the best hypothesis. The head point is simply the last foreground pixel that is intercepted by the projected line found in a bottom to top search. This procedure allows the method to correctly estimate the body orientation even if the bounding box provided by the pedestrian detector is not accurate, and can also cope with different leg positions (spread apart or close together).

Finally, the patches are created such that their centers lay on the line segment related to the body of the person. The total number of patches N_p created at initialization is defined by the user. The width of the patches is also defined by the user and its value usually depends on the desired pacth aspect ratio. It is important that this definition does not include too much of the background, otherwise the matching may diverge to it. As we are going to present in the next

section, the tracking procedure uses the height of the patches (in the WCS) to reconstruct the displacement vector of each patch. Therefore, the z coordinate of each patch center is extracted by using the ground position $[X_f, Y_f]^\mathsf{T}$ and the camera projection matrix in Equation (3.13). Later, we refer to this value as the height L_i of the central point of patch i.

Figure 3.8 illustrates the automatic initialization procedure. An example of pedestrian detection for a given frame is shown in Figure 3.8(a), and the foreground pixels identified by the background removal algorithm are shown in Figure 3.8(b). The line segments related to the body of detected pedestrians are shown in Figure 3.8(c), and the corresponding patches illustrated in Figure 3.8(d).

Figure 3.8 – Initialization procedure. First the pedestrians are detected (a). Then, a background subtraction is performed (b) to localize the head and feet points of each foreground blob (c). Finally, patches are created in alignment to this line (d).



(a) (b)

(c) (d)

Source: Author

The initialization procedure is performed at each frame to account for new people appearing in the scene. To avoid duplicate tracks we reject all detections for which the overlap between associated bounding boxes and any of the existing targets is above a percentage threshold. The percentage is computed as the ratio between the intersection pixels and the total number of pixels inside the bounding box of the detection. The threshold used in our experiments was 1%, so that we allow only virtually no overlaps. Indeed, we observed in our experiments that

if the target is not already being tracked (and therefore needs to be initialized), it is better to perform the initialization in a frame where the person appears alone and not occluded by some other target.

3.2.2 Patch matching

Each pedestrian region is divided in a set of non-overlapping patches, which are tracked individually. There are several potential features to be use to describe these patches, and a variety of methods for computing the similarity between two image regions. Different methods can be used in our tracking framework, with the only requirement that a distance metric is provided to evaluate the similarity between two image regions. In this work we use color histograms as features. It is clear that more recently proposed features, like those based on convolution neural networks (SADEGHIAN ALEXANDRE ALAHI, 2017), would increase the accuracy of our technique. Nonetheless, this dissertation is interested in providing a framework that is able to combine different cues for tracking and analyze their relevances in the complete approach.

To better cope with illumination changes, the proposed histograms involve only the chromaticity information in the CIELab space, i.e., channels *a* and *b*. For efficiency reasons, we assume that these two channels are independent, so a given image region is described by two normalized histograms h^a and h^b with N_b bins, so that they can be viewed as discrete probability density functions. Given the histograms h^a_m and h^b_m related to the model (for the a- and b-channels, respectively), and given the corresponding histograms h^a_i and h^b_i related to a candidate region i, we use the Bhattacharyya distance between the candidate histograms and the model histograms (GALL et al., 2010):

$$b_i = \frac{1}{2}\left(\sqrt{1 - BC(h^a_m, h^a_i)} + \sqrt{1 - BC(h^b_m, h^b_i)}\right), \tag{3.11}$$

where BC is the Bhattacharyya coefficient defined as follows:

$$BC(h_1, h_2) = \sum_{j=1}^{N_b} \sqrt{h_1(j)h_2(j)}. \tag{3.12}$$

Given the patch model at a frame and a set of candidate patches in the subsequent frame, the model patch is matched to the candidate patch that presents the smallest Bhattacharyya distance. One simple way of creating the candidate patches is to determine a fixed region around the previous position of the patch and then exhaustively search for the candidate that minimizes

the matching distance, as in (ADAM; RIVLIN; SHIMSHONI, 2006; DIHL; JUNG; BINS, 2011). However, since we are dealing with pedestrians and have knowledge on camera parameters, it is possible to use the maximum displacement allowed for a person (in the WCS) to create a customized search region.

More precisely, given the maximum speed s_{max} (in meters per second) allowed for a pedestrian and the frame rate F_r (in frames per second) of the video sequence, the maximum inter-frame displacement for each pedestrian in the WCS is $r = s_{max}/F_r$. However, to make the tracker more adaptable to different situations and to allow it to recover from failure, we introduce a relaxation parameter $\alpha_r > 0$ and compute an extended radius $\hat{r} = (1 + \alpha_r)r$. Also, to simplify the geometry of the search region, we actually consider a square with dimensions $2\hat{r} \times 2\hat{r}$, which encloses the circle with the maximum possible displacement. This square region is projected to the image plane and then sampled to create the candidates. Additionally, instead of centering the search region in the previous target position, we first compute a motion prediction vector based on the displacement history of the target. This predicted vector is then added to the previous position to generate a ground point which is used as the center of the search region. Section 3.2.3 presents the motion prediction step in details.

Figure 3.9 shows the projected region in two distinct frames. As the number of candidates in a region is proportional to its size when projected, this value is not constant over time (the search region is larger when the target is closer to the camera and smaller when it is far from the camera).

Figure 3.9 – Search regions for a subject in two different frames (red dashed line). For clarity, the regions shown here were multiplied by a scale factor.

(a) (b)

Source: Author

Once the patches are matched against the possible candidates, displacement vectors in the world coordinate frame are extracted. More precisely, given the central points $\boldsymbol{c}_i = [u_i, v_i]^\mathsf{T}$ of the patches in image coordinates, and $\boldsymbol{d}_i = [\Delta u_i, \Delta v_i]^\mathsf{T}$ the associated displacement vector

computed in the patch matching step, it is possible to estimate the motion vector in the WCS based on the camera matrix (by assuming that the movement is parallel to the ground plane). First, it is necessary to reconstruct the point that represents the displaced patch center $\boldsymbol{m}_i = \boldsymbol{c}_i + \boldsymbol{d}_i$ in the WCS. This 3D point will be referred as $\boldsymbol{M}_i = [X_i, Y_i, Z_i]^\mathsf{T}$. Assuming that any displacement vector $\boldsymbol{d}_i$ corresponds to a translational displacement in the WCS that is parallel to the ground plane, Z_i can be set to a fixed value, namely the height L_i of the patch computed at initialization. The projection of $\boldsymbol{M}_i$ in the image plane is then given by Equation (3.13):

$$w \begin{bmatrix} u_i + \Delta u_i \\ v_i + \Delta v_i \\ 1 \end{bmatrix} = P \begin{bmatrix} X_i \\ Y_i \\ L_i \\ 1 \end{bmatrix}, \tag{3.13}$$

where P is the 3×4 projection matrix and w is the projection scale parameter. With the analysis of equation (3.13), it is possible to assert that X_i and Y_i are obtained by solving a simple linear system of two equations and two unknowns. Finally, the displacement vector $\boldsymbol{D}_i$ in the WCS associated with the patch i is given by the difference between the reconstructed point $[X_i, Y_i]^\mathsf{T}$ and the world point associated with the original patch center $[u_i, v_i]^\mathsf{T}$ and reconstructed using the same technique. The displacement vectors $\boldsymbol{D}_i$ of each target will be robustly combined using WVMF. Additionally, a prediction vector and a vector related with the nearest detection will also be included in the filter.

3.2.3 Motion prediction

Pedestrians typically move along relatively smooth trajectories, without sudden turns. Hence, given the temporal series of displacement vectors $\boldsymbol{D}(t)$ in the WCS, we can predict the displacement vector $\boldsymbol{D}^p(t+1)$ at the subsequent frame $t+1$. Although there are several predictive filters, we have used the Double Exponential Smoothing technique (LAVIOLA, 2003) which is very efficient and has a prediction performance shown to be equivalent to Kalman filters in the original paper (LAVIOLA, 2003) (in the context of predicting a user's pose). A predicted displacement vector $\boldsymbol{D}^p(t+1)$ is obtained with the following expression:

$$\boldsymbol{D}^p(t+1) = \left(2 + \frac{\alpha}{1-\alpha}\right) \boldsymbol{D'}(t) - \left(1 + \frac{\alpha}{1-\alpha}\right) \boldsymbol{D''}(t), \tag{3.14}$$

where α is the smoothing parameter, $\boldsymbol{D'}(t)$ and $\boldsymbol{D''}(t)$ are auxiliary variables computed through

$$\boldsymbol{D'}(t) = \alpha \boldsymbol{D}(t) + (1-\alpha)\boldsymbol{D'}(t-1), \tag{3.15}$$

$$\boldsymbol{D''}(t) = \alpha \boldsymbol{D'}(t) + (1-\alpha)\boldsymbol{D''}(t-1). \tag{3.16}$$

The predicted motion vector can play an important role when all the patches were badly matched (e.g. during a total occlusion). It may also be used to remove the jitter of the target trajectories by continuously smoothing the current position with the displacement history.

3.2.4 Combining motion cues using WVMF

Let us consider a given pedestrian and a specific frame t, at position $\boldsymbol{X}(t)$ in the WCS. For this pedestrian, we have a set of N_p displacement vectors $\boldsymbol{D}_i$ related to the individual patches, and another displacement vector $\boldsymbol{D}_{N_p+1} = \boldsymbol{D}^p(t+1)$ (in the WCS) related to the predicted motion vector, as described in Section 3.2.3. In pedestrian tracking, it is natural to assume a translational motion of the body (disregarding arms and legs movements), which is parallel to the ground plane. In this context, all displacement vectors, when computed in the WCS, should be similar.

The Weighted Vector Median Filter (WVMF) (ASTOLA; HAAVISTO; NEUVO, 1990) is a flexible tool to compute weighted averages of vectors, implicitly detecting and removing the influence of outliers. Hence, it seems adequate to combine all the displacement cues for a given pedestrian (patches + predicted) into a single translational displacement in the WCS. Given a set of $N = N_p + 1$ displacement vectors (where N_p is the number of patches), the first step of WVMF consists of computing the distance from each vector to all others:

$$s_i = s(\boldsymbol{D}_i) = \sum_{j=1}^{N} \|\boldsymbol{D}_i - \boldsymbol{D}_j\|, \;\; i = 1, ..., N, \tag{3.17}$$

where $\|\cdot\|$ is a vector norm (in this work, we employed the L_2 norm). The filtered displacement vector $\boldsymbol{D}_f$ is then defined according to

$$\boldsymbol{D}_f = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \boldsymbol{D}_i, \tag{3.18}$$

where $w_i = f(s_i)$, and f is a nonnegative monotonically decreasing function (so that vectors

that are farther from the median are associated with smaller weights).

As in (FÜHR; JUNG, 2012), we use a modification of the weights w_i that also includes the matching error b_i of each patch in the filtering process, such that weight decreases as function of both distance from the median and matching error. More precisely, the proposed weights for the WVMF are given by

$$w_i = e^{-\left[(s_i/\beta)^2 + (b_i/\gamma)^2\right]}, \tag{3.19}$$

where β and γ are parameters that control the decay of the weight. As in (DIHL; JUNG; BINS, 2011; FÜHR; JUNG, 2012), the β parameter is defined adaptively as the minimum value of the distances s_i at each time step.

It is important to point out that the displacement vector $\boldsymbol{D}_{N_p+1}$ does not originate from patch matching, but instead it is computed using motion prediction. Hence, the vector does not have a matching error. To include it in the WVMF formulation, its matching error b_{N_p+1} is set artificially to the median value of the best matching distances extracted in the last T_p frames. In this manner, the predicted vector always has a strong importance in the WVMF such that, when the weights of the patches decrease, the prediction automatically gains importance in the computation of the filtered vector, which is useful particularly in occlusions. This scheme is similar to the one proposed by Dihl et al. (DIHL; JUNG; BINS, 2011). However, preliminary results indicated that the predictions tend to be corrupted during longerterm occlusions. In fact, the performance of the tracker is actually reduced rather than improved when using these predictions because the algorithm implicitly trusts a prediction that is corrupted. To avoid this problem, our system only updates the current prediction of a target if the median matching error of the patches is not considered an outlier with respect to past frames (we use the same statistical approach presented in Section 3.2.7 to determine this). However, if this error is detected as an outlier, we assume that the tracker has performed poorly (possibly due to an occlusion) and the last computed prediction is kept in order to not corrupt the displacement estimate.

After applying the WVMF, the displaced points in the WCS are projected back onto image using the heights L_i of the patches; these projections correspond to the new patch centers in the image. This is done because, as mentioned before, even if the points have the same X and Y coordinate in the world, their projections do not necessarily constitute a line in the image plane vertical orientation. The effect of moving patches in this manner is that their centers will always correspond, in the world, to a 3D line that is perpendicular to the ground plane. Therefore, the tracker automatically adjusts the orientation among the patches at each time step in a way that is coherent to what is observed in the image. Finally, the position of the pedestrian at time $t+1$ is given by $\boldsymbol{X}(t+1) = \boldsymbol{X}(t) + \boldsymbol{D}_f$, where $\boldsymbol{D}_f$ is the result of the WVMF according to

Equation (3.18).

3.2.5 Refining the tracks with detection results

A pedestrian detection scheme (DOLLÁR; BELONGIE; PERONA, 2010) is used in the proposed approach to initialize the tracks. However, the information provided by the detector can also be explored in the tracking framework, since it provides additional indications on where each person is located. In this work, the output of the pedestrian detector is used as an additional cue to find the motion of each target, along with the displacement vectors provided by the multiple patches and the predicted motion vector. For that, the set of the pedestrian detections is analyzed at each frame both for initialization purposes and also to create additional vectors to be included in the WVMF, which we refer as "detection vectors". This is done using a simple yet efficient association procedure as follows.

Again, let $\boldsymbol{X}(t)$ denote the ground plane position (in the WCS) of a given pedestrian at frame t, and let $\boldsymbol{u}_1^d$, $\boldsymbol{u}_2^d$, ..., $\boldsymbol{u}_K^d$ denote the lower middle point of the bounding box (in image coordinates) of the K detected pedestrians at frame $t+1$. As before, we use Equation (3.10) to find the ground plane positions $\boldsymbol{X}_1^d$, $\boldsymbol{X}_2^d$, ..., $\boldsymbol{X}_K^d$ of the detected pedestrians (for the sake of simplicity, let us assume that $\boldsymbol{x}_i^d$ are defined in ascending order with respect to the distance from $\boldsymbol{x}(t)$). Then, a circular search region centered at $\boldsymbol{X}(t)$ with radius r_d is created, and $K_d \leq K$ detection results are considered possible matches if $\|\boldsymbol{X}(t) - \boldsymbol{X}_i^d\| \leq r_d$.

If only one detection is within the search region (i.e. $K_d = 1$), the associated ground point is used to create a WCS displacement vector $\boldsymbol{D}_{N_p+2} = \boldsymbol{X}_i^d - \boldsymbol{X}(t)$, which is included in the WVMF formulation of Eq. (3.18) with the same matching error as the predicted motion (i.e. using $N = N_p + 2$ and $b_{N_p+2} = b_{N_p+1}$). If $K_d > 1$, more than one detection result could be associated to the target. In this case, the WVMF is initially computed without any detection vector, generating an initial filtered displacement vector $\boldsymbol{D}_f$. From all candidate displacement vectors $\boldsymbol{x}_i^d - \boldsymbol{x}(t)$, $i = 1, ..., K_d$, we select the closest one to $\boldsymbol{D}_f$ (in terms of Euclidean distance), and recompute the WVMF including this detection vector.

As we are going to show in the experiments section, this additional cue greatly improves results by allowing the tracker to recover from failure after an occlusion and also by preventing bad initialized targets to diverge to the scene background.

3.2.6 Scale estimation

The calibration can also be used to compute the target scale in the image. Several tracking algorithms such as (ADAM; RIVLIN; SHIMSHONI, 2006; BERTINETTO et al., 2016) create a large number of candidates at different scales in order to select both the position and the scale at the current frame. However, this is not needed in a calibrated scenario (or at least the number of scales can be greatly reduced using the geometry information provided by the calibration). As shown in (FÜHR; JUNG, 2014), if the scale is correctly estimated at initialization, one can compute the target's height at initialization and kept this value throughout the sequence.

When the target is initialized, the height of the target (in the WCS) can be obtained by solving for L_i in Eq. (3.13), using only the vertical component of the bounding box. Since this height is fixed for a given pedestrian, the feet location of the target is displaced, and the head location in the ICS is computed using the camera projection equation with the height stored at initialization. The patches can also change size in the image due to scale modifications but since we computed the height of the target in the image we simply need to resize the patches according to their new image heights and original aspect ratios – similar to the patch creation procedure. As a consequence, the scale of the target is automatically adjusted during the tracking process.

3.2.7 Track termination

When developing a multi-pedestrian tracker, an important step is to determine when a track is no longer valid. This may happen when the person leaves the viewing area of the camera, or when the tracker gets lost. Although the first case may be tackled by evaluating specific entry-exit zones (such as the borders of the image), it is not trivial to determine if a system is tracking a subject correctly.

The proposed termination procedure is based solely on a quality measure of the tracker: if the tracking of a specific target has generate low matching accuracies for a sufficiently long period of time (T_n frames), the track is terminated. This procedure works either when the person leaves the viewing area of the camera or when the tracker is actually lost.

More precisely, for each pedestrian, the median matching error of the patches is stored at each frame. Then, a statistical strategy for outlier detection (HAN; KAMBER, 2001) is used to detect if the tracker got lost. Initially, the most recent T_n error values in a target history are removed from the history to create a set called $\mathcal{S}_{last}$. The set of all the errors except for the last T_n values is called $\mathcal{S}_{hist}$. To test if the elements in $\mathcal{S}_{last}$ are outliers with respect to the values in $\mathcal{S}_{hist}$,

the third quartile of $\mathcal{S}_{hist}$, denoted by $Q3$, is extracted along with the interquartile range, IQR. Then, the tracker is considered lost if all the values in $\mathcal{S}_{last}$ are larger than $Q3 + 1.5IQR$ (HAN; KAMBER, 2001). In our experiments we used two values of T_n depending on the current position of the target on the image and the sequence frame-rate F_r. If the target is placed on the image borders we used $T_n = k_b F_r$, whereas the value $T_n = k_r F_r$ was used for the remainder of the image. We use $k_b < k_r$ to produce a temporal window T_n that is smaller on the boundary zone because it is assumed that these are usually the exit zones of pedestrians. In the middle of the image, T_n is larger to allow the tracker to recover from failure, which can be induced by an occlusion, for instance.

