UNIVERSIDADE DA CORUÑA

Facultade de Informática
Departamento de Computación

PhD Thesis

Understanding target trajectory behavior: a dynamic scene modeling approach

Brais Cancela Barizo
March 2015

PhD advisors:
Marcos Ortega Hortas
Manuel F. González Penedo

March 2015
Universidade da Coruña

Facultade de Informática
Campus de Elviña s/n
15071, A Coruña (Spain)

Copyright notice:

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording and/or otherwise without the prior permission of the authors.

To my family

Acknowledgments

I would like to thank Manuel F. González Penedo for introducing me to both the research world and your research group. Thank you for letting me do things my way after my mental block with the tracking models. Marcos Ortega Hortas, thank you for your smart advices and your multiple explanations about the difference between *this* and *these*! Unfortunately, some things will never be corrected.

What to say about the VARPA lab? You were listening my complaints about everything, not matter what. So, you are welcome! Jokes aside, it was a pleasure to work with all of you, even if nobody knows what I was doing till this year. I am sorry, I cannot stop saying stupid jokes with you. Special thanks to Noelia for being our 'troubleshooter'. You were always willing to help, making my life easier.

I also want to express my gratitude to Professor Shaogang Gong and to Doctor Timothy Hospedales. Thank you for allowing me to enjoy a research stay in Queen Mary University of London (UK), working with two of the most recognized specialists in scene understanding. Thank you to the Queen Mary Computer Vision Laboratory members for your warm welcome, and for the wonderful experience of working together. And of course, I want to mention the great memories that I have of the three months I spent there, and all the people that I had the opportunity to meet. Next time, I will teach you my basketball skills!

Last but far from least is a heartfelt thank you to my family, my couple and all my friends for putting up with me all these years. I am not good at sharing my feelings, but you know how I feel.

After all, it was fun.

"Today's scientists have substituted mathematics for experiments, and they wander off through equation after equation, and eventually build a structure which has no relation to reality."

Nikola Tesla

Abstract

Human behavior analysis is one of the most active computer vision research fields. As the number of cameras are increased, especially in restricted environments, like airports, train stations or museums, the need of automatic systems that can catalog the information provided by the cameras becomes crucial. In the case of crowded scenes, it is very difficult to distinguish people behavior because of the lack of visual contact of the whole body. Thus, behavior analysis remains in the evaluation of trajectories, adding high-level knowledge approaches in order to use that information in several applications like video surveillance or traffic analysis.

The proposal of this research is the design of a fully-automatic human behavior system from a distance. On the one hand, two different multiple-target tracking methods and a target re-identification procedure are presented to detect every target in the scene, returning their trajectories as output. On the other hand, a novel behavior analysis system, which includes information about the environment, is provided. It is based in the idea that every person tries to reach a goal in the scene following the same path the majority of people should use. An extremely fast abnormal behavior metric is presented, providing our method with the capabilities needed to be used in real-time scenarios.

Resumen

El análisis de comportamiento humano es uno de los campos más activos en la rama de visión por computador. Con el incremento de cámaras, especialmente en entornos controlados tales como aeropuertos, estaciones de tren o museos, se hace cada vez más necesario el uso de sistemas automáticos que puedan catalogar la información proporcionada. En el caso de entornos concurridos, es muy difícil el poder distinguir el comportamiento de personas en base a sus gestos, debido a la falta de visión de su cuerpo al completo. Por ende, el análisis de comportamiento se realiza en base a sus trayectorias, añadiendo técnicas de razonamiento de alto nivel para ulilizar dicha información en múltiples aplicaciones, tales como la video vigilancia o el análisis de tráfico.

El propósito de esta investigación es el desarrollo de un sistema totalmente automático para el análisis de comportamiento de las personas. Por una parte, se presentan dos sistemas para el seguimiento de múltiples objetivos, así como un sistema novedoso para la re-identificación de personas, con la intención de detectar todo objeto de interés en la escena, devolviendo sus trayectorias como salida. Por otra parte, se presenta un sistema novedoso para el análisis de comportamiento basado en información del entorno de la escena. Está basado en la idea que que toda persona, cuando intenta llegar a un cierto lugar, tiende a seguir el mismo camino que suele utilizar la mayoría de la gente. Se presentan una serie de métricas para la detección de movimientos anómalos, haciendo que este método sea ideal para su utilización en sistemas de tiempo real.

Resumo

A análise do comportamento humano é un dos campos máis activos na rama da visión por computadora. Co incremento de cámaras, especialmente en entornos controlados tales coma aeroportos, estacións de tren ou museos, faise cada vez máis necesario o uso de sistemas automáticos que poidan catalogar a información proporcionada. No caso de entornos concurridos, é moi complicado de poder distinguir o comportamento de persoas dacordo cos seus xestos, debido á falta dunha visión completa do corpo do suxeito. Por tanto, a análise de comportamento tende a realizarse en base á traxectoria, engadindo técnicas de razoamento de alto nivel para ulilizar dita información en diversas aplicacións, tales coma a video vixiancia ou a análise de tráfico.

O propósito desta investigación é o desenrolo dun sistema totalmente automático para a análise do comportamento das persoas. Por unha parte, preséntanse dous sistemas para o seguimento de múltiples obxectivos, así coma un sistema novidoso para a re-identificación de persoas, coa intención de detectar todo obxecto de interés na escena, devolvendo as traxectorias asociadas como saída. Por outra parte, preséntase un sistema novidoso para a análise de comportamente baseada na información do entorno da escena. Está baseado na idea de que toda persoa, cando intenta acadar un certo luegar, tende a seguir o mesmo camiño que xeralmente usa a maioría da xente. Preséntanse unha serie de métricas para a detección de movementos anómalos, facendo posible que este método poida ser utilizado en sistemas de tempo real.

Contents

Chapter 1

Introduction

Behaviour analysis is one of the most active research fields. The key idea is to develop an automatic system that can catalog every action any target is doing. A target is any object in a scene that has to be followed. Depending on the behaviour analysis problem, the type of target could vary. Furthermore, the kind of action depends on the kind of behaviour the system tries to search, either involving individual actions (such as walking, running, jogging . . .) or grouping events (having a meeting, leaving a group, fighting . . .).

Every automatic behaviour analysis system can, ideally, be divided in three different techniques, as depicted in Fig. 1.1:

- *Detection:* having a video sequence as input, the detection block should be able to detect every target at every frame in the scene. Computer vision techniques are used to solve this issue.
- *Tracking:* using the detection block information, the tracking block should assign label identifications to each detected target in the video. That means grouping all the detections that belong to the same target into an unique label, following them until the video has finished or the target has left the scene.

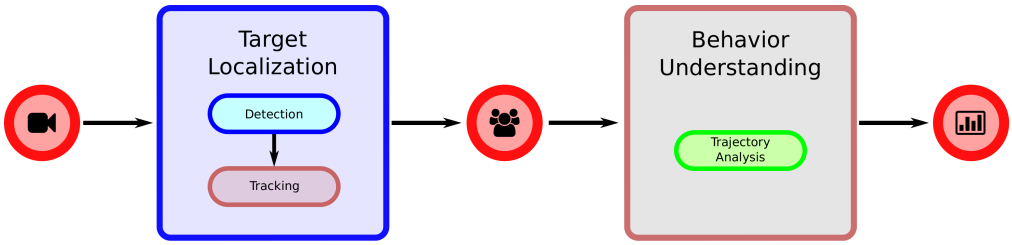


Figure 1.1: Behavior analysis framework structure.

- *High-Level Behaviour Analysis:* the detection information, along with the tracking identification, is used to catalog the behaviour of each target. Pattern recognition techniques are used to classify every action.

Furthermore, the system architecture highly depends on the number of targets to take into account at the same time. To this end, behaviour analysis systems are divided into two big groups:

- *Action Recognition:* systems are focused to catalog individual actions. These kind of techniques describe the behaviour of isolated people. Detection and tracking systems are focused to only detect one target in the scene. Camera are placed near the target, causing the detection system to obtain a detailed silhouette of the target. The high-level analysis is done by taking into account both spatial and temporal information. Only a small number of actions are detected, according to recent surveys in the topic (Turaga, Chellappa, Subrahmanian, & Udrea, 2008; Poppe, 2010; Weinland, Ronfard, & Boyer, 2011). The explanation of these models is out of the scope of this thesis.