When the color features used for patch matching are not very discriminative (e.g. colors of a pedestrian similar to the background), an erroneous track that should be terminated may not be detected by the outlier detection rule. To handle such cases, we additionally terminate a track if no result of the pedestrian detector was found within the patch search region in the last $k_d F_r$ frames.

3.3 Collective behavior recognition

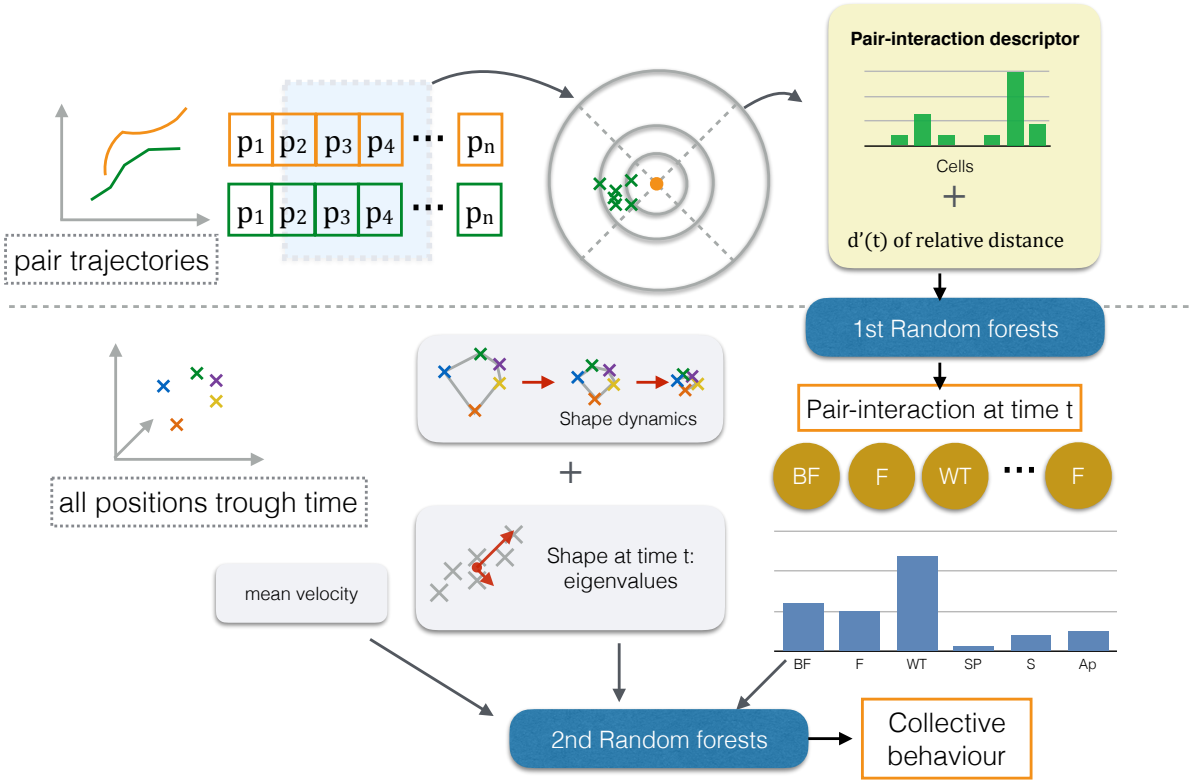
Automatic or semi-automatic analysis of human behavior have been studied by the computer vision community for several years. In particular, there has been increasing interest in inferring semantic information about the relation and interaction among people in a video sequence. In this dissertation, we propose a hierarchical approach for collective behavior detection solely based on the peoples trajectories in the ground plane. In the first level, pairwise interactions between agents are extracted based on the expected distances among people and the dynamics of the relative distance between the pair. Then, additional information such as the group speed and shape are fused with the pairwise interactions using Random Forests to detect higher level collective behavior. Thus, we aim to tackle two goals at the same time. On a global scale, we are trying to detect events that are occurring in the scene involving an arbitrary (and possibly variable) number of people. Additionally, we are also attempting to identify which subjects are involved in this collective activity, as well as their role in it. For instance, we want to recognize which subjects are following others in a *chasing* activity. This information can help surveillance systems to automatically associate levels of importance for each person in the scene and detect mischievous behavior. These pairwise interactions, in addition to being relevant by themselves, also contribute to the collective recognition layer. In order to propose a method capable of adapting to different scenarios and events, we investigate a set of features that are

able to describe a wide range of different behaviors. Yet, in order to motivate our choices, we present our reasoning on a defined ensemble of interactions and behaviors for better illustration, as it is discussed later.

Throughout the remaining of this work, it is important to notice that we assume the pedestrians ground plane positions in the sequence to be given, and they are the only input to the method. As stated before, we chose to use world coordinates instead of image coordinates since they generalize much better across different scenarios, being independent of the camera setup used to acquire the sequence. Also, given our self-calibration method described in Section 3.1, we do not feel that this is a prohibitive constraint. In fact, some of the experiments described in Section 4.3 were performed using our own self-calibration method to automatically extract the planar homography and to convert annotated bounding boxes to ground plane coordinates.

A schematic overview of the method is illustrated in Figure 3.10, and details of each layer in our hierarchical classifier are provided next.

Figure 3.10 – Overview of the proposed method: the trajectories of a pair are described using a number of spatial cells and the derivatives of their relative distance. This is fed to a Random Forest that classifies the interaction among six possible answers. The time-accumulated interactions together with features of shape analysis and speed profiling are given to a second Random Forest, which finally classifies the collective activity observed in the sequence.



Source: Author

3.3.1 Pairwise interactions

Let an interaction between two pedestrians o_1 and o_2 be denoted as $o_1 \rightarrow o_2$, where o_1 is the anchor subject, and o_2 is the target subject. It is important to notice that, unlike some works that define symmetric interactions (e.g. (CHOI; SAVARESE, 2014)), in our work $o_1 \rightarrow o_2$ is different from $o_2 \rightarrow o_1$. By choosing asymmetric interactions, it is possible to identify the role of each person in the observed group/pair activity. For example, in a chasing event we can determine the chaser(s) and the chasee(s) individually.

Before we present our pairwise interaction descriptor, we must define the types of interactions we aim to detect in our videos. They should be interesting enough to provide useful information about the scene by themselves, but also adequate as input cues for detecting higher level collective behavior. We defined six different pairwise interactions: *being-followed* (BF), *following* (F), *walking-together* (WT), *standing pair* (SP), *splitting* (S), and *approaching* (Ap). These interactions were basically the same used in (CHOI; SHAHID; SAVARESE, 2009), but the "follow" behavior was split into following and being-followed here, recalling that our interactions are not symmetric in general, opposed to other methods (CHOI; SHAHID; SAVARESE, 2009; CHOI; SAVARESE, 2012).

The first cue that is important for identifying pairwise interactions is the set of relative positions between two subjects within a time window, which can be explored to detect if they are near/far each other, side by side and so on. In this work, we divide the region around a subject into cells using boundaries in the polar domain (both distances and angles) w.r.t. the anchor subject. One descriptor is created for each one of the neighboring subjects, i.e., we define a pairwise descriptor. More precisely, the distances boundaries in our Personal Interaction Descriptors (PIDs) were obtained according to the studies of Hall (HALL, 1973), which sets different radii for the expected intimate, personal, social and public interactions between two subjects, illustrated at Table 3.1.

Table 3.1 – Distances thresholds for different levels of interactions as proposed by (HALL, 1973) and used in this work to build pairwise interaction descriptors.

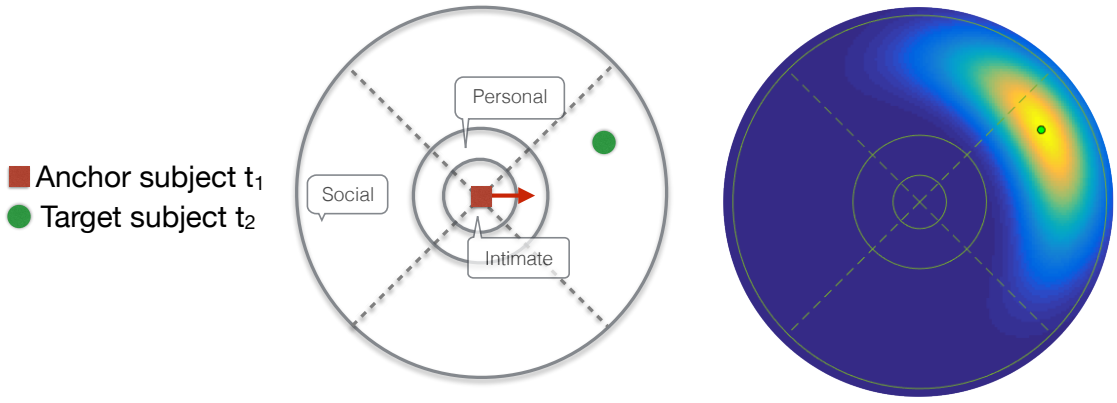
Level of interaction	**Approximated distance**
Intimate	up to 0.5 meters
Personal	0.5 to 1.25 meters
Social	1.25 to 3.5 meters
Public	more than 3.5 meters

In fact, the concept of proxemics was already used in the context of group detection

in some methods, such as (JR et al., 2007; SOLERA; CALDERARA; CUCCHIARA, 2015). Additionally, we divide the region into four equidistant angular sectors to identify back, front left and right sides of the anchor subject. The cell disposition around the anchor is illustrated in Figure 3.11(a), and the cells are aligned (rotated) assuming that the anchor subject is moving horizontally to the right.

It is worth noticing that the spatial distribution of the neighboring pedestrians was also explored in (CHOI; SHAHID; SAVARESE, 2009; CHOI; SAVARESE, 2014). However, they used a classifier to obtain the orientation of each agent, which is based on image features and might be sensitive to the camera setup, and that leads to a higher-dimensional representation. Instead of using image features, we estimate the local orientation of a moving pedestrian based on the corresponding trajectory, filtered by a Double Exponential Smoothing technique (LAVIOLA, 2003) (as used for tracking) to reduce the effect of trajectory jitter. However, one limitation of the proposed method is that it fails for stationary pedestrians, or for slowly moving ones (for which the orientation estimation is very noisy). In fact, when the speed of the anchor subject is smaller than a threshold T_s, we disregard the orientation part of the proposed descriptor by setting a random orientation value for the target pedestrian, so that there is no bias to a particular orientation (and information is encoded by the distance only).

Figure 3.11 – (a) Four angles and three distances are used to divide the region around a subject into bins/cells. (b) A normal distribution is used to introduce a soft boundary between cells.



(a) (b)

Source: Author

We then compute a histogram of relative positions between the targets within a temporal window using the spatial bins illustrated in Figure 3.11(a). More precisely, for a given frame f_t at time t, we analyze a temporal window with T_1 frames (assumed to be a power of two) centered at t, i.e. from $f_{t-T_1/2+1}$ to $f_{t+T_1/2}$. A traditional histogram could be obtained by simply counting the number of relative positions that lie in each bin along all frames in the window. However, this

approach is very sensitive to samples that lie close to the boundaries between bins, so that similar pairwise behaviors may generate considerably different histograms. This problem is amplified when the number of samples is small, as it is often the case on human behavior classification.

A known method for improving the estimate of the underlying probability density function (PDF) from a histogram is kernel density estimation (KDE) (HWANG; LAY; LIPPMAN, 1994), in which a kernel centered at each observation is used to obtain a continuous PDF of the data. In this work, we use a Gaussian kernel defined in polar coordinates (ρ, θ) given by (disregarding normalization):

$$G_{\text{KDE}}(\rho, \theta; \rho_o, \theta_o) = \exp\left[-\frac{(\rho - \rho_o)^2}{2\sigma_\rho} - \frac{d_\theta(\theta, \theta_o)^2}{2\sigma_\theta}\right], \tag{3.20}$$

where ρ_o and θ_o are the polar coordinates of the target subject relative to the anchor (i.e. the location of the sample),

$$d_\theta = \min\{|\theta - \theta_o|, 2\pi - |\theta - \theta_o|\} \tag{3.21}$$

is a function that computes the smallest difference between two angles and σ_ρ, σ_θ are the scale parameters in the distance and orientation domains, respectively.

To obtain the histogram, one could just integrate the kernel-smoothed PDF over each bin. In this work, such integral is approximated by sampling a constant number of points around its center (ρ_o, θ_o) equally spaced in a $K_s\sigma_\rho \times K_s\sigma_\theta$ grid, normalizing and then summing over each spatial bin, where K_s controls the extent of the sampling region. If $h_{o_t}(c)$ denotes[5] the histogram for target o at time t at cell c, it is given by

$$h_{o_t}(c) = \sum_{\tau=t-\frac{T_1}{2}+1}^{t+\frac{T_1}{2}} a_g(\tau) \sum_{\boldsymbol{s}\in S_{o\tau}} \chi_c(\boldsymbol{s}) G_{\text{KDE}}(\boldsymbol{s}; \boldsymbol{p}_{o_t}), \tag{3.22}$$

where

$$\chi_c(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in A_c, \\ 0 & \text{otherwise.} \end{cases}$$

is the indicator function for bin c (A_c is the spatial region that defines c), S_{o_t} is the set of samples from the grid generated by the target o_t, $\boldsymbol{p}_{o_t}$ is the vector containing the polar coordinates of o_t, and $a_g(t)$ is a normalization factor for the samples generated at frame t such that the final histogram sum is equal to one. We show in the experiments (Section 4.3) that the use of the KDE smoothing can improve the discrimination quality of our descriptor significantly.

[5]For the following expressions in this section, let the target subject o_2 at time t be expressed as o_t.

The process for obtaining the KDE-smoothed histograms is illustrated in Figure 3.11(b). More precisely, the heat map shows the PDF induced by the proposed kernel for the target agent in Figure 3.11(a).

The second cue in the proposed PID is the dynamic aspect of a pair trajectory. The KDE-smoothed histogram encodes the cumulative relative position of the target agent w.r.t. the anchor, but temporal information is lost. For instance, in the approaching and splitting interactions, the distances decrease and increase, respectively, in a reasonably-sized temporal window. However, such information is lost when computing the histogram: these two different interactions could lead to the exactly same histogram.

In order to include this information in the PID, we compute the relative speed $d'(t)$, where $d(t)$ is the relative distance between the pedestrians under analysis. We evaluate $d'(t)$ within the temporal window in a pyramidal fashion, and append these values to the PID. At the first level of the pyramid, we take the mean of the derivatives for the whole temporal window. Next, we evaluate the averages at the first and second half, generating two values and so on. More precisely, for each level $l \in \{0, 1, \ldots, l_{max}\}$, where l_{max} is the highest level, we build a 2^l-dimensional feature vector

$$\boldsymbol{d}'_l(t) = (\mu_1^l(t), \cdots, \mu_{2^l}^l(t))^T, \tag{3.23}$$

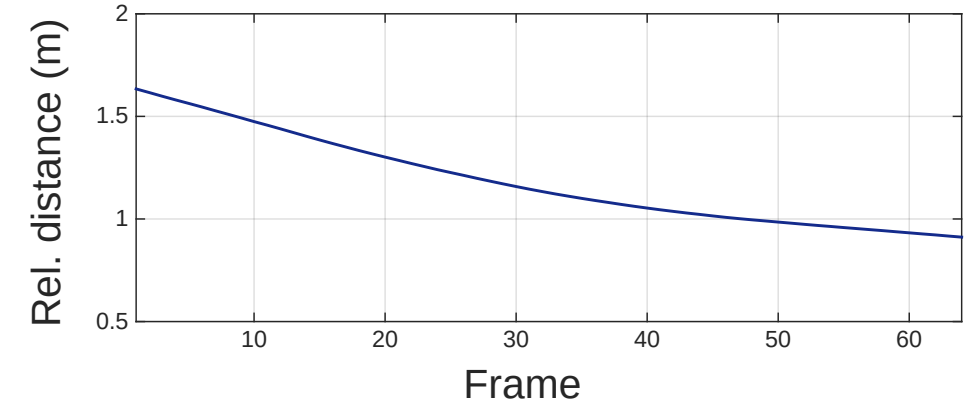
where

$$\mu_k^l(t) = \frac{2^l}{T_1} \sum_{\tau=\frac{T_1}{2^l}(k-1)}^{\frac{T_1}{2^l}k-1} d'\left(t + \tau - \frac{T_1}{2} + 1\right) \tag{3.24}$$

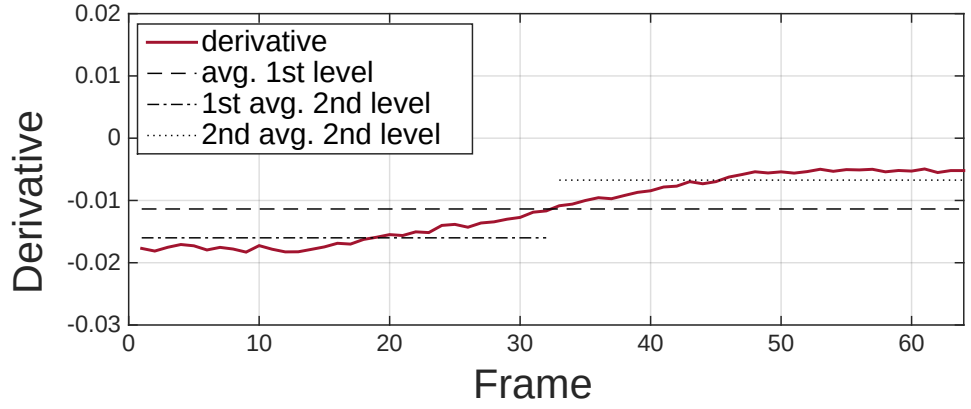
is the average of $d'(t)$ in the k^{th} partition interval at level l. Finally, a consolidated vector $\boldsymbol{d}'(t)$ is obtained by concatenating $\boldsymbol{d}'_l(t)$ for $l = 0, ..., l_{max}$, and it encodes hierarchical information on the relative speed for the pair of pedestrians within the analyzed time window. The dimensionality of $\boldsymbol{d}'(t)$ is $2^{l_{max}+1} - 1$, which is much smaller than using different intervals to compute the spatial histograms, as proposed in (CHOI; SHAHID; SAVARESE, 2009).