- *Crowd Analysis:* systems try to infer behaviour between groups of targets. This is a more complex discipline both in target detection and tracking. Cameras are usually placed far from the floor, in order to give a better perspective to the movements of the targets.

This thesis is focused in the development of a novel behaviour analysis technique from a distance, that is, using crowd analysis. Thus, a more detailed explanation of the state of the art of the three big blocks is included below. For more information, there is also recent surveys in the topic (Zhan, Monekosso, Remagnino, Velastin, & Xu, 2008; Candamo, Shreve, Goldgof, Sapper, & Kasturi, 2010; Popoola & Wang, 2012).

1.1 Detection System

In a scene with multiple targets, it is necessary to have a balance between accuracy and velocity. In this section a brief explanation of the classic multiple-target detection systems is provided. Note that more recent techniques were developed, but they are essentially variants of these big three groups.

1.1.1 Background Subtraction

This is the most basic technique. Having a video frame, instead of trying to detect the foreground pixels that belongs to a target, the model does exactly the opposite: it detects all the pixels that belongs to the background. Thus, all the pixels that are not recognized as background should be, after removing the presence of noise, foreground pixels. After that, a blob detection technique is used to detect every different target in the scene.

To model the background, a training step must be previously defined. In early attempts, the background was modelled by simply choosing its maximum and minimum intensity value (Haritaoglu, Harwood, & Davis, 1998, 2000). However, this technique requires the training images not to have any possible target that may distort the result. Thus, a simple statistical approach can be used by modelling every pixel scene as a Gaussian distribution. Assuming the system is trained using only a background sequence, the Gaussian model performs a good approach of the background. Hence, pixel values that are considered as outliers by the Gaussian distribution are marked as *foreground*.

Unfortunately, illumination changes and artifacts provided by the video compression introduce some error in the result, due to the 'simplicity' of the proposed model. To solve that, Stauffer and Grimson (Stauffer & Grimson, 1999) store the background as a Mixture of Gaussians (MoG). A bunch of Gaussians, are used to define each background pixel. If a pixel value does not fit in the combination of the background distributions, it is considered as foreground until it is included into the model with enough evidence. Different color spaces can be used (for instance, L*a*b obtains good results under sudden illumination changes. The method is updated every time step, so no train step is needed. Unfortunately, this update causes a bad estimation if the tracking object stops. After a few iterations, the tracking object will be set as part of the background. The same problem arises in (Jepson, Fleet, & El-Maraghi, 2003), where a recursive update of a Gaussian Mixture is used to track human faces. It can cope with partial occlusions, but fails when dealing with both total occlusions and collisions, since it is implemented to track isolated people.

Horprasert et al. (Horprasert, Hardwood, & Davis, 2000) used a method based on a color background-subtraction. Two different parameters, chromaticity and brightness, are computed and normalized for each pixel by using linear combinations of the RGB color-space. Using four different thresholds, a mask image indicates whether a pixel is part of the background or the foreground. It can also detect shadow pixels caused by the targets, which are often removed. Unfortunately, this technique requires the tuning of a parameter that is very dependent of the lighting

condition, which varies in every scene. Thus, a bad calibration of this threshold introduces a lot of noise.

In a similar way, Kim et al. (Kim, Chalidabhongse, Harwood, & Davis, 2005) use a codebook. Each background pixel consists of one or more codewords. The brightness, frequency of occurrence, the longest interval during the training period that the codeword has not occurred, and the first and last access time are stored into each codeword. Foreground pixels are detected by testing the difference between the current image and the codebook. This method can be used in a dynamic or a static way. Doshi and Trivedi (Doshi & Trivedi, 2006) combines the codebook model with the shadow suppression in HSV-color space (Cucchiara, Grana, Piccardi, Prati, & Sirotti, 2001) to improve the methodology.

Maddalena and Petrosino (Maddalena & Petrosino, 2008) use a self-organizing map to model the background. Each background pixel is represented as a set of $n \times n$ weight vectors, typically 3. The method obtains good results, but the processing time is too high to be used in real-time systems.

1.1.2 Optical Flow

The problem with the background subtraction is that it is very difficult to segment a pair of targets when they are overlapped. Whereas this is not a problem when dealing with controlled scenarios, it is a crucial point when dealing with crowded scenes, when every target could be merged into an unique blob. Thus, a different technique must be used.

Instead of trying to define the background, optic flow techniques try to detect moving pixels within the scene. Thus, new information is included in each foreground pixel: velocity and orientation of the motion.

Lucas and Kanade (Lucas, Kanade, et al., 1981) developed the first optic flow algorithm to be used in vision systems. It is based in the idea that the variation of the target position in a video scene is relatively small. Thus, it it possible to locate the moving target by looking for the same point of interest (corner, feature, ...) near the last time that point of interest was previously detected. To control the algorithm complexity, the point of interest between frames is computed within a small region centered in the feature previous position. Thus, the algorithm fails when target movement are too quick, causing the motion to happen far from the local region where to look after it. To control this problem, a new version was developed by Bouguet (Bouguet, 2001) using a pyramidal cascade.

Several different approaches were developed to detect motion into an image. For instance, by using the Fourier domain (Shizawa & Maze, 1991; Heeger, 1988), the

local image phase (Fleet & Jepson, 1990) or, more recently, tensor voting (Min & Medioni, 2008; Rashwan, García, & Puig, 2013). In a different way Farnebäck (Farnebäck, 2003) detect the image structure as a tensor, providing a 2^{nd} order degree polynomial to describe it.

Although there still exist interesting works dealing with dense optic flow techniques (Zimmer, Bruhn, & Weickert, 2011), a number of points of interest are usually chosen before, in order to decrease the computational cost of the model. In that sense, the work of Shi and Tomasi (Shi & Tomasi, 1994) is the most used. However, recently techniques are also achieving promising results (Brox & Malik, 2011).

1.1.3 High Level Approaches

The main idea into the high-level systems is to add information a priori about the objects of interest. Contrary to background subtraction and optic flow techniques, high-level approaches have the ability to distinguish between different types of targets. In particular, when dealing with people, information about the human shape is searched. For instance, Dalal and Triggs (Dalal & Triggs, 2005) introduced the Histograms of Oriented Gradients (HoGs), which are used to train a Support Vector Machine (SVM) for each part of the body. A similar idea was also used in (Desai, Ramanan, & Fowlkes, 2009; Felzenszwalb, Girshick, McAllester, & Ramanan, 2010) to train any object in the scene. However, these methods try to detect every part in the body, many of which are occluded in crowded scenes.

To solve this, Li et al. (M. Li, Zhang, Huang, & Tan, 2008, 2009) simplified the method, trying to locate only the omega shape created by the head and the shoulders. A HOG feature based SVM is used to confirm every target previously located using a Viola-Jones type classifier (Viola & Jones, 2001), which improves the speed of the algorithm. Using the same omega shape detection, Rodriguez et al. (Rodriguez, Sivic, Laptev, & Audibert, 2011) improve the detection in crowded scenes including a density estimation parameter. Unlike Li approach, this model requires the computation of the HoG descriptor in every pixel image, which increases the computational cost.

1.2 Tracking System

Once every target is detected within the frame, a tracking system has to assign every new target detection with its target identification. Although this idea seems to be simple, it is very complicated to solve it when dealing with crowded scenes. Target occlusions or people crossing each other multiples times are just a couple of

examples that often causes the system to either lose a target or to switch target identifications. In this section an explanation of different state-of-the-art tracking methods is provided.

1.2.1 Appearance Approaches

An appearance approach tries to model each target by considering color, texture information, or a combination of both descriptors. This techniques are often used in the re-identification problem: the ability to recognize whether two different detections belongs to the same target or not. This is one of the most challenging fields in the computer vision, since all the image problems appear in it: target pose and local illumination changes. The ability to develop a good re-identification technique is crucial to obtain a robust multiple-target tracking.

For instance, Collins et al. (Collins, Liu, & Leordeanu, 2005) approach is based in a pool of 49 different histograms, using different combinations into the RGB-color space. The technique selects a rectangular regions surrounding the target and divide it into two big regions, foreground and background. A log-likelihood ratio between the foreground and the background region is used to select a smaller pool of best features. Finally, this small set is used to determine whether a new detection belong to a previously tracked target or not. Additionally, a pool of historical 'best features' is maintained to use them in cases of object occlusion recovery. This method obtains good results under homogeneous backgrounds with constant brightness, failing when the duration of the occlusion is high, causing the best pool of histograms to be changed if the background highly varies.