Figure 3.12(a) shows the relative distances of two pedestrians computed in a temporal window placed between an *approaching* and a *standing-pair* interaction. The corresponding derivative and the multiscale derivative averages (only two levels) are shown in Figure 3.12(b). As it can be observed, the averages in the two halves of the time interval indicate different behaviors, which could not be captured using the average along the whole interval. In fact, using only a single average value would be prone to noise and would not be discriminative for detecting the transitions between interactions. Also, our aim was to encode this type of information using

Figure 3.12 – The dynamics of the relative distance between a pair (a) is encoded in the multiscale derivative averages (b)



(a)



(b)

Source: Author

a small amount of data, since it is known that classification methods often suffer from the "curse of dimensionality" (HUGHES, 1968), i.e., higher-dimensional feature vectors tend to require larger training datasets.

3.3.2 Collective behavior descriptor and classification

The final goal of our method is to use all the detected pairwise interactions and to classify the collective activity of a given group. Our main hypothesis is that there are different cues of information that are required to describe an interaction or activity. For that reason, our collective descriptor also uses Random Forests to mix different kinds of information. Once more, we extract data in a given temporal window of T_2 (also a power of two) frames and assume that a single activity appears in the scene (as in (CHOI; SAVARESE, 2014; AMER; LEI; TODOROVIC, 2014)), and eventual additional trajectories are rejected. Despite the fact that our collective

method provides a general approach for inferring group activities, in this dissertation we chose to study a subset of them that commonly appear in surveillance systems and represent a good sample of the activities seen in real world applications: *Gathering*, *Talking*, *Dismissal*, *Walking*, *Chasing* and *Queuing*. These interactions were also tackled in (CHOI; SAVARESE, 2012), which also allows comparisons with other methods.

Different types of pairwise interactions are expected to arise in a single collective activity. For instance, let us consider the *queuing* event. People waiting in line are related trough a *standing-pair* interaction and, if a person is directing him/herself to the line, an *approaching* interaction is observed. Even more, if two or more people are advancing significantly in that line, our definition would indicate that there are *following* and *being-followed* interactions appearing.

In order to describe the multitude of these interactions, our Collective Behavior Descriptor (CBD) starts by building a histogram of the pairwise interactions detected by our first classifier within the temporal window. This histogram is normalized to account for variation in the number of people that compose the group, such as its sum is always equal to one. Since we defined six interactions in Section 3.3.1, this histogram represents the first six dimensions in our descriptor.

However, only using pairwise interactions is not enough to differentiate all the classes of collective behavior that we are interested on, since some collective behaviors may present the same distribution of pairwise interactions (e.g. both *Walking* vs *Chasing* could involve *following*, *being-followed* and *walking-together*). To overcome this limitation, we add new features related to the speed and spatial distribution of the observed pedestrians. The first feature is the mean speed $v_\mu(t)$ of the group averaged inside the temporal window:

$$v_\mu(t) = \frac{1}{T_2} \sum_{\tau=t-\frac{T_2}{2}+1}^{t+\frac{T_2}{2}} \frac{1}{\#S_\tau} \sum_{\boldsymbol{s}\in S_\tau} \|\boldsymbol{v}_{s\tau}\|, \tag{3.25}$$

where S_τ is the set of subjects in the group at time τ and $\boldsymbol{v}_{s\tau}$ is the velocity vector of pedestrian s at frame τ. The velocity cue is necessary to reduce the confusion between behaviors such as chasing, talking and walking groups, whose speed profiles are clearly distinct from each other.

Another relevant source of information is the spatial distribution of the pedestrians along the group, and how it changes in time. For instance, in behaviors such as *gathering* and *dismissal*, the group goes from disperse to compact and from compact to disperse, respectively. To encode the temporal variation of the group shape, we first define the group dispersion $\delta(t)$ at frame t as

$$\delta(t) = \sqrt{\frac{1}{\#S_t} \sum_{s\in S_t} \|\boldsymbol{p}_{st} - \boldsymbol{\mu}_t\|^2}, \tag{3.26}$$

where $\boldsymbol{p}_{st}$ is the position of a subject s and $\boldsymbol{\mu}_t$ is centroid of the group at frame t. Finally, the temporal dispersion change $s(t)$ is given by

$$s(t) = \frac{1}{T_2} \sum_{\tau=t-\frac{T_2}{2}+1}^{t+\frac{T_2}{2}} \delta'(\tau), \tag{3.27}$$

i.e. it is the average of the derivatives along the temporal window. In a gathering behavior, $s(t)$ is expected to be negative, where the opposite should happen in a dismissal behavior. For both $s(t)$ and $v_\mu(t)$, we also experimented with the same pyramidal averaging approach described in Section 3.3.1, yet the overall classification performance remained almost constant. Therefore, we decided to use just the temporal average (first level of the pyramid) to keep the descriptor compact.

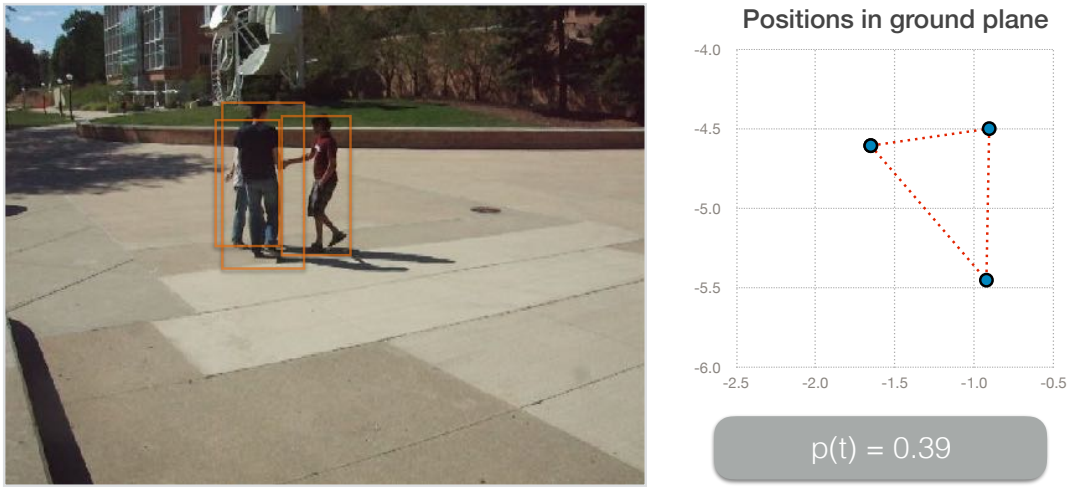
Given the framework described so far, there are two types of behavior commonly observed in surveillance scenarios for which the system would still have trouble differentiating: *queuing* and *talking*[6]. The reason is that the interactions observed will be mainly *standing-pair* and the dynamics of both activities are nearly identical, i.e. the related subjects stand still, so that $v_\mu(t) \approx 0$ and $s(t) \approx 0$. Instead of relying on image cues to obtain the orientation of the pedestrians, we use a feature that allows to distinguish a line (queueing) from a disperse distribution of people. In fact, this descriptor is given by

$$p(t) = \begin{cases} \lambda_{max}/\lambda_{min} & \text{if } \#S_g > 1 \text{ and } \lambda_{min} > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.28}$$

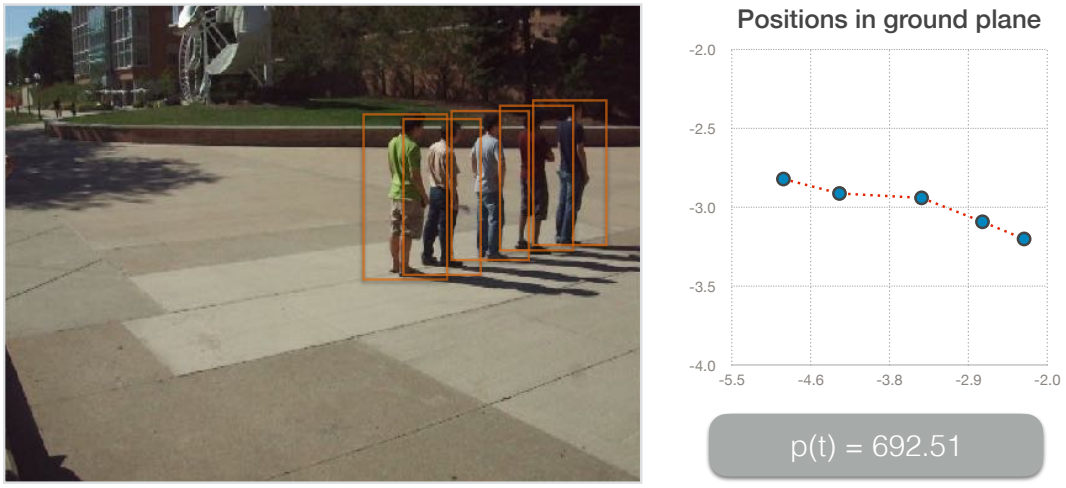
where $\lambda_{min} \leq \lambda_{max}$ are the two non-negative eigenvalues of the covariance matrix of a set of 2D points that represent the subjects ground plane positions at time t. It is worth remembering that we employed a similar idea to compute the quality of a pedestrian foreground mask for self-calibration in Section 3.1. Once again, we compute $p(t)$ at the center of our temporal window and append it to the descriptor, which is fed to the second random forest. Figure 3.13 shows two different values of $p(t)$ for the end of a *Gathering* and *Queuing* event – clearly, the values of function p are able to differentiate the disposition of ground plane positions in the example. We show in our experiments that this single value can make a vital difference in classification accuracy.

[6]To clarify, the *talking* behavior in this dissertation corresponds to a group of still people standing together.

Figure 3.13 – The value of $p(t)$ for two different activities.



(a) Gathering. Source: Author



(b) Queuing. Source: Author

4 EXPERIMENTAL RESULTS

In this chapter, we describe the experiments carried out for self-calibration (and its proposed applications), multiple-object tracking and collective behavior. For self-calibration and tracking, we chose two well know datasets that are publicly available: the PETS 2009 dataset (FERRYMAN; ELLIS, 2010)[1] (sequence *S2.L1* and *View-001*) with 768×576 images and the TownCentre (TC) dataset (BENFOLD; REID, 2011)[2] that is composed of 1920×1080 full-HD images. Both sequences are monocular and contain several people, also providing the ground truth for pedestrian tracking/detection as well as calibration matrices. To further test our geometry-aware pedestrian detector described in Section 3.2.5, we also created a small dataset consisting of a video sequence take from an on-board camera (iPhone 5S smartphone) mounted on the dashboard of a vehicle passing by an urban area. This dataset contains 2013 high-resolution frames (1080p), and a total of 1498 pedestrians annotations (bounding boxes). The sequence, together with its ground truth annotation, is available publicly for future references and benchmarks[3].

For collective behavior we used the dataset provided by Choi et al (CHOI; SAVARESE, 2012)[4], which contains people locations, interactions and group activities annotated for a total of 33 sequences captured in the same camera setup. Since our recognition method requires ground plane trajectories, we calibrated the camera using our self-calibration method described in this dissertation – the quality of calibration was inspected visually. Please notice that we revised the interaction annotations included in the dataset, since our set of interactions differ a little with respect to the work of Choi and colleagues (see Section 3.3.2 for details). For collective behavior recognition we used annotated bounding boxes to evaluate our method using the same protocol as other state-of-the-art methods (CHOI; SAVARESE, 2012; AMER; LEI; TODOROVIC, 2014). To test the ability of our method to generalize between different camera setups we also used the BEHAVE dataset (BLUNSDEN; FISHER, 2010) for testing[5]. Figure 4.1 shows sample frames of the five datasets used in the experimental analysis.

[1] PETS dataset is made available at <http://www.cvg.reading.ac.uk/PETS2009/a.html>

[2] TownCentre dataset is available at <http://www.robots.ox.ac.uk/~lav/Research/Projects/2009bbenfold_headpose/project.html>

[3] The car dataset can be obtained at <https://github.com/gustavofuhr/car_pedestrian_dataset>

[4] Available at <http://www-personal.umich.edu/~wgchoi/eccv12/wongun_eccv12.html>

[5] <http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/>

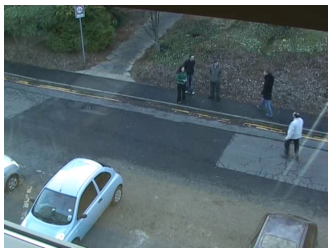
Figure 4.1 – The five different datasets used in the experiments of this dissertation.

(a) PETS

(b) TownCentre

(c) Car

(d) Choi



(e) BEHAVE

Source: Author

4.1 Experiments on self-calibration and geometric-aware detection

The experiments described in this section aim to test our self-calibration scheme in addition with the application of pedestrian detection. Since we propose a whole new method for tracking, the experiments of tracking are described separately (Section 4.2). Notice, however, that we use our self-calibration method in the tracking experiments and to calibrate the Choi dataset used for collective behavior. For the majority of the experiments, three camera calibration results were compared: the ground truth camera parameters, the projection matrix used for initialization (see Section 3.1.2) and the final projection matrix after the non-linear optimization (Section 3.1.3) – these last two are identified in the plots as "*sf:initial matrix*" and "*sf:non-linear optimization*", respectively. Additionally, some experiments show comparisons with the method proposed by Lv et al. (LV; ZHAO; NEVATIA, 2002b), which inspired our initial calibration scheme.

To extract the poles as required by our approach, 230 frames were used for the PETS dataset and 1000 frames for the TownCentre dataset (with a step of 5 frames). This is around 40 seconds of video for both datasets, which is a very reasonable initialization window for surveillance systems. The number of poles extracted was for 376 and 252 for the PETS and the TownCentre datasets, respectively. The minimum foreground ratio was set to $t_a = 0.2$, which was able to reject false positives without relaying too much on the background segmentation

(which could be noise). We chose $q_s = 0.11$, also empirically, for which we observed that effectively removed poorly segmented pedestrians. Furthermore, we evaluate in the next section the impact of reducing the number of poles as input to our method.

Once the poles are extracted, the initial calibration takes about one minute to estimate the initial projection matrix. To create vanishing points we set the minimum distances between two poles, $d_p = 20$ and the parameters of the RANSAC were set such as a point is considered an inlier if its distance to the line is less than 75 pixels and the maximum number of interactions is set to 100. Finally, we take the average height a person to be $Z_{avg} = 1.65m$. The non-linear optimization also converges at around one minute. For the applications described in this dissertation, they all require a low error in computing the world height of a person, therefore we favor angular coherence and found experimentally that $\alpha = 0.999$ is a good choice for the cost function (3.2). The threshold for removing duplicate poles was set to $t_{sp} = 5cm$ and sampling square size was set to $2m$. The whole self-calibration procedure was implemented in MATLAB and tested in a 2.4 GHz Intel Core i7 processor with 8GB of RAM. For the optimization, we experimented with two different solvers: the Simplex method (LAGARIAS et al., 1998) and the Levenberg-Marquardt-Fletcher algorithm (FLETCHER, 1971). For the latter, we used an L_2 norm for the cost function given by Eq. (3.4), and both of them converged at roughly the same time. Both solvers have random aspects in their algorithms, so the optimization was run 5 times and the solution with the lowest cost function was kept. In addition, we observed by visual inspection that the Simplex method gives a slight better result and therefore the following experiments made use of the projection matrix provided by this method.

4.1.1 Self-calibration error

The evaluation of a calibration matrix is not as straightforward as it may seem. One could evaluate the error of the parameters themselves (e.g. comparison of individual intrinsic and extrinsic parameters extracted from the projection matrices), but a global quality index based on these individual parameters is difficult to obtain. In this work, we choose to evaluate the calibration by computing the error in the projection, particularly related to vertical poles. The details of this metric are as follows.

First, we extract the central point in the image (assuming that typically the camera is focused at the most relevant portion of the scene) and, using the ground-truth homography, the corresponding point on the ground plane in the WCS is computed. Then, we randomly sample points around this ground plane central point using a Gaussian distribution with standard

deviation equal to 3 meters (set empirically), and create vertical poles at random heights. Once again, we use a Gaussian distribution for the pole heights, this time centered at 1.6 meters and with a standard deviation of 30 centimeters (trying to emulate pedestrians). The error metric is then defined as the mean error of the re-projected poles using the ground truth calibration matrix and the estimated calibration matrix, using Eq. (3.2). Since the poles are created at random, we repeat this process 10 times and report the average here.

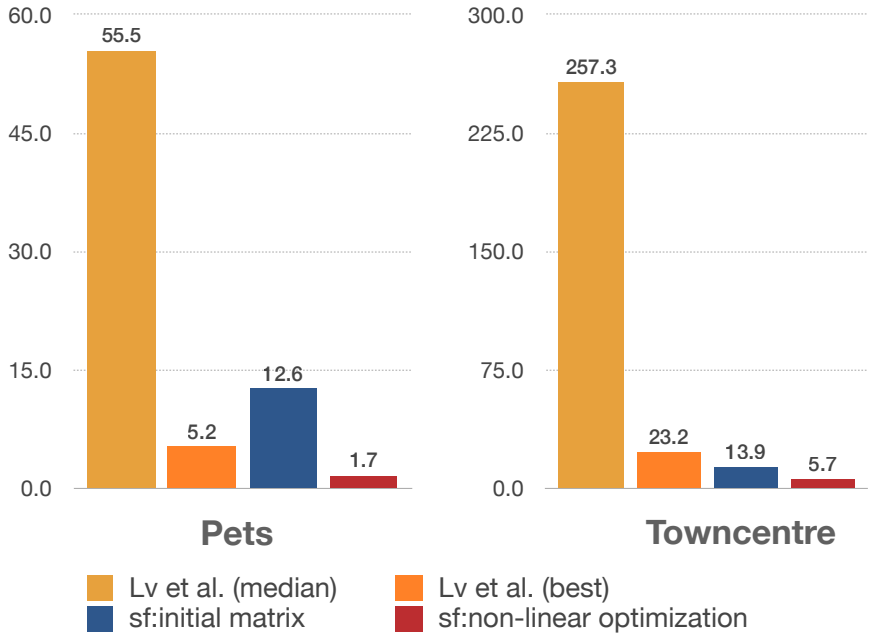
For the experiments regarding the self-calibration error, three methods were compared: i) the projection matrix used for initialization; ii) the final projection matrix after the non-linear optimization; and iii) the calibration technique proposed by Lv et al. (LV; ZHAO; NEVATIA, 2002b). Since (LV; ZHAO; NEVATIA, 2002b) is designed to work for scenarios containing a single pedestrian, we selected 10 random pedestrians (in each scenario) to generate 10 calibration matrices, and then extract the best and median errors.

The comparison of the evaluated camera calibration methods is shown in Figure 4.2, and our full calibration procedure presented the best results in both datasets. It is also interesting to note that our initial calibration scheme is better than the median errors of (LV; ZHAO; NEVATIA, 2002b), although the execution of (LV; ZHAO; NEVATIA, 2002b) for the best single pedestrian presented lower errors than our initial calibration for PETS. In fact, this result was expected: Lv et al.'s approach is designed for controlled conditions, such as a single pedestrian walking, covering the majority of the scene in its path and without many occlusions occurring, as for some of the pedestrians in the PETS sequence. For the TC sequence, however, people walk mostly along straight lines – i.e. they do not wander around the scene, and even the result with the best pedestrian is poor. Another consideration about (LV; ZHAO; NEVATIA, 2002b) is that they choose the frames in which the poles are extracted by using the detection of gait cycles. This is not robust for the majority of subjects tested due to a number of reasons: when the subject is occluded or stops for a significant number of frames, if the background segmentation is not very clean or if the orientation w.r.t. the camera is not at a good angle to see the legs crossing, such as when the person is walking toward the camera.

We also evaluated the impact of the number of poles used as input for our method. To accomplish that, we randomly sampled increasing percentages of the whole set of poles. We then measured the average calibration error and its standard deviation for 50 runs at each percentage value. The results are depicted in Figure 4.3.

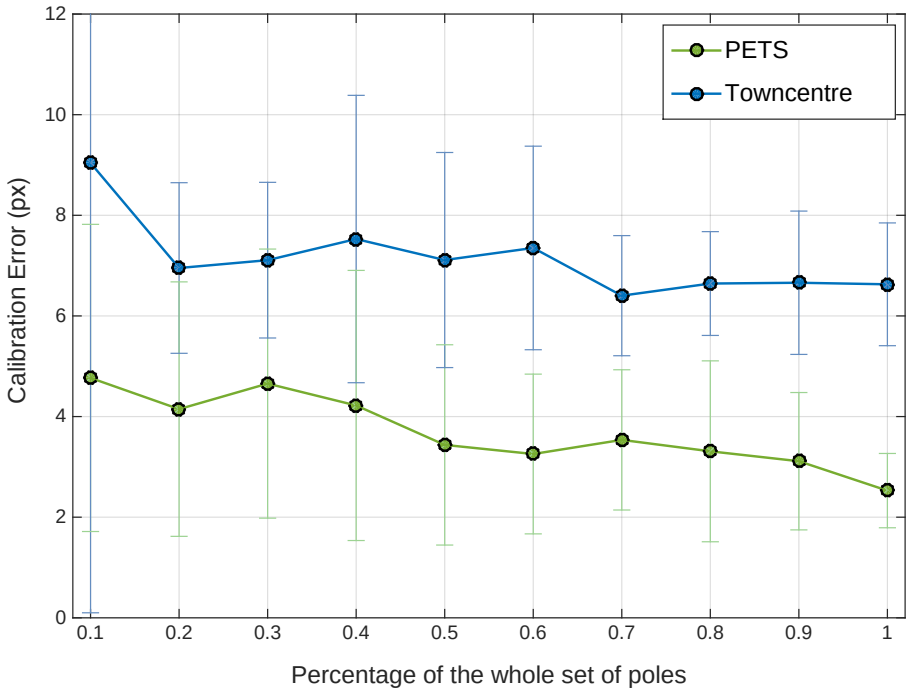
These results indicate that although the error and standard deviation present a tendency to decrease when a large set of poles is used, it is still possible to achieve good results when a small number of poles is provided to the method. Indeed, we observed in our experiments that it

Figure 4.2 – Calibration error of the Lv et al (LV; ZHAO; NEVATIA, 2002b) (best subject and median error) and the two stages of our method. The error is derived from Eq. (3.4) applied to multiple poles using the projection matrix. See text for more details.



Source: Author

Figure 4.3 – Calibration error for different percentages of the initial set of poles.



Source: Author

is more important to have poles that are well spread in the scenario than to have a large number of them concentrated in a small region of the scene.

4.1.2 Geometry-aware pedestrian detection

To test the improvement for pedestrian detection, we created a modified version[6] of the method proposed by Dollár and colleagues (DOLLÁR; BELONGIE; PERONA, 2010) to reduce the number of candidates by only creating geometrically coherent ones, as described in Section 3.1.4.1. We create candidates by analyzing the ground plane image pixels with stride equal to 10 and set 5 uniformly spaced heights, with $k = 2$ for Eq. (3.7). We also implemented a modification to the traditional pedestrian detector based on HOG+SVM (DALAL; TRIGGS, 2005) as well. This implementation uses the OpenCV standard models and the source code was made available[7]. The following results for this modification were made with a pyramid of images of 10 levels using a reduction of 5% at each level and 3 uniformly spaced heights, with stride equal to 10.

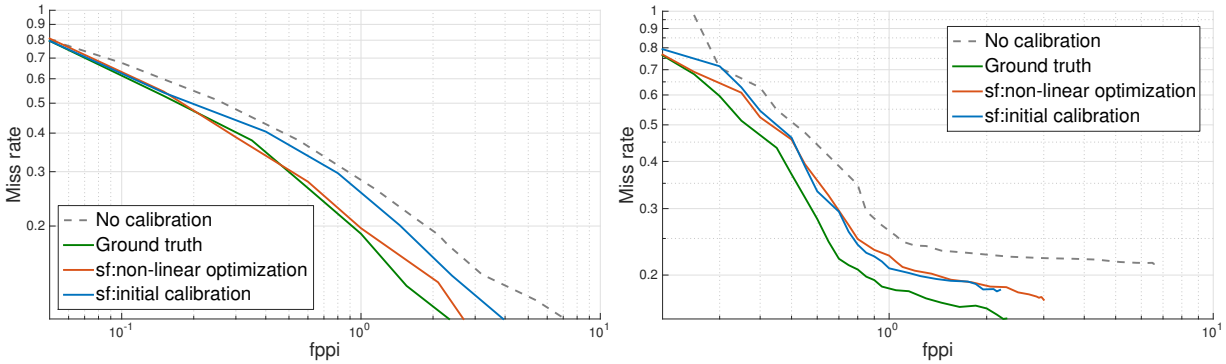
We show results, as described in (DOLLAR et al., 2012), using the number of false positives per image (FPPI) against the miss rate ($1-$recall) with log-log scaled axes. As described in (DOLLAR et al., 2012), a good comparison point tends to be around $fppi = 1$, which can be acceptable for many applications. The curves for all the frames in the PETS dataset and 500 frames of the TownCentre dataset, obtaining by varying the acceptance threshold of the detections, are illustrated in Figures 4.4 and 4.5, respectively. The results show that using calibration to generate candidates decreases the miss rate at several values of fppi for both baseline pedestrian detectors, particularly for the PETS dataset. It is also clear that the optimization phase of our self-calibration method is important for our proposed modifications on detectors, since results from *sf:non-linear-calibration* are indeed better than *sf:initial calibration*. This is more noticeable in the TownCentre dataset, where the initial estimate of calibration show a larger inconsistency in the Z-axis.

In addition to these two datasets, we test the pedestrian detection scheme for mounted vehicle cameras using the aforementioned Car dataset. For this dataset (suited for Driver Assistance Systems), a different calibration method was used (PAULA; JUNG; SILVEIRA, 2014), since the camera is not static and background removal is unfeasible. Once again, we applied our modified detector based on the HOG+SVM and Dollár detectors – the results are shown in Figure 4.6. The results indicate that our approach of applying geometric information in the detector's pipeline indeed increases the overall accuracy of the system. The main reason behind this improvement is that the generation of candidates is more coherent with the pedestrians

[6]This project (written in C++), it is open source and can be obtained at <https://github.com/gustavofuhr/opencv_dollar_detector>

[7] <https://github.com/gustavofuhr/pedestrian_detector_calibrated>.

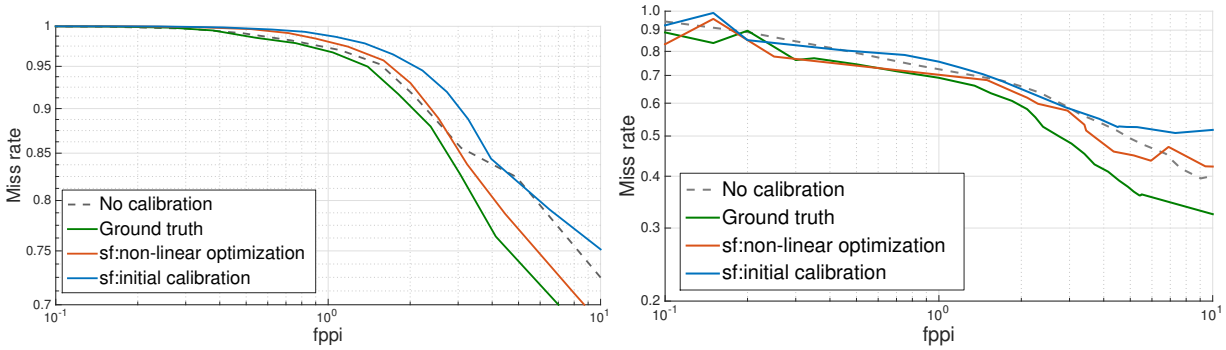
Figure 4.4 – FPPI vs. missrate for PETS dataset.



(a) HOG+SVM (b) Dollár

Source: Author

Figure 4.5 – FPPI vs. missrate for TownCentre dataset.



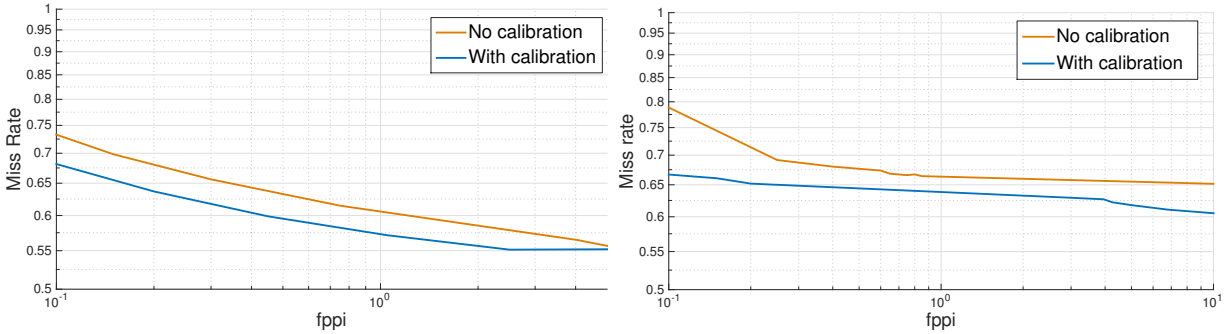
(a) HOG+SVM (b) Dollár

Source: Author

appearing in the scene, and detection results with implausible pedestrian heights are avoided. An example of comparison between the baseline detector and the proposed approach is shown in Fig. 4.7. As it can be observed, the two baseline methods evaluate in the experiments produce detections (marked with arrows) that are clearly incompatible with the dimensions of real pedestrians. On our approach, such candidates would either have a very small score (due to the Gaussian weight based on pedestrian height estimates) or would not even be tested, since their heights are clearly much larger than a expected person at that position.

We also evaluate the number of candidates and running time of the tested pedestrian detectors with and without calibration. Table 4.1 shows the number of candidate bounding boxes used for both baseline methods – Dollár and HOG+SVM – and our modifications to these methods, as well as the execution times (computation of the feature pyramid and total detection times). As it can be observed, we generate a much smaller number of candidates than the baseline method at comparable execution speeds for the Dollár modification and lower execution times for the HOG+SVM baseline method (speed-up factors between 2.5 and 4.1, depending on the

Figure 4.6 – FPPI vs. missrate for Car dataset.



(a) HOG+SVM (b) Dollár

Source: Author

dataset). We believe that the running time did not improve in the case of Dollár because the implementation by the author computes (in a very efficient way) the pyramid of features before the creation of candidate bounding boxes. The time to compute the feature pyramid depends on the minimum and maximum scales: for the baseline detector, we used the default values provided by their implementation[8], and for our method we used the adaptive values described in Section 3.1.4.1.

Table 4.1 – Number of candidates and times comparison for our modifications on both detectors: Dollár(DOLLÁR et al., 2009) and HoG+SVM(DALAL; TRIGGS, 2005).

Dollár baseline	No calibration			Calibration		
	PETS	TC	Car	PETS	TC	Car
Number of candidates (× 1k)	116	666	**134**	**68**	**544**	138
Feature Pyramid Comput. Time (ms)	24.2	142	207	**18**	**133**	**206**
Average time per frame (ms)	**293**	**1551**	**242**	**293**	1760	255

HoG+SVM baseline	No calibration			Calibration		
	PETS	TC	Car	PETS	TC	Car
Number of candidates (× 1k)	71	147	37	**18**	**49**	**4.9**
Average time per frame (s)	6.7	15.9	2.9	**2.3**	**6.4**	**0.7**
Speed-up factor	1×	1×	1×	**2.9**×	**2.4**×	**4.1**×

Since the number of candidates created at a frame depends on the size of the input image and also the range of scales scanned in the baseline approach (which is usually set by the user), we downsampled 1080p images from the Car dataset using multiple scaling factors, and the results are shown in Figure 4.8. Since our method samples the ground plane to generate multiple candidates and the horizon line of the video is within the image plane, such sampling could, theoretically, go on forever. However, we make a threshold based on pixels to limit the minimum height for the creation of candidates. In this experiment, we set the threshold to 10% of the

[8]The original code of Dollár's detector is available at <http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html>

Figure 4.7 – Comparison between the baseline detector (red) and the proposed improvement (blue). Top picture is a comparison with the HOG+SVM detector and the bottom is with Dollár's.

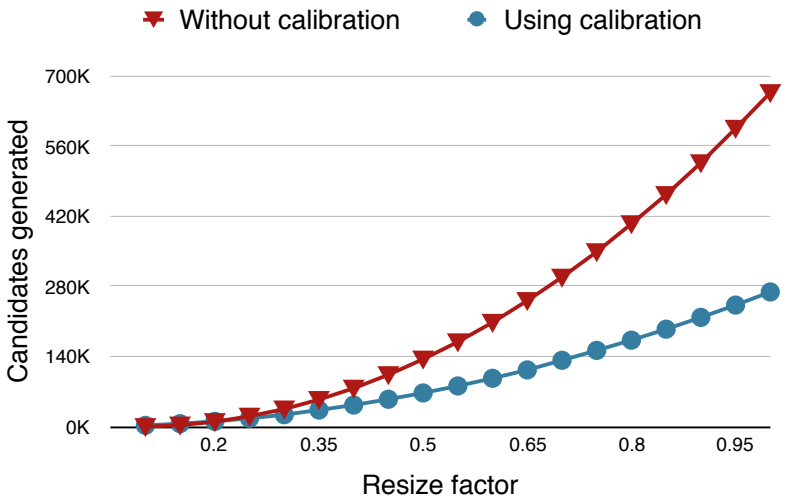
Source: Author

height from the re-scaled image. Clearly, the number of candidates generated by our method is much inferior then a common sliding-window technique.

4.2 Pedestrian tracking

In this section we present several experimental results obtained with the proposed algorithm for pedestrian tracking. First, we evaluate the method using the ground truth annotation provided by the datasets and in Section 4.2.1 we show the results using our self-calibration

Figure 4.8 – Number of candidates generated by the methods as a function of image downsampling factor.



Source: Author

method. For the results described in this section we set the number of patches $N_p = 6$, the number of bins $N_b = 128$ for the color histograms and $s_{max} = 1.5m/s$ as the maximum expected pedestrian speed with a relaxation parameter $\alpha_r = 0.5$. Additionally, the smoothing parameter for motion prediction was set to $\alpha = 0.08$ (see Eq. (3.14)). To combine patch displacement vectors using WVMF we chose $\gamma = 0.25$ as proposed in (FÜHR; JUNG, 2012) and $T_p = 50$ frames. The circular region around a pedestrian to associate with detections was set to $r_d = 1m$. For termination, $k_b = 0.5$ and $k_r = k_d = 3$.

Validation was performed qualitatively, by visual inspection of tracking results, and also quantitatively, by computing tracking errors using ground truth data. We compared our approach to the *FragTrack*[9] algorithm (ADAM; RIVLIN; SHIMSHONI, 2006) and the *TLD*[10] tracker (KALAL; MATAS; MIKOLAJCZYK, 2010). Since neither FragTrack nor TLD were designed for multiple object tracking, we used the initializations provided by the proposed method (variant Proposed-ALL). The termination frames for these two methods were manually specified for the sequences by observing the last frame where the targets appear in the ground truth. For these two methods, each subject was individually tracked and the results were combined to calculate the objective quality metric values. We used the implementations provided by the authors in their websites (see footnotes), and the parameters were set to the values suggested in the corresponding papers. An exception was the size of the search window used in FragTrack, which was set to 30×30. In addition, we also compared our results with the method proposed

[9]FragTrack: <http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm>
[10]TLD: <http://info.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html>

by Pirsiavash et al. (PIRSIAVASH; RAMANAN; FOWLKES, 2011), referred as *DP+NMS*[11]. As mentioned before, this method is non-causal (explores future information), which contrasts with our online causal approach.