1.2.2 Kalman Filter

The Kalman filter (Kalman, 1960) is a linear quadratic estimator that, using information about the target stored along time as input, predicts posterior target estimations. The Kalman filter works under noise information, producing optimal predictions of possible new states of the variables involved in the filter.

Assume a input vector z_k. The Kalman filter needs to store both the conditional mean of the input $\hat{x}_k$, the conditional covariance P_k, and the input covariance R_k to model the state system, defined by

$$x_{k+1} = A_{k+1}x_k + B_{k+1}u_{k+1} + w_k, \tag{1.1}$$

where A_{k+1} is the state transition model, B_{k+1} is the control input model, u_{k+1} the control vector (which if often avoided), and $w_k \sim \mathcal{N}(0, Q_k)$ is the noise vector (Q_k is the covariance matrix of the input). To translate the state vector to the input

vector space, the observation model matrix H_k is used. The model works in two different steps, the *prediction step* and the *updating step*.

Prediction Step

In this step, the new state measurement is obtained by solving the equation

$$x_{k+1} = A_{k+1}\hat{x}_k + B_{k+1}u_{k+1}, \tag{1.2}$$

the predicted estimation is defined as

$$\tilde{z}_{k+1} = H_k x_{k+1}, \tag{1.3}$$

and the predicted estimate conditional covariance

$$\tilde{P}_{k+1} = A_{k+1}P_k A_{k+1}^T + Q_k. \tag{1.4}$$

Updating step

Once we now the real estimation z_{k+1}, the model is updated following these equations: first, the estimation error is computed

$$\tilde{y}_{k+1} = z_{k+1} - \tilde{z}_{k+1}. \tag{1.5}$$

The, the state measurement is updated by solving the equation

$$\hat{x}_{k+1} = \hat{x}_k + K_k\tilde{y}_{k+1}, \tag{1.6}$$

where

$$K_k = \tilde{P}_{k+1}H_k^T S_k^{-1}, \tag{1.7}$$

and

$$S_k = H_k\tilde{P}_{k+1}H_k^T + R_k. \tag{1.8}$$

Furthermore, the conditional covariance is updated by solving

$$P_{k+1} = (I - K_k H_k)\tilde{P}_{k+1}. \tag{1.9}$$

The difficulty of this method is to find an optimal Q_k and R_k covariance matrices, since updating them over each step will increase the computational cost. This filter is often used in tracking systems to predict the position of each target in the scene (Black, Ellis, & Rosin, 2002; Mittal & Davis, 2003; Iwase & Saito, 2004; Magee, 2004). However, the model fails when a target is lost during a long period, because a linear prediction technique cannot deal with the uncertainty produced by the human movement, that is, sudden changes in the direction of in the speed.

1.2.3 Particle Filter

The Kalman filter, along with any other linear prediction method, is able to predict the position under regular movements. However, this kind of algorithm only provides one unique prediction. To solve that, it is desirable to throw multiple approximated positions, assigning different probabilities to each one depending on how close it is to the real target position.

That is the key point of the particle filter algorithm. It is a technique for implementing recursive Bayesian filter by Monte Carlo sampling. In mathematical terms, the idea is to represent the posterior density by a set of random particles with associated weights, and then the estimates are computed based on these samples and weights.

Suppose we have a target that we need to track along the scene. The particle filter algorithm for visual tracking follows these steps:

1. Initialize the state vector of the linear prediction technique used (e.g. Kalman) and get an appearance model for the target.
2. Generate a set of N particles.
3. For each new frame
   (a) Find the predicted state of each particle using the state equation and get an appearance model for the predicted position.
   (b) Compute the distance between the predicted and the target appearance model.
   (c) Weight each particle based on similarity between appearance models.
   (d) Select the state of the target based on the weighted particles (mean, maximum value, ...).
   (e) Sample the particles for next iteration.