In order to verify the impact of different components of our method on the quality of tracking results, we tested 4 variants of the proposed algorithm. The first one is a variant that includes neither predictions nor the detection vectors; we call this implementation *Proposed-NP+ND*. This version can be considered almost the same approach as (FÜHR; JUNG, 2012), but applied to multi-target tracking, i.e. it includes only our initialization and termination steps. The second version, called *Proposed-ND*, only removes the detection vectors. The third variant (*Proposed-NP*) only excludes the prediction vectors in the WVMF computation. Finally, the variant called *Proposed-ALL* is the implementation of all the components described in this dissertation.

To analyze the tracking results quantitatively, we applied the MOT metrics (BERNARDIN; STIEFELHAGEN, 2008), which evaluate the tracking performance for a given multi-target scenario using two indicators: the Multiple Object Tracking Accuracy (MOTA) and the Multiple Object Tracking Precision (MOTP). The MOTA performance is defined by equation (4.1):

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \tag{4.1}$$

where m_t, fp_t, mme_t and g_t are the number of misses, false positives, mismatches and number of objects present at time t, respectively. Therefore, an ideal tracker should present the highest possible MOTA value of one, while lower values (that can be even negative) indicate a higher fraction of errors. Additionally, the MOTP represents the average error of the tracks that were considered correct (error distance below a certain threshold). Therefore, a good tracking method must have high MOTA and small MOTP values.

In order to use the MOT metrics, both a distance function and a threshold must be specified. Several authors use the overlap between the ground truth and the tracking result as a distance (BREITENSTEIN et al., 2011). However, this metric tends to favor methods that generate their outputs using detection bounding boxes, because this is usually how the ground truth is obtained. Furthermore, the bounding box is not a good approximation of the target when pedestrians appear oblique in the image (due to camera perspective). Instead, we use the Euclidean distance (on the ground plane) between the tracker result and ground truth data in the WCS. The threshold used in our experiments was 50 cm for the PETS dataset and 1 m for the TownCentre sequence. Notice that the threshold for the TownCentre is larger because the

[11] DP+NMS: <http://www.ics.uci.edu/~hpirsiav/>

bounding boxes in the dataset ground truth are not very well aligned with the pedestrians.

The MOT results are presented in Table 4.2. It is possible to observe that, for both datasets, our tracking accuracy (measured by the MOTA value) is higher than all the other methods. Due to the fact that many people appear at the same time throughout the sequence and because there is a light post in the middle of the scene, many occlusions occur in this scenario. We manually count them by visual inspection and identified 28 occlusions that we considered severe and could significantly lead the trackers to failure. Additionally, we counted 16 partial occlusions in which a large portion of target remains still visible. Of this total of 44 occlusions, our method was able to recover a good position for 39 of them. The 5 failures were associated to severe occlusions, and in those cases the targets were quickly identified as lost and removed by our termination scheme.

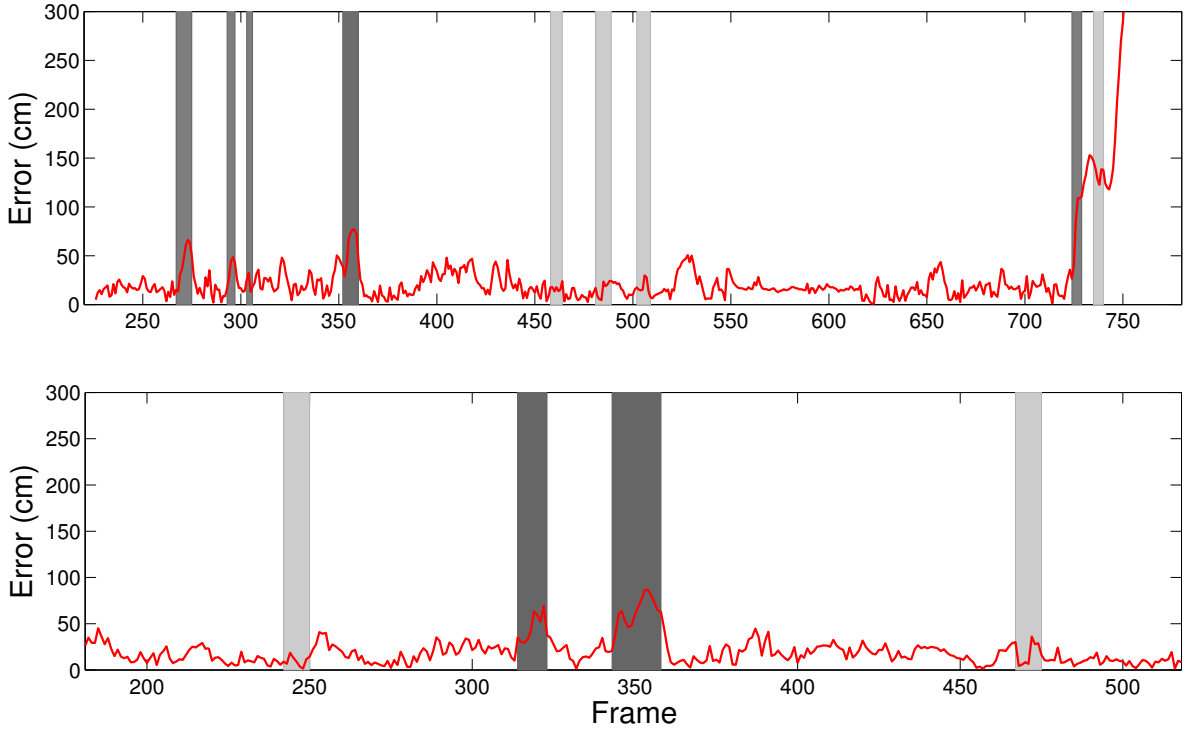
The errors related to the two longest successfully tracked tracks are shown in Figure 4.9, consisting of more than 500 frames for the first track and around 400 frames for the second one. It is also possible to notice in the plot that many occlusions occur along the tracks (shaded regions in the plot) and that, with an exception of one, the method was able to recover good positions afterwards. For the first subject (top row), a severe occlusion around frame 730 caused the tracker to fail. However, within a few frames, the target was terminated and another tracker associated with the same subject was initialized.

Table 4.2 – MOTA and MOTP values of the tested methods for the two sequences involved in the experiments. Best values are shown in bold.

Method	PETS		TownCentre		Observations
	MOTA (%)	MOTP	MOTA (%)	MOTP	
Fragtrack	-45.22	26.03	-22.72	57.71	
TLD	-55.53	24.19	-34.99	59.32	
DP+NMS	58.11	20.81	38.00	56.24	Offline method.
Proposed-NP+ND	-30.64	22.94	-89.22	60.96	W/o prediction and detection.
Proposed-ND	5.77	22.92	-11.91	57.81	W/o detection vectors.
Proposed-NP	54.97	**18.96**	15.47	56.06	W/o prediction.
Proposed-ALL	**59.40**	20.90	**45.20**	**51.8**	With all components.

As observed in (BERNARDIN; STIEFELHAGEN, 2008), the problem of defining the optimal distance threshold for the MOT metric given a dataset remains open. Therefore, we have computed the metrics using different distance thresholds for the PETS dataset, as shown in Figure 4.10. As expected, MOTA values are low for small thresholds, and increase as the threshold is relaxed. It can also be observed that the proposed approach (full version) presents consistently the largest MOTA values as the threshold varies. The MOTP values also increase when the threshold becomes larger, mainly due to the fact that more tracks are considered correct (but their errors get larger, increasing the MOTP values). Two variants of our method

Figure 4.9 – The error curves of two subjects in the PETS dataset. Dark grey highlights correspond to periods of severe occlusion, while light grey highlights are small occlusions.



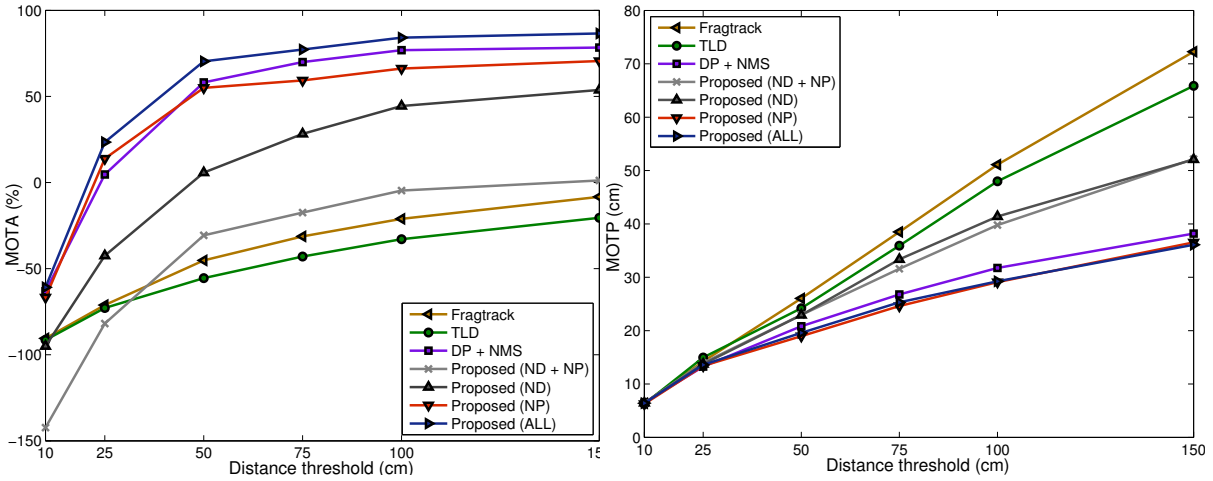
Source: Author

have shown the best (smaller) MOTP values. Not surprisingly, these versions are the ones that include the detection vectors. Hence, as we expected, the use of a pedestrian detection algorithm seems to help the tracker not only to recover from failure, but also to increase tracking accuracy. Additionally, the prediction vectors seem to be useful for enforcing a principal direction to the target based on the previous frames, removing the erratic motions that can arise from simple patch matching.

Two implementations of the method were carried out: one in Matlab and another in C++. The latter version, with pre-computed pedestrian detections and using an implementation of ViBE in GPU, has shown frame-rate values ranging from 3 to 14 frames per second depending on the number of targets that are being tracked (the average frame rate for the PETS dataset was around 4). It is important to point out that the pedestrian detection algorithm used in this work can be greatly accelerated to perform in real-time, as recently presented in (BENENSON et al., 2012). Furthermore, our approach can be parallelized (using GPUs and/or multi-threads), since each target is tracked independently. For the sake of comparison, we measured the average time taken by FragTrack and TLD methods to process a frame while tracking a single person (for all the PETS sequence). As before, we used the implementations provided by the authors (the TLD

[11] The project is open source and available at <https://github.com/gustavofuhr/multi_pedestrian_tracking>

Figure 4.10 – MOTA and MOTP values of all the tested methods for the PETS sequence, varying the distance threshold.



Source: Author

tracker is made in MATLAB and C++, while the FragTrack code is provided in C++). For the TLD tracker, the average time was around $116\ ms$ and for the FragTrack the value was around $496\ ms$. Our method only took around $28\ ms$ in average to process each person at each frame[12]. As for the DP+NMS method, the processing of the entire sequence took several hours since it relies on a computationally expensive method for pedestrian detection.

Finally, for the sake of illustration, some example frames for the PETS sequence are shown in Figure 4.11. It can be observed that the proposed method presents accurate tracking results, being able to keep the same identifier of a target for hundreds of frames despite the relative large inter-frame displacement of the PETS dataset. A video of the results (for Proposed-ALL and PETS dataset) is available in the project website: <http://inf.ufrgs.br/~gfuhr/?file=research/multi-people-tracking>.

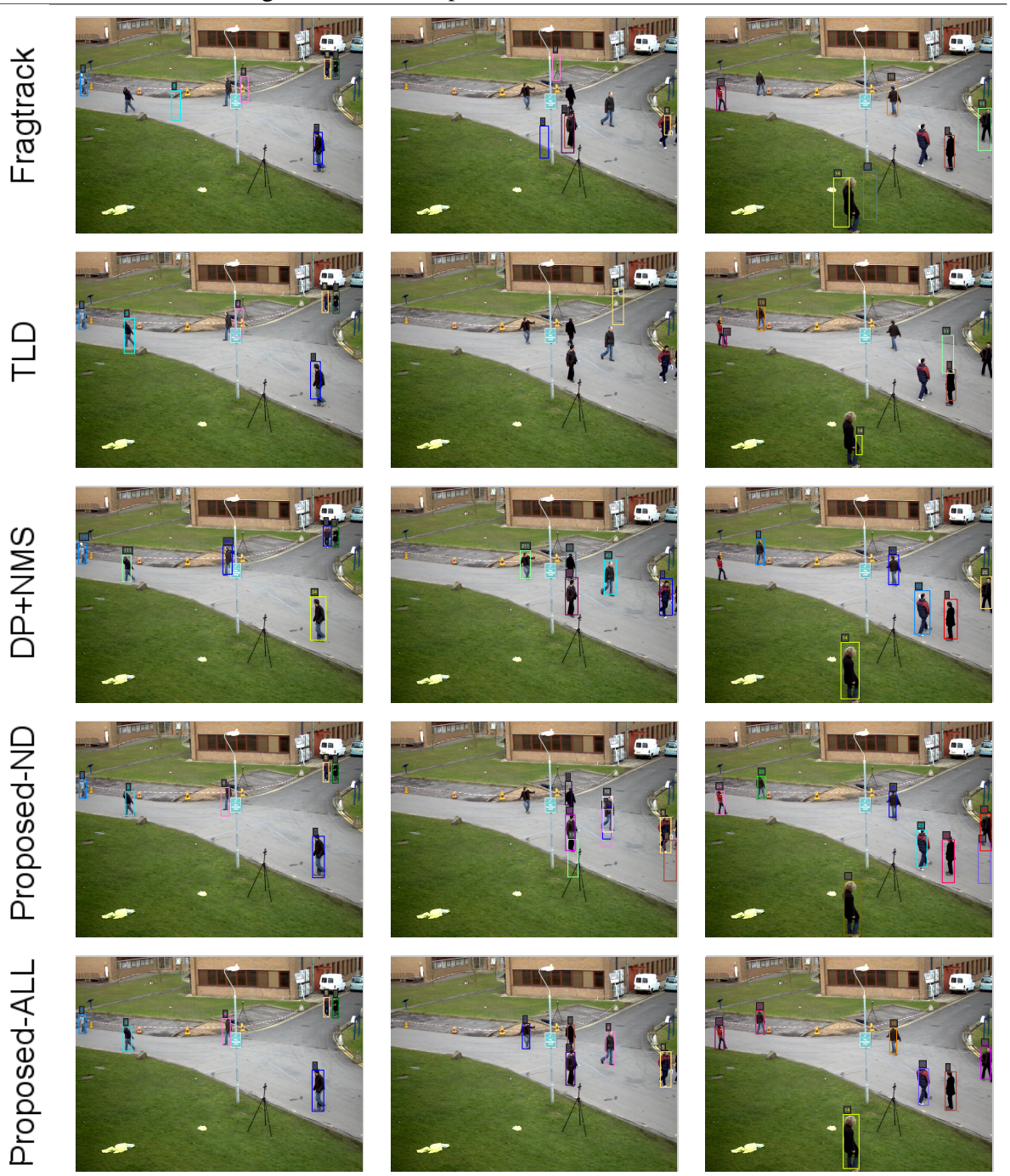
4.2.1 Tracking with self-calibration

To attest the performance of tracking together with our self-calibration results, we tested our tracker with the matrices given by the two variants: *sf:initial-matrix* and *sf:non-linear optimization*. We compare the MOTA and MOTP results with the ground truth calibration and added the results from the tracking-by-detection method DP+NMS (PIRSIAVASH; RAMANAN; FOWLKES, 2011). The reader should recall that this method is non-causal and that does not require calibration. Results are depicted in Figure 4.12.

It is possible to observe that the self-calibration was successfully integrated with the

[12]The computer used in these tests has an Intel Core i7 2700K 3.5GHz processor with 16GB of RAM.

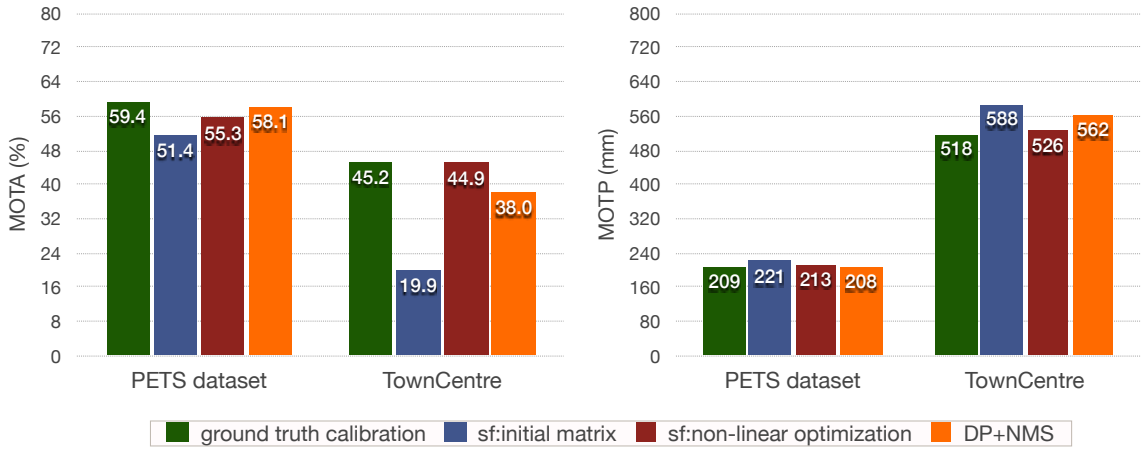
Figure 4.11 – Example frames of the PETS dataset.



Source: Author

tracker. More precisely, our non-linear refinement increases significantly the performance of the tracker in comparison with the initial projection matrix (larger MOTA and smaller MOTP, indicating better results), being close to the ground truth calibration value. Also, the tracking results with our full self-calibration scheme (*sf:non-linear-optimization*) are very close to the non-causal DP+NMS, slightly worse for PETS and better for TownCentre. Moreover, the tracking results for the TownCentre dataset clearly shows the vital role of the optimization step in our framework. We believe that the reason for such results is two fold. First, the initial calibration

Figure 4.12 – MOTA and MOTP values for the tracker proposed with different calibrations and an additional method of the literature. See the text for discussion



Source: Author

matrix for TownCentre presents significantly larger errors than the one from PETS – thus, the optimization phase plays a larger role in the final calibration matrix. This was also observed in the detection experiments of the Towncentre, in Section 4.1. The second aspect comes from the tracking itself. Our tracking approach relies on the calibration and height estimate to correctly track the targets appearing in the sequence. Therefore, if the error in orientation in the Z axis orientation is large (as in *sf:initial-matrix* for TownCentre) the tracker will fail.