Contrary to the Kalman filter, this is a more robust algorithm, since it can compute several different positions, including the one predicted by the Kalman filter. It is widely used in tracking systems (Gustafsson et al., 2002; S. K. Zhou, Chellappa, & Moghaddam, 2004; Breitenstein, Reichlin, Leibe, Koller-Meier, & Van Gool, 2011; Hlinka, Sluciak, Hlawatsch, Djuric, & Rupp, 2012). It can be viewed as a generalization of the linear prediction models. However, the problem of this technique is the computational cost: $\mathcal{O}(N)$, where N is the number of particles. A small number causes the system to behave similar to the linear filter chosen. On the contrary, a high number makes the model useless in real-time applications.

1.2.4 Associated-based Tracking

An associated-based tracking formulates the tracking by joining trajectory segments until a complete trajectory is defined. The segments are often obtained by low-level trackers, such as the Kalman filter. This kind of tracked are the most used nowadays, being the most studied. The development of this technique is quite recent.

In first place, a selection of *tracklets* is performed (Bose, Wang, & Grimson, 2007; Xing, Ai, & Lao, 2009; Song, Jeng, Staudt, & Roy-Chowdhury, 2010). A *tracklet* is defined as a set of detections that, with a high probability, belongs to the same target. Given a sequence, the detection system detects a set of different possible targets per frame. This detection system is often a high-level approach detector which produces object hypotheses as observations for data association, focusing on an specific kind of target, such as vehicles or pedestrians (Luo, Zhao, & Kim, 2014). Then, a low-level tracker is conducted to obtain the probability that two different detections belong to the same target.

Finally, *tracklets* are combined to find complete trajectories. Multiple techniques were used to deal with this problem. For instance, Shitrit et al. (Ben Shitrit, Berclaz, Fleuret, & Fua, 2011) formulated the associated-based tracking as a global optimization problem. All trajectories are obtained by maximizing the model energy. In an opposite way, Andriyenko and Schindler (Andriyenko & Schindler, 2011; Andriyenko, Schindler, & Roth, 2012) use a similar scheme. They infer the most usual path for every target, performing the matching by minimizing the energy related to each trajectory.

Yang and Nevatia (Yang, Huang, & Nevatia, 2011; ?, ?) model the energy by using a Conditional Random Field (CRF). An ad-hoc minimization is used to find the optimum energy. A probabilistic graph is also used by Benfold and Reid (Benfold & Reid, 2011), who introduced a Markov Chain Monte Carlo Data Association to estimate the most probable trajectories.

1.2.5 Re-Identification

As previously explained, tracking approaches are based in assigning the same identification to detections that belongs to the same person, until the person leaves the scene. However, if the target reappears in the scene again, it is possible to recover the previous identification? This is the key point of the re-identification procedures. The ability to re-identificate a person, no matter where it appears.

The classic re-identification field can be viewed as a retrieval problem: given a predefined 'gallery' set of known individuals, systems try to label each new 'probe' detection with the identity of the matching gallery individual. The problem is

significantly simpler, because it can be divided into a series of independent tasks: 'For each probe person, find the top most similar in the gallery'.

Contrary to the tracking approach, spatial information is no longer useful to deal with this problem. Thus, the appearance information is crucial. Studies have investigated good feature representations (Farenzena, Bazzani, Perina, Murino, & Cristani, 2010) and discriminative models (Hirzer, Roth, Köstinger, & Bischof, 2012; Tao, Jin, Wang, Yuan, & Li, 2013) to maximise the chance of correct matching. They considered the contexts of single-shot (Prosser, Zheng, Gong, Xiang, & Mary, 2010; Zheng, Gong, & Xiang, 2011) (one image per person per camera) as well as multi-shot (Bialkowski, Denman, Lucey, Sridharan, & Fookes, 2012; Karaman & Bagdanov, 2012) (a series of images per person per camera, obtained from tracking) scenarios. More information can be obtained in recent books (Gong, Cristani, Yan, & Loy, 2014) or surveys (Vezzani, Baltieri, & Cucchiara, 2013) on the topic.

1.3 Trajectory Behavior Analysis

Once every target in the scene is tracked, the information of its trajectory is used by a high-level layer which determines its behavior. Although there exist different approaches in the literature, it can be divided into two big groups: *clustering-based behavior analysis* and *social force models*.

1.3.1 Clustering-based Approaches

These techniques make use of classic machine learning procedures. The premise is simple: 'usual' trajectories can be grouped into a limited number of clusters. Thus, trajectories that do not fit within any of these clusters are marked as abnormal. The different techniques relies on how the so-called 'usual' paths can be obtained.