4.3 Collective behavior recognition

In the experiments described in this section our goal was to evaluate the accuracy of the interaction and collective behavior classifiers. Additionally, we would like to assess which cues of information are indeed relevant for inferring interactions and collective activities. Since we used Random Forests for the two levels of inference (pairwise interactions and collective activity), each feature in our descriptors is treated individually in the classification phase. For a different classification method, such as SVMs, it would be tricky to verify the importance of each feature dimension. For instance, if one or more values related to a given cue are added to the descriptor, it would be difficult to see if either the distance function, the feature normalization or the information itself is to blame for a possible decrease in classification accuracy. On the contrary, since Random Forests classifiers take the features "as they are" and treat them independently, there is a more direct relation between cue importance and performance, which enables us to see clearly which components of our systems are responsible for the overall performance of the

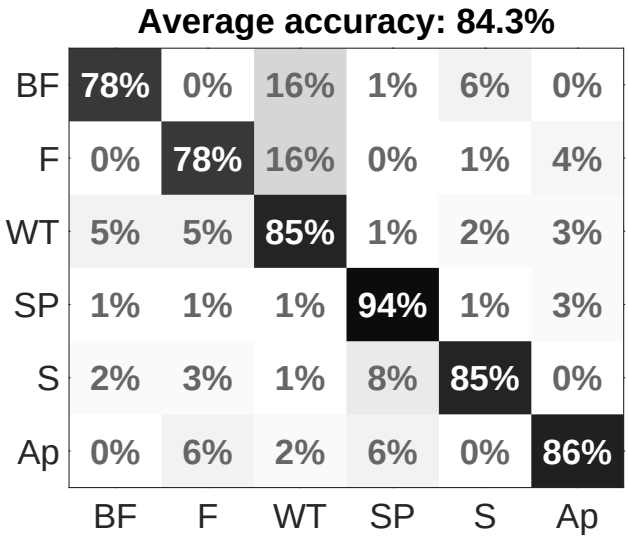
method.

Despite the relatively large number of papers for detecting interactions or collective behavior, there is still no standard experimental framework for these problems. In this work, as mentioned, we chose to use the dataset proposed in (CHOI; SAVARESE, 2012), which has been gaining a lot of popularity in the last years. The dataset consists of 6 different collective behaviors: *Gathering*, *Talking*, *Dismissal*, *Walking*, *Chasing* and *Queuing*. We carefully annotated all the pairwise interactions that appear in the sequences using the six interaction described in Section 3.3.1, making them available together with our source code. Because there is still no standard protocol defined for experiments in collective behavior recognition, we chose to use a LOSO (leave-one-sequence-out) cross validation approach due to the limited amount of samples.

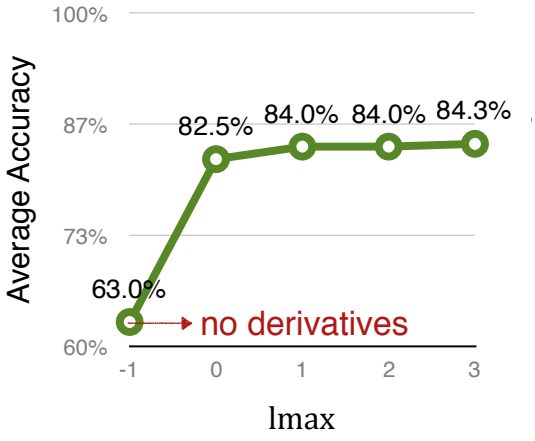
In all experiments, we used $T_1 = T_2 = 64$, which showed a good compromise between robustness to noise and detection lag. The first set of experiments aims to attest if our interaction descriptor (PID) is indeed discriminative despite its very small size (usually less than 20 dimensions). For these experiments we use a step of 5 frames in classification/testing and we defined our standard deviation for the PID at $\sigma_\theta = \pi/8$ (so that each sector in the histogram contains four standard deviations in the orientation) and $\sigma_\rho = 0.25$, based on experiments. We used $K_s = 3$ when generating samples for the KDE, which covers virtually all the area under the Gaussian kernel. To define the number of levels l_{max} used in the speed pyramid, we computed the accuracy for different values as shown in Figure 4.13(b). Based on this plot, we believe that $l_{max} = 1$ presents a good compromise between descriptor size and accuracy, and this value was chosen as the default.

Figure 4.13(a) depicts the confusion matrix of our pairwise interaction detector, with an average per-class accuracy of 84.3% and minimum individual accuracy of 78% for both *following* and *being-followed*. It worth noticing that *following* and *being-Followed* sometimes are classified as *walking-together*, around 16% of times. This is due to noise in the estimates for ground plane position. For example, when a group of several people is walking in two or more rows, we annotated that the subjects in the back rows are following the ones in front of them, even if the whole group is walking as one. When this group is far from the camera, mapping from image to world coordinates gets more sensitive to noisy observations, and the relative positions in our histograms can present jitter. As a consequence, there might be a mix up between front (or back) to bins in the subject to the left or right sides. Despite the fact that the confusion matrix (showed in Figure 4.14(a) shows a frame where is clear that one person is behind another, so the system correctly assigns a *following* interaction to it. On the other hand, if the person is closer to the other subject and in a more diagonal relative position, the systems mistakes the interaction

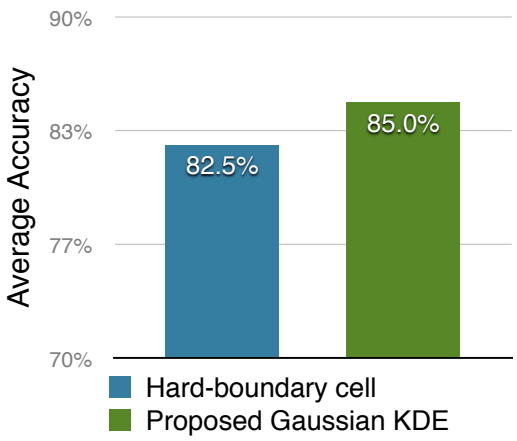
Figure 4.13 – Experimental results for our interaction descriptor. (a) Confusion matrix. (b) Impact of the number of levels in the pyramidal representation. (c) Impact of the KDE-smoothed histogram.



(a) Source: Author



(b) Source: Author

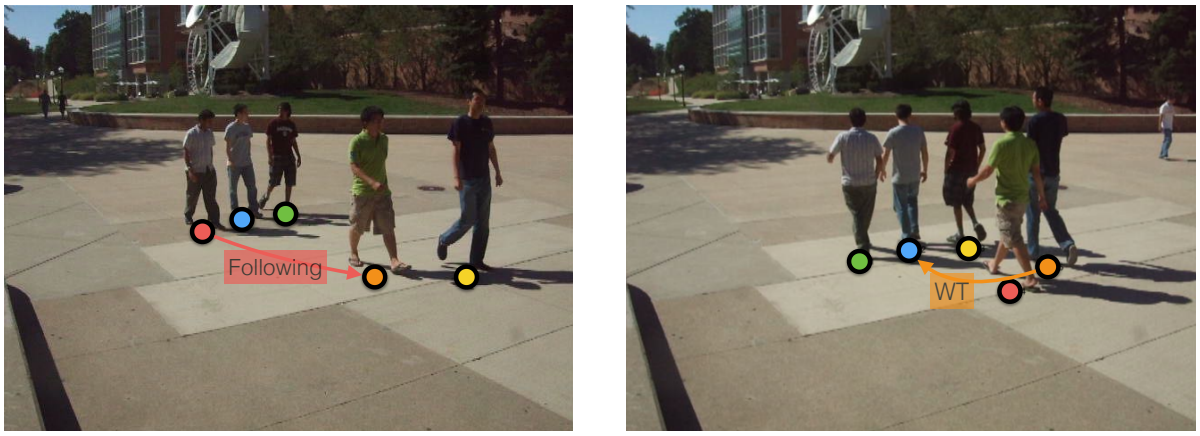


(c) Source: Author

for a *walking-together*, as observed in Figure 4.14(b).

Also, the results of Figure 4.13(b) and 4.13(c) suggest that the use of a Gaussian KDE and pyramidal representation of the relative speed are important to increase the accuracy of the classifier. More specifically, Figure 4.13(b) shows that the pyramid of speeds increases the overall results by around 20% (notice that, in the plot, when $l_{max} = -1$ means that no derivatives were included). The results shown in Figure 4.13(c) indicate that using hard boundaries in the histogram for a reduced number of cells ($\#S_{o_t} = 12$) leads to performance decrease of around 2.5% compared to using our approach inspired on KDE.

Figure 4.14 – Two different frames showing a success and a fail case of our interaction estimation. Circles represent ground plane points projected to the image.



(a) Success case - Seq09, frame #80 (b) Fail case - Seq08, frame #483

Source: Author

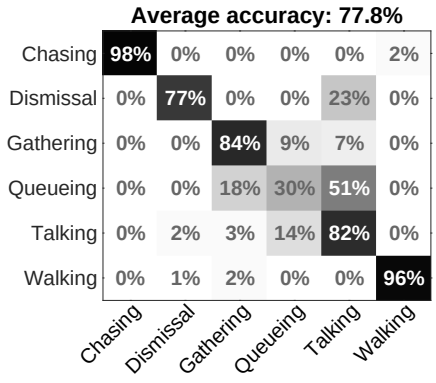
The second set of experiments evaluated the accuracy of the collective behavior recognition method proposed in this dissertation, using the same LOSO cross validation scheme. To evaluate the effect of different cues, we tested three different versions of our method using different sets of cues, as shown in Fig. 4.15. These results show that, despite the CBD descriptor without linear group information (Eq. (3.28)) be able to generate good overall results, the *queuing* and *talking* classes are not well distinguished. The reason is clear, since both present the same profile of pairwise interactions (standing-group), shape dynamics (no increase/decrease) and velocity (near to zero).

It is also important to note that the proposed method is able to identify the role of each pair of pedestrians involved in a given collective behavior due to the hierarchical nature of our approach. Fig. 4.16 illustrates some frames of different detected collective behaviors, along with the corresponding pairwise interactions. For instance, the higher-level chasing event shown in Fig. 4.16(a) is characterized by *following* and *walking-together* pairwise lower-level events.

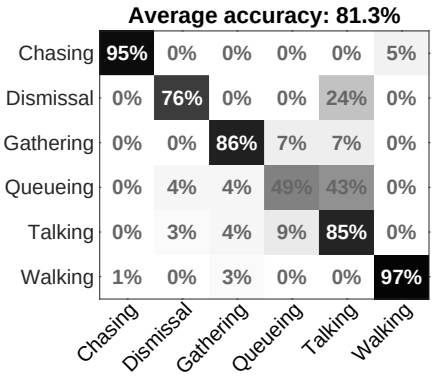
In order to compare our results with other methods, we also ran experiments using the same 3-fold validation proposed by Choi et al. (CHOI; SAVARESE, 2012). Our average accuracy in such configuration was 91%, which is better than state-of-the-art methods (CHOI; SAVARESE, 2012) and (AMER; LEI; TODOROVIC, 2014), which report 79.2% and 87.2%, respectively, using the same protocol. The confusion matrix of (CHOI; SAVARESE, 2012) and ours in this 3-fold validation scheme is presented in Figure 4.17 [13]. It is interesting to note that our lowest per-class accuracy was 80%, compared to 43.5% in (CHOI; SAVARESE, 2012).

[13]The paper of Amera et al. (AMER; LEI; TODOROVIC, 2014) does not present the confusion matrix, only the average precision.

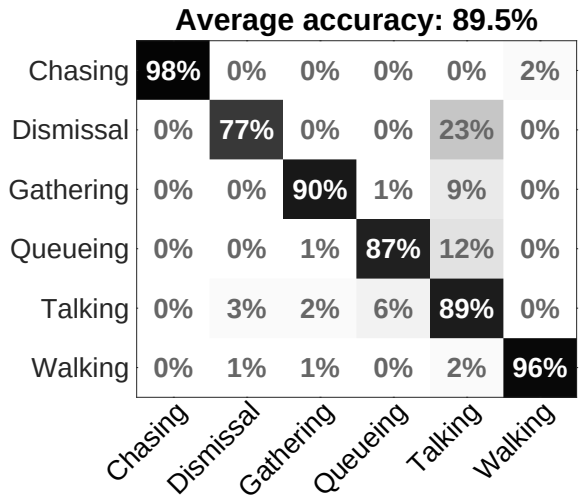
Figure 4.15 – Confusion matrices for the collective behavior method, where “I” indicates the use of interaction histograms, “V” the use of pyramidal mean velocities, “D” the spatial distribution dynamics encoded in (3.27) and “S” the shape encoded from the eigenvalue ratios.



(a) I + V



(b) I + V + D

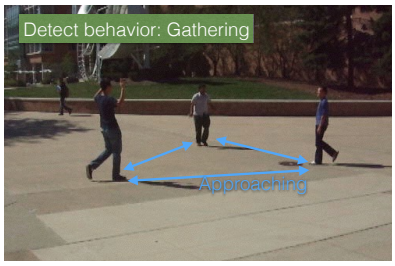


(c) I + V + D + S (full approach)

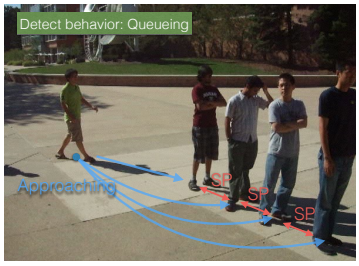
Source: Author

Figure 4.16 – Selected frames from Choi dataset showing interactions and behavior estimates.

(a) Chasing



(b) Gathering



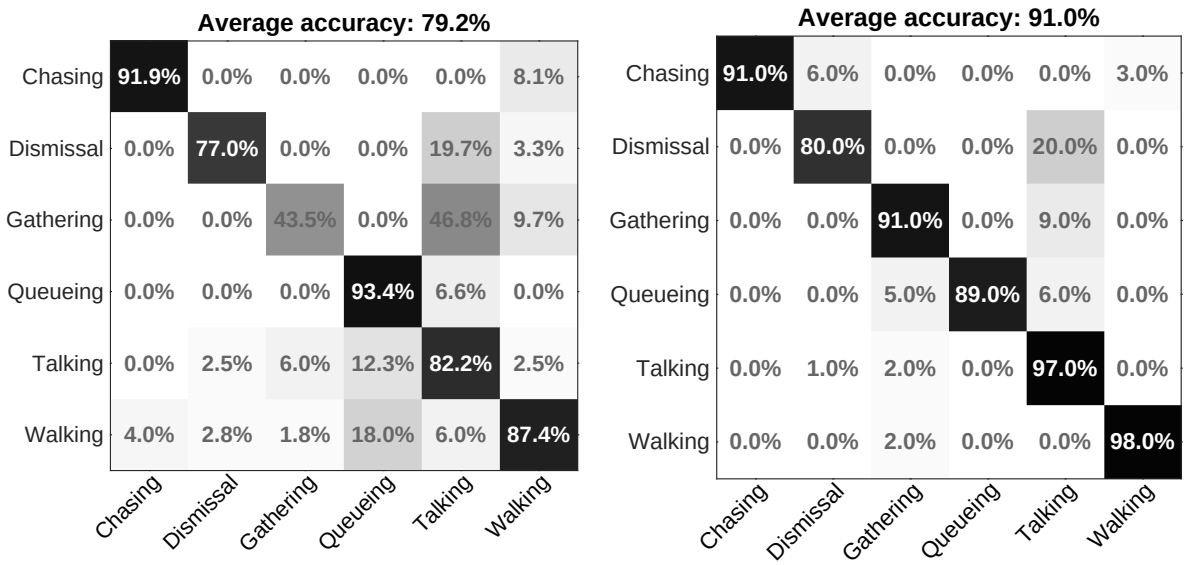
(c) Queueing

Source: Author

Somewhat surprisingly, our average accuracy in the 3-fold cross validation was superior to the leave-one-sequence-out protocol used so far. We believe that the main reason behind this is that

each fold proposed by Choi (CHOI; SAVARESE, 2012) is very well balanced.

Figure 4.17 – Confusion matrices of (CHOI; SAVARESE, 2012) and our proposed method under the 3-fold validation scheme.



(a) Method by Choi (CHOI; SAVARESE, 2012) (b) Ours

Source: Author

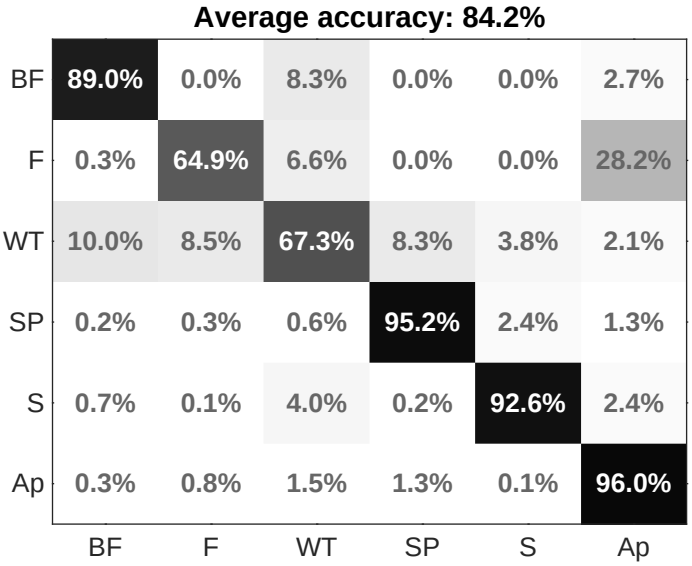
Despite the very good overall and per-class performance, the system is inherently unable to differentiate activities that depend on the subject orientation at static positions and present the same group shape. An example of this is the inability to distinguish events of "waiting to cross a street" and "queuing". However, we believe that being able to generalize to any camera setup is far a better choice for surveillance systems than to use an image-based classifier for subject orientation, which might be crucial to detect just a few specific events.

To investigate the potential of our method in generalizing across different scenarios (and more importantly, different camera setups), we carried out a set of experiments on the BEHAVE dataset (BLUNSDEN; FISHER, 2010). The first 7 sequences of the first set were used, given a total of around 63k frames. The dataset presents almost the same activities of the ones described by Choi et al. (CHOI; SAVARESE, 2012) and used in this dissertation, with the inclusion of *fighting* events and the lack of *queuing* events. Upon analysis, it was clear to us that fighting would not be possible to recognize using solely trajectories – thus, we decide to ignore the frames in which such event appears. Additionally, some sequences presented wrong annotation or no annotation at all. We carefully annotated and corrected the interactions, bounding boxes and collective activities. We made this data publicly available[14] to stimulate a broader application of this dataset in the future.

[14] <https://github.com/gustavofuhr/behave_comp_anno>

Since our goal is to attest how the method deals with different camera setups, this time we did not perform a leave-one-sequence-out approach for this set of experiments. Instead, we performed a cross-dataset validation: train only with the Choi dataset and test with BEHAVE. The interactions were the same as proposed before and the result of classification can be seen in Figure 4.18.

Figure 4.18 – Interaction estimates using only Choi dataset as the training set and Behave as testing.



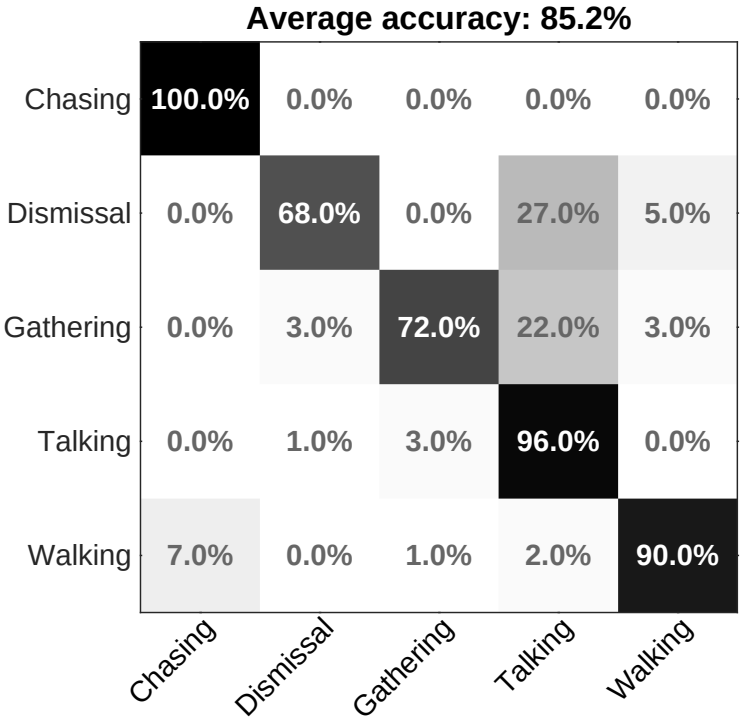
Source: Author

It is worth noting that the interactions in the BEHAVE dataset are mainly *standing pair*, *splitting*, *approaching*, *walking-together* with very few *being-followed* and *following* samples. However, our method has shown an impressive capability of recognizing interaction in new, unseen sequences.

Finally, we apply the same protocol for testing collective behaviors, training with Choi dataset and testing with BEHAVE. The parameters of the proposed method where kept the same, with the exception of removing the PCA-based shape descriptor $p(t)$ defined by Equation (3.28). The reason behind this choice is that for the BEHAVE dataset there is no need to differentiate between standing groups, since there are no *queuing* events. Also, we removed the three sequences from the Choi dataset that were labeled as *Queuing*, so that this class was completely ignored in the experiment. The results can be observed in Figure 4.19.

There results indicate that our method is capable of generalizing for different camera setups in the context of collective behavior well. Moreover, it does not require two much training in terms of variability to correctly estimate the collective behavior. We believe that this comes from the formulation of a two stage approach, which helps the understanding of new instances

Figure 4.19 – Confusion matrix related to cross-dataset validation of collective behavior classification – Choi dataset as the training set and BEHAVE as testing.



Source: Author

of previously trained behaviors. One instance where this is clear is the from *Dismissal* events – and analogous to *Gathering*. In the BEHAVE dataset, these events usually appear as a part of a group leaving the remaining subjects of a standing group. On the other hand, in the Choi dataset, the *Dismissal* events always concern the whole subjects (no subject is left). Despite that, because we based our descriptor on interactions and shape dynamics using ground plane trajectories, we are able to generalize across different exemplars of the same class.

Two sample frames of the BEHAVE dataset are illustrated in Figure 4.20, along with the corresponding interactions and behaviors correctly estimated by our method. Once again, we can see that our method is able to infer the correct information at the different levels of interaction and collective behavior, providing useful information for the intended applications.

Figure 4.20 – Selected frames from BEHAVE dataset showing interactions and collective behavior estimates.

(a) Gathering (b) Dismissal

Source: Author

5 CONCLUSIONS AND FUTURE WORK

As presented in this dissertation, there are several problems related to the task of extracting trajectories and inferring semantic information from surveillance cameras. We proposed in this work several contributions to key elements of this pipeline, namely self-calibration, pedestrian detection and tracking and collective behavior recognition. In fact, instead of focusing on a single problem, we proposed contributions across these four problems by identifying common steps and objectives among them. Moreover, we show how camera calibration can play an essential role to reduce the complexity and increase the performance of such systems.

This dissertation presented a new self-calibration approach for static video cameras based on pedestrian detectors and background removal. It first computes a linear estimation of the projection matrix based on people poles, by extending the approach presented in (LV; ZHAO; NEVATIA, 2002b). Then, a novel non-linear cost function is used to penalize orientation and height errors of re-projected vertical poles, aiming to keep coherence with the expected standing pose of pedestrians in surveillance scenarios. We have also proposed an application of the camera self-calibration method for pedestrian detection. Within the widely used sliding-window framework, we devised a simple approach to create detection candidates that are coherent with the scene geometry. We showed that our self-calibration method significantly improves upon previous work of Lv. et al. (LV; ZHAO; NEVATIA, 2002b) and that our proposed modification for pedestrian detectors increases accuracy and reduces the number of candidates need to be tested. As future work, we also intend to include in our pipeline simpler strategies to compute the initial calibration, since our optimization is capable of greatly improving a given initialization. Additionally, we would like to combine our approach for generating detection candidates to modern detectors based on Convolutional Neural Networks (CNNs), such as the Fast R-CNN (GIRSHICK et al., 2016), to replace costlier approaches such as selective search.

Additionally, this work presented a robust approach to multiple pedestrian tracking using monocular calibrated cameras. Our method explores the motion of independently tracked patches, extracted for each pedestrian, and combines these results with a predicted motion vector and the result of a pedestrian detector in a robust manner using a weighted median filter vector. Our experimental results indicated that the proposed tracker is able to handle short-term occlusions and scale changes. Also, it presented accuracy and precision metrics comparable to (or better than) competitive tracking techniques at near real-time performance. Besides, we show that the results produced by the proposed self-calibration scheme are accurate enough for tracking purposes, with results very close to those obtained using the ground truth calibration. As future

work, we intend to investigate the use of different features in our framework that can help improve tracking performance. Particularly those based on deep learning such as proposed in (BERTINETTO et al., 2016) could easily be adapted to our framework.

Finally, we proposed a novel method to describe and detect interactions and collective activities using only trajectory information in a surveillance scenario. We presented novel compact descriptors (PID and CBD) that combine cues with different natures, which are fed to a two-layered Random Forest to achieve the final classification. Our experimental results showed that the proposed method achieves higher accuracy than competitive approaches. We also showed that our method is able to generalize across different camera setups, due to the use of ground plane trajectories and very small feature vectors. Further work here will concentrate on filtering the pool of interactions to reduce noise in the collective activity recognition. We are also interested in extending the method for different, more complex activities that involve a sequence of collective behaviors, such as pick-pocketing. Also, there is a surprising lack of datasets and protocols for behavior analysis. We made available annotations for a publicly available dataset and we are still interested in generating more sequences with different events/scenarios for activity recognition in the context of video surveillance.

6 APPENDIX

6.1 Accepted Publications

For the sake of completeness, we listed below the papers published as at the time of the writing of this dissertation.

FÜHR, G., JUNG, C. R. **Robust patch-based pedestrian tracking using monocular calibrated cameras**. 25th Conference on Graphics, Patterns and Images (SIBGRAPI), 2012. p.166-173.

Abstract - Although several methods for pedestrian tracking can be found in the literature, robustly tracking a person in unconstrained environments is an open and active research problem. In this paper, we propose a method that represents each pedestrian as a set of multiple fragments, aiming robustness with respect to occlusions. These patches are tracked individually and their translation vectors are combined robustly in the world coordinate frame using Weighted Vector Median Filters (WVMF). Additionally, the algorithm uses the camera parameters to both estimate the person scale in a straightforward manner and to limit the search region used to track each fragment. Experiments carried out using two publicly available datasets (PETS and TownCentre) are presented, and they indicate that the proposed method is robust to partial occlusions and large scale changes. According to our experiments, the proposed approach outperforms, regarding the quality of localization, some of the methods in the current state of the art.

FÜHR, G., JUNG, C. R. **Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras**. Pattern Recognition Letters (Special Issue), 2013, 39, p.11-20.

Abstract - This paper presents a new approach for tracking multiple people in monocular calibrated cameras combining patch matching and pedestrian detection. Initially, background removal and pedestrian detection are used in conjunction with the vertical standing hypothesis to initialize the targets with multiples patches. In the tracking step, each patch related to a given target is matched individually

across frames, and their translation vectors are combined robustly with pedestrian detection results in the world coordinate frame using weighted vector median filters. Additionally, the algorithm uses the camera parameters to both estimate the person scale in a straightforward manner and to limit the search region used to track each fragment. Our experimental results indicate that our tracker can deal with occlusions and video sequences with strong appearance variations, presenting results comparable to or better than existing state-of-the-art algorithms.

FÜHR, G., JUNG, C. R. **Camera self-calibration based on non-linear opti- mization and applications in surveillance systems**. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2015.

Abstract - This paper presents a new approach for self-calibration of static cameras in the context of surveillance applications. Initially, a pedestrian detector is applied and the responses are validated using background removal. Then, foreground-related pixels within the detection results are used to estimate the feet-head line segments of each person (called poles), which are used to find a linear estimate for the camera matrix. Finally, a non-linear cost function is used to refine the initial estimate, aiming to improve mostly the orientation of the re-projected poles. We also present different applications of self-calibration in tasks related to video surveillance itself, such as improvements to pedestrian detection and tracking algorithms, and augmented reality applications, such as the insertion of virtual cameras to aid the placement of real cameras in the scene.

FÜHR, G., DE PAULA, M. B, JUNG, C. R. **On the use of calibration for pedestrian detection in on-board vehicular cameras**. Conference on Graphics, Patterns and Images (SIBGRAPI), 29, 2016.

Abstract - This paper presents a new approach for pedestrian detection in the context of Driver Assistance Systems (DAS). Given a camera with known intrinsic parameters, a flexible online calibration scheme that explores the expected road geometry is used to obtain the extrinsic parameters. With the full camera parameters, the expected geometry and size of a standing person is used to customize a baseline pedestrian detector based on sliding windows and multiple scales. Our experimental

results show that the proposed approach allows the use of detachable cameras in the context of DAS, improving the accuracy of the baseline pedestrian detector. Furthermore, the flexible calibration scheme allows to estimate the distance from detected pedestrians to the camera using detachable cameras, opposed to the fixed onboard cameras in commercial vehicles that support vision-based DAS.

6.2 Submitted for Publication

FÜHR, G., JUNG, C. R. **From Pairwise Interactions to Collective Behavior Recognition Using Layered Random Forests**.

Abstract - This paper presents a novel hierarchical approach for collective behavior detection based on trajectories. In the first layer, we introduce a novel feature called Personal Interaction Description (PID), which combines the spatial distribution of a pair of pedestrians within a temporal window with a pyramidal representation of the relative speed to detect pairwise interactions. These interactions are then combined with higher level features related to the mean speed and shape formed by the pedestrians in the scene, generating a Collective Behavior Descriptor (CBD) that is used to identify collective behaviors in a second stage. In both layers, Random Forests were used as classifiers, since they allow features of different natures to be combined seamlessly. Our experimental results indicate that the proposed method achieves better results than state of the art techniques in benchmarked datasets, being also capable of identifying the role of each pedestrian in the detected collective behavior.

6.3 Resumo estendido

Nas últimas décadas, o número de câmeras distribuídas em ambientes internos e externos aumentou de maneira significativa. Porém, a enorme quantidade de dados capturados por estas câmeras ainda é processado, em grande parte, de maneira manual. Especificamente na área de vigilância, se mostra clara a necessidade de um sistema inteligente capaz de analisar e extrair informações semânticas das sequências de vídeo. Essa necessidade favoreceu o surgimento de várias propostas da comunidade de Visão Computacional nos últimos anos.

O escopo das aplicações de sistemas de vigilância inteligente é bastante amplo (HAER-

ING; VENETIANER; LIPTON, 2008). Muitas são referentes a segurança, como aplicações voltadas para detecção de intrusos (LIM; TANG; CHAN, 2014), objetos abandonados (FERRYMAN et al., 2013) entre outros; além disso, existem aplicações como controle de tráfego (XIA et al., 2016), sinopse de vídeo (RAV-ACHA; PRITCH; PELEG, 2006; LEE; GRAUMAN, 2015) e até análise de público para mercados de varejo (DENMAN et al., 2012).

Esta tese tem como objetivo fornecer uma solução robusta para os problemas de rastreamento de pedestres e análise de comportamento coletivo em sequências de vigilância. Mais precisamente, deseja-se extrair as trajetórias de todos os pedestres presentes na sequência, as interações entre eles (por exemplo, um par está se aproximando, andando junto, etc.) e a atividade global da cena (encontro de pessoas, perseguição, etc.). O *pipeline* proposto (Fig. 1.3) é inicializado com a calibração da câmera. Nossa hipótese é que, em sistemas de vigilância, onde a câmera se mantém estática por (ao menos) um período razoável de tempo, a calibração deve simplificar e adicionar informações relevantes aos problemas de rastreamento e detecção de eventos coletivos, reduzindo a dependência de uma configuração específica de câmera.

No Seção 3.1 nós propomos uma técnica de calibragem de câmera sem nenhuma intervenção manual. O algoritmo utiliza detecção de pedestres e segmentação de fundo para detectar pontos de fuga e a linha do horizonte da câmera. A matriz de projeção da câmera é estimada utilizando uma abordagem composta por dois estágios: inicialização e otimização não-linear. Adicionalmente, nós mostramos como utilizar a calibração de câmera para simplificar a detecção de pedestres. Dois diferentes métodos de detecção de pessoas da literatura (DALAL; TRIGGS, 2005; DOLLÁR et al., 2009) são modificados para criar candidatos coerentes com a geometria da cena. Resultados mostram que nossa proposta reduz o número de falsos positivos e ao mesmo tempo reduz significativamente o número de candidatos e o tempo de execução.

Adicionalmente, nós propomos a utilização de calibração em rastreamento de pedestres (Seção 3.2). Mais detalhadamente, nós desenvolvemos um novo *framework* para rastreamento de pedestres que combina informações de diferentes origens em coordenandas de mundo utilizando *weighted median filter vectors*. Nossa técnica de rastreamento divide os pedestres em segmentos, rastreando-os individualmente, para aumentar a robustez a oclusões. Além disso, informações de predição de movimento e detecção de pedestres são combinadas de maneira causal. Experimentos em sequências de vídeo popularmente utilizada na literatura mostram que nosso método consegue rastrear pedestres de maneira robusta, atingindo performance perto de tempo real.

Dadas as trajetórias extraídas no plano do sistema de coordenadas do mundo, nós propomos um método capaz de extrair o tipo de interação entre um par de pessoas além da atividade apresentada por um grupo. O processo é realizado em duas etapas subsequentes utilizando

Random Forests como a ferramenta de classificação. Na primeira etapa, um histograma das distâncias relativas para cada par de pessoas é construído. A definição dos limites deste histograma é baseada no trabalho psicológico de Hall (HALL, 1973) que identifica níveis de interação que pessoas apresentam de acordo com a distância entre as mesmas. Além disso, nós extraímos a velocidade relativa do par, com o objetivo de identificar se eles estão se aproximando, se afastando ou mantendo uma velocidade constante. Depois de realizada a classificação de interações par a par, nós fornecemos estas informações junto com fatores relativos a dinâmica da disposição do grupo e velocidade média para a segunda camada de classificação. Em experimentos em dois datasets comumente utilizadas na literatura nós demonstramos que nosso método é capaz de distinguir atividades com alta acurácia e consegue generalizar para diferentes disposições de câmera.

REFERENCES

ADAM, A.; RIVLIN, E.; SHIMSHONI, I. Robust fragments-based tracking using the integral histogram. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2006. v. 1, p. 798–805.

AMER, M. R.; LEI, P.; TODOROVIC, S. Hirf: Hierarchical random field for collective activity recognition in videos. In: **European Conference on Computer Vision**. [S.l.]: Springer, 2014. p. 572–585.

ARROYO, R. et al. Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls. **Expert Systems with Applications**, Elsevier, v. 42, n. 21, p. 7991–8005, 2015.

ASTOLA, J.; HAAVISTO, P.; NEUVO, Y. Vector median filters. **Proceedings of the IEEE**, IEEE, v. 78, n. 4, p. 678–689, 1990.

BAE, S.-H.; YOON, K.-J. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: **Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2014. p. 1218–1225.

BARNICH, O.; DROOGENBROECK, M. V. Vibe: A universal background subtraction algorithm for video sequences. **IEEE Transactions on Image Processing**, v. 20, n. 6, p. 1709–1724, 2011.

BENENSON, R. et al. Pedestrian detection at 100 frames per second. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2012. p. 2903–2910.

BENFOLD, B.; REID, I. Stable multi-target tracking in real-time surveillance video. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2011. p. 3457–3464.

BERNARDIN, K.; STIEFELHAGEN, R. Evaluating multiple object tracking performance: the CLEAR MOT metrics. **EURASIP Journal on Image and Video Processing**, v. 2008, p. 246309, jan. 2008. ISSN 1687-5176.

BERTINETTO, L. et al. Fully-convolutional siamese networks for object tracking. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2016. p. 850–865.

BLUNSDEN, S.; FISHER, R. The behave video dataset: ground truthed video for multi-person behavior classification. **Annals of the BMVA**, British Machine Vision Association, v. 4, n. 1-12, p. 4, 2010.

BORGES, P. V. K.; CONCI, N.; CAVALLARO, A. Video-based human behavior understanding: a survey. **Circuits and Systems for Video Technology, IEEE Transactions on**, IEEE, v. 23, n. 11, p. 1993–2008, 2013.

BOUMA, H. et al. Automatic detection of suspicious behavior of pickpockets with track-based features in a shopping mall. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **SPIE Security+ Defence**. [S.l.], 2014.

BREITENSTEIN, M. et al. Online multiperson tracking-by-detection from a single, uncalibrated camera. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 33, n. 9, p. 1820 –1833, 2011.

BROUWERS, G. M. et al. Automatic calibration of stationary surveillance cameras in the wild. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2016. p. 743–759.

CHAKRABORTY, I.; CHENG, H.; JAVED, O. 3d visual proxemics: Recognizing human interactions in 3d from a single image. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2013. p. 3406–3413.

CHANG, X.; ZHENG, W.-S.; ZHANG, J. Learning person–person interaction in collective activity recognition. **Image Processing, IEEE Transactions on**, IEEE, v. 24, n. 6, p. 1905–1918, 2015.

CHENG, Z. et al. Recognizing human group action by layered model with multiple cues. **Neurocomputing**, Elsevier, 2014.

CHOI, W. Near-online multi-target tracking with aggregated local flow descriptor. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2015. p. 3029–3037.

CHOI, W.; SAVARESE, S. Multiple target tracking in world coordinate with single, minimally calibrated camera. In: **Proceedings of the 11th European conference on Computer vision: Part IV**. [S.l.: s.n.], 2010. p. 553–567.

CHOI, W.; SAVARESE, S. A unified framework for multi-target tracking and collective activity recognition. In: **European Conference on Computer Vision**. [S.l.: s.n.], 2012. p. 215–230.

CHOI, W.; SAVARESE, S. Understanding collective activitiesof people from videos. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 36, n. 6, p. 1242–1257, 2014.

CHOI, W.; SHAHID, K.; SAVARESE, S. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In: **International Conference on Computer Vision Workshops**. [S.l.: s.n.], 2009. p. 1282–1289.

CHU, H. et al. A heat-map-based algorithm for recognizing group activities in videos. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 1, p. 1–8, 2012.

CIPOLLA, R.; DRUMMOND, T.; ROBERTSON, D. P. Camera calibration from vanishing points in image of architectural scenes. In: **BMVC**. [S.l.: s.n.], 1999. v. 99, p. 382–391.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2005. v. 1, p. 886–893.

DENG, Z. et al. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2016. p. 4772–4781.

DENMAN, S. et al. Identifying customer behaviour and dwell time using soft biometrics. In: **Video Analytics for Business Intelligence**. [S.l.]: Springer, 2012. p. 199–238.

DIHL, L. L.; JUNG, C. R.; BINS, J. C. Robust adaptive patch-based object tracking using weighted vector median filters. In: **Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2011. p. 149–156.

DOLLÁR, P.; BELONGIE, S.; PERONA, P. The fastest pedestrian detector in the west. In: **British Machine Vision Conference**. [S.l.: s.n.], 2010. p. 68.1–68.11.

DOLLÁR, P. et al. Integral channel features. In: **British Machine Vision Conference**. [S.l.: s.n.], 2009. p. 91.1–91.11.

DOLLAR, P. et al. Pedestrian detection: An evaluation of the state of the art. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 34, n. 4, p. 743–761, 2012.

DOUXCHAMPS, D.; CHIHARA, K. High-accuracy and robust localization of large control markers for geometric camera calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 31, p. 376–383, 2009.

ENZWEILER, M.; GAVRILA, D. Monocular pedestrian detection: Survey and experiments. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 31, n. 12, p. 2179 –2195, 2009.

FAGOT-BOUQUET, L. et al. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2016. p. 774–790.

FENG, L.; BHANU, B. Understanding dynamic social grouping behaviors of pedestrians. **IEEE Journal of Selected Topics in Signal Processing**, v. 9, p. 317–329, 2015.

FERRYMAN, J.; ELLIS, A. Pets2010: Dataset and challenge. In: **IEEE Advanced Video and Signal-Based Surveillance**. [S.l.: s.n.], 2010. p. 143–150.

FERRYMAN, J. et al. Robust abandoned object detection integrating wide area visual surveillance and social context. **Pattern Recognition Letters**, Elsevier, v. 34, n. 7, p. 789–798, 2013.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, ACM, v. 24, n. 6, p. 381–395, 1981.

FLETCHER, R. **MODIFIED MARQUARDT SUBROUTINE FOR NON-LINEAR LEAST SQUARES.** [S.l.], 1971.

FLEURET, F. et al. Multi-camera people tracking with a probabilistic occupancy map. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 30, n. 2, p. 267–282, 2008.

FOUHEY, D. F. et al. People watching: Human actions as a cue for single view geometry. **International Journal of Computer Vision**, Springer, v. 110, n. 3, p. 259–274, 2014.

FÜHR, G.; JUNG, C. Camera self-calibration based on non-linear optimization and applications in surveillance systems. **Circuits and Systems for Video Technology, IEEE Transactions on**, 2015.

FÜHR, G.; JUNG, C. R. Robust patch-based pedestrian tracking using monocular calibrated cameras. In: **Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2012. p. 166 –173.

FÜHR, G.; JUNG, C. R. Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras. **Pattern Recognition Letters (Special Issue)**, v. 39, p. 11–20, 2014.

FÜHR, G.; JUNG, C. R.; PAULA, M. B. de. On the use of calibration for pedestrian detection in on-board vehicular cameras. v. 29, 2016.

GALL, J. et al. Optimization and filtering for human motion capture - a multi-layer framework. **International Journal of Computer Vision**, Springer, v. 87, n. 1, p. 75–92, 2010.

GE, W.; COLLINS, R. T.; RUBACK, R. B. Vision-based analysis of small groups in pedestrian crowds. **Pattern Analysis and Machine Intelligence**, v. 34, n. 5, p. 1003–1016, 2012.

GEIGER, A.; LENZ, P.; URTASUN, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE. **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**. [S.l.], 2012. p. 3354–3361.

GIRSHICK, R. et al. Region-based convolutional networks for accurate object detection and segmentation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 38, n. 1, p. 142–158, 2016.

GREEN, M. W. **The appropriate and effective use of security technologies in U.S. schools: a guide for schools and law enforcement agencies.** [s.n.], 2005. Available from Internet: <http://www.osti.gov/scitech/servlets/purl/974410>.

GUAN, J. et al. Extrinsic calibration of camera networks based on pedestrians. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 5, p. 654, 2016.

HAERING, N.; VENETIANER, P. L.; LIPTON, A. The evolution of video surveillance: an overview. **Machine Vision and Applications**, Springer, v. 19, n. 5-6, p. 279–290, 2008.

HALL, E. T. **The silent language.** [S.l.]: Anchor, 1973.

HAN, J.; KAMBER, M. **Data Mining: concepts and techniques**. [S.l.]: Morgan Kaufmann, 2001.

HARITAOGLU, I.; HARWOOD, D.; DAVIS, L. W4 s: A real-time system for detecting and tracking people in 2 1/2d. In: **European Conference on Computer Vision**. [S.l.: s.n.], 1998. v. 1406, p. 877–892.

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. [S.l.]: Cambridge Univ Press, 2000.

HELD, D.; THRUN, S.; SAVARESE, S. Learning to track at 100 fps with deep regression networks. **arXiv preprint arXiv:1604.01802**, 2016.

HOIEM, D.; EFROS, A. A.; HEBERT, M. Putting objects in perspective. **International Journal of Computer Vision**, v. 80, n. 1, p. 3–15, 2008.

HUANG, S. et al. Camera calibration from periodic motion of a pedestrian. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2016. p. 3025–3033.

HUGHES, G. On the mean accuracy of statistical pattern recognizers. **IEEE Transactions on Information Theory**, v. 14, n. 1, p. 55–63, 1968.

HUIS, J. R. v. et al. Track-based event recognition in a realistic crowded environment. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **SPIE Security+ Defence**. [S.l.], 2014. p. 92530E–92530E.

HWANG, J.; LAY, S.; LIPPMAN, A. Nonparametric multivariate density estimation: a comparative study. v. 42, n. 10, p. 2795–2810, 1994.

IBRAHIM, M. S. et al. A hierarchical deep temporal model for group activity recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2016. p. 1971–1980.

JIANG, N.; TAN, P.; CHEONG, L.-F. Symmetric architecture modeling with a single image. **ACM Transactions on Graphics**, v. 28, n. 5, p. 113:1–113:8, dec. 2009.

JR, J. C. S. J. et al. Understanding people motion in video sequences using voronoi diagrams. **Pattern Analysis and Applications**, Springer, v. 10, n. 4, p. 321–332, 2007.

JUNEJO, I.; FOROOSH, H. Robust auto-calibration from pedestrians. In: **IEEE International Conference on Video and Signal Based Surveillance**. [s.n.], 2006. p. 92–97. ISBN 0-7695-2688-8. Available from Internet: <http://dx.doi.org/10.1109/AVSS.2006.99>.

JUNEJO, I. N.; FOROOSH, H. Trajectory rectification and path modeling for video surveillance. In: IEEE. **IEEE International Conference on Computer Vision**. [S.l.], 2007. p. 1–7.

KALAL, Z.; MATAS, J.; MIKOLAJCZYK, K. Pn learning: Bootstrapping binary classifiers by structural constraints. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2010. p. 49–56.

KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-learning-detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 7, p. 1409 –1422, 2012.

KANHERE, N. K.; BIRCHFIELD, S. T. A taxonomy and analysis of camera calibration methods for traffic monitoring applications. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 11, n. 2, p. 441–452, 2010.

KIM, J.-S.; KWEON, I. S. Camera calibration based on arbitrary parallelograms. **Computer Vision and Image Understanding**, Elsevier, v. 113, n. 1, p. 1–10, 2009.

KRAHNSTOEVER, N.; MENDONçA, P. R. Bayesian autocalibration for surveillance. In: **International Conference on Computer Vision**. [S.l.: s.n.], 2005. v. 2, p. 1858–1865.

KUSAKUNNIRAN, W.; LI, H.; ZHANG, J. A direct method to self-calibrate a surveillance camera by observing a walking pedestrian. In: IEEE. **Digital Image Computing: Techniques and Applications, 2009. DICTA'09.** [S.l.], 2009. p. 250–255.

LAGARIAS, J. C. et al. Convergence properties of the nelder–mead simplex method in low dimensions. **SIAM Journal on Optimization**, SIAM, v. 9, n. 1, p. 112–147, 1998.

LAVIOLA, J. Double exponential smoothing: an alternative to kalman filter-based predictive tracking. In: **Proceedings of the Workshop on Virtual Environments**. [S.l.: s.n.], 2003. p. 199–206.

LEE, Y. J.; GRAUMAN, K. Predicting important objects for egocentric video summarization. **International Journal of Computer Vision**, Springer, v. 114, n. 1, p. 38–55, 2015.

LEPETIT, V.; FUA, P. Monocular model-based 3d tracking of rigid objects: A survey. **Foundations and trends in computer graphics and vision**, v. 1, n. CVLAB-ARTICLE-2005-002, p. 1–89, 2005.

LI, R.; CHELLAPPA, R.; ZHOU, S. K. Recognizing interactive group activities using temporal interaction matrices and their riemannian statistics. **International journal of computer vision**, Springer, v. 101, n. 2, p. 305–328, 2013.

LIM, M. K.; TANG, S.; CHAN, C. S. isurveillance: Intelligent framework for multiple events detection in surveillance videos. **Expert Systems with Applications**, Elsevier, v. 41, n. 10, p. 4704–4715, 2014.

LIPTON, A. J. Keynote: intelligent video as a force multiplier for crime detection and prevention. In: IET. **Imaging for Crime Detection and Prevention, 2005. ICDP 2005. The IEE International Symposium on**. [S.l.], 2005. p. 151–156.

LIU, J.; COLLINS, R. T.; LIU, Y. Robust autocalibration for a surveillance camera network. In: IEEE. **Applications of Computer Vision (WACV), 2013 IEEE Workshop on**. [S.l.], 2013. p. 433–440.

LIU, W. et al. Leveraging long-term predictions and online learning in agent-based multiple person tracking. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 25, p. 399–410, 2015.

LV, F.; ZHAO, T.; NEVATIA, R. Self-calibration of a camera from video of a walking human. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2002. v. 1, p. 562 – 567 vol.1. ISSN 1051-4651.

LV, F.; ZHAO, T.; NEVATIA, R. Self-calibration of a camera from video of a walking human. In: **International Conference on Pattern Recognition**. [S.l.: s.n.], 2002. v. 1, p. 562–567.

LV, F.; ZHAO, T.; NEVATIA, R. Camera calibration from video of a walking human. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE Computer Society, Washington, DC, USA, v. 28, n. 9, p. 1513–1518, 2006. ISSN 0162-8828. Available from Internet: <http://dx.doi.org/10.1109/TPAMI.2006.178>.

MCPHAIL, C.; WOHLSTEIN, R. T. Using film to analyze pedestrian behavior. **Sociological Methods & Research**, v. 10, n. 3, p. 347–375, 1982.

MICUSIK, B.; PAJDLA, T. Simultaneous surveillance camera calibration and foot-head homology estimation from human detections. In: **Computer Vision and Patter Recognition**. [S.l.]: IEEE, 2010. p. 1562–1569.

MILAN, A. et al. Mot16: A benchmark for multi-object tracking. **arXiv preprint arXiv:1603.00831**, 2016.

OLIVER, N. M.; ROSARIO, B.; PENTLAND, A. P. A bayesian computer vision system for modeling human interactions. **Pattern Analysis and Machine Intelligence**, v. 22, n. 8, p. 831–843, 2000.

ORGHIDAN, R. et al. Camera calibration using two or three vanishing points. In: CITESEER. **Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), Wrocław, Poland**. [S.l.], 2012. p. 9–12.

PAULA, M. B. de; JUNG, C. R.; SILVEIRA, L. da. Automatic on-the-fly extrinsic camera calibration of onboard vehicular cameras. **Expert Systems with Applications**, Elsevier, v. 41, n. 4, p. 1997–2007, 2014.

PIRSIAVASH, H.; RAMANAN, D.; FOWLKES, C. C. Globally-optimal greedy algorithms for tracking a variable number of objects. In: **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2011. p. 1201–1208.

QURESHI, F. Z.; TERZOPOULOS, D. Surveillance in virtual reality: System design and multi-camera control. In: IEEE. **IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2007. p. 1–8.

RAV-ACHA, A.; PRITCH, Y.; PELEG, S. Making a long video short: Dynamic video synopsis. In: IEEE. **2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)**. [S.l.], 2006. v. 1, p. 435–441.

SADEGHIAN ALEXANDRE ALAHI, S. S. A. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. **arXiv preprint arXiv:1701.01909**, 2017.

SEBE, I. O. et al. 3d video surveillance with augmented virtual environments. In: ACM. **First ACM SIGMM international workshop on Video surveillance**. [S.l.], 2003. p. 107–112.

SOLERA, F.; CALDERARA, S.; CUCCHIARA, R. Socially constrained structural learning for groups detection in crowd. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2015.

TAJ, M.; CAVALLARO, A. Recognizing interactions in video. In: **Intelligent Multimedia Analysis for Security Applications**. [S.l.]: Springer, 2010. p. 29–57.

TAYLOR, G.; MAI, F. Behind the scenes: What moving targets reveal about static scene geometry. In: **Proceedings of the IEEE International Conference on Computer Vision Workshops**. [S.l.: s.n.], 2013. p. 546–553.

TERAMOTO, H.; XU, G. Camera calibration by a single image of balls: From conics to the absolute conic. In: **Proc. of 5th Asian Conference on Computer Vision**. [S.l.: s.n.], 2002. p. 499–506.

TSAI, R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. **IEEE Journal of Robotics and Automation**, v. 3, n. 4, p. 323–344, 1987.

UIJLINGS, J. R. et al. Selective search for object recognition. **International journal of computer vision**, Springer, v. 104, n. 2, p. 154–171, 2013.

WANG, B. et al. Tracklet association by online target-specific metric learning and coherent dynamics estimation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, 2016.

WANG, X. Intelligent multi-camera video surveillance: A review. **Pattern recognition letters**, Elsevier, v. 34, n. 1, p. 3–19, 2013.

XIA, Y. et al. Towards improving quality of video-based vehicle counting method for traffic flow estimation. **Signal Processing**, Elsevier, v. 120, p. 672–681, 2016.

XIANG, Y.; ALAHI, A.; SAVARESE, S. Learning to track: Online multi-object tracking by decision making. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2015. p. 4705–4713.

YILMAZ, A.; JAVED, O.; SHAH, M. Object tracking: A survey. **ACM CSUR**, ACM, v. 38, n. 4, p. 1–45, 2006.

YU, X. et al. Automatic camera calibration of broadcast tennis video with applications to 3D virtual content insertion and ball detection and tracking. **Computer Vision and Image Understanding**, v. 113, p. 643–652, May 2009.

ZHANG, S. et al. How far are we from solving pedestrian detection? In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2016. p. 1259–1267.

ZHANG, S. et al. Multi-target tracking by learning local-to-global trajectory models. **Pattern Recognition**, v. 48, n. 2, p. 580–590, 2015.

ZHANG, Z. A flexible new technique for camera calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 22, n. 11, p. 1330–1334, 2000.

ZHANG, Z. Camera calibration with one-dimensional objects. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 26, n. 7, p. 892–899, 2004.

ZHANG, Z. et al. Practical camera calibration from moving objects for traffic scene surveillance. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 23, n. 3, p. 518–533, 2013.