Three different path models are used in the state-of-the art:

Centroid: The most usual and simple, which were used in early approaches to the topic (Hu, Xiao, Xie, Tan, & Maybank, 2004; Naftel & Khalid, 2006; Hu, Xie, Fu, Zeng, & Maybank, 2007) . Each cluster is described as one unique trajectory (a sequence of points). Several clustering techniques are used to obtain the initials 'usual' paths: Hybrid (Karypis, Han, & Kumar, 1999), agglomerative (Buzan, Sclaroff, & Kollios, 2004), where we merge clusters until we obtain the desired number; divisive (Biliotti, Antonini, & Thiran, 2005), Graph-based (X. Li, Hu, & Hu, 2006), Spectral (Hu et al., 2007) or direct (B. Morris & Trivedi, 2008), using techniques such as k-means or fuzzy c means.

(a) (b)

Figure 1.2: These trajectories are defined as similar using classic distance measure techniques, while including the scene information the red routes are clearly abnormal.

Envelope: The variation of each cluster is included when modeling it. Two common representations are usually chosen: extremal points path (Makris & Ellis, 2005; Wang, Tieu, & Grimson, 2006) or Gaussian distribution representation (B. T. Morris & Trivedi, 2008; Wang, Ma, & Grimson, 2009; Wang, Ma, Ng, & Grimson, 2011; B. T. Morris & Trivedi, 2011; Xu, Gong, & Hospedales, 2013).

Sub-paths: Instead of modeling complete paths, multiple segments are used to define partial trajectories. Then, transitional probabilities between them are used to model the complete paths (Piciarelli & Foresti, 2006; Bashir, Khokhar, & Schonfeld, 2007).

Once the 'usual' paths are modeled, a metric is necessary to determine whether a new trajectory fits within a cluster or not. The most simple one is the euclidean distance. However, these methods obtain poor results, requiring trajectories with the same size to be compared. In (Keogh & Pazzani, 2000), Keogh et al. presented the Dynamic Time Warping (DTW) technique. Basically, this method tries to find a time warping that minimizes the distance between two different trajectories. It can be used with trajectories with different sizes. Buzan et al. (Buzan et al., 2004) introduced a similar idea, the Longest Common Subsequence (LCSS). It can also be used with unequal length data, becoming more robust to noise. The reason is that not all the trajectory points need to be matched. Similar to these methods, Piciarelli and Foresti (PF) (Piciarelli & Foresti, 2006) uses a dynamic time warping window, which is increased along time, that is, the maximum error allowed is low at the starting trajectory point, becoming larger while we are reaching the end. The performance of these metrics were tested in (B. Morris & Trivedi, 2009), achieving good results.

The problem with these metrics is that they do not take into account information about the environment. For instance, let's consider a highway. Imagine two trajectories that start in the same position, having similar ending points (only differs a few), like in Fig. 1.2-(b). Although every metric distance determines all routes are similar, the red one is clearly abnormal since it crosses the central reservation. Analogue, two parallel routes, like in Fig. 1.2-(a) may result similar without any context. However, in the highway, one of them is behaving contrary to the traffic motion.

1.3.2 Social Force Models

They are based in the idea that some stimuli, like the scene properties and other people interactions, affect the pedestrian trajectory (Helbing, Farkas, & Vicsek, 2000), (Burstedde, Klauck, Schadschneider, & Zittartz, 2001). This approximation is often used in computer graphic schemes, developing a set of different *forces* that are added to infer the new movement (Treuille, Cooper, & Popović, 2006), (Reynolds, 2006), (van den Berg, Patil, Sewall, Manocha, & Lin, 2008). The main drawback of these techniques is that although they are good approaches to model the usual human behavior, there exist infinite solutions to model a normal behavior. Thus, how can we use these kind of systems to decide whether a trajectory is abnormal or not? Furthermore, these systems can distinguish high-level behaviours, but in a general way. They take the complete scene as a whole. No individual information about each pedestrian is provided.

More recently, techniques that try to merge computer vision techniques with social models are arising. For instance, some works introduce the *social force* model to detect abnormal behavior (Mehran, Oyama, & Shah, 2009), (Pellegrini, Ess, Schindler, & Van Gool, 2009). Flow models were also included to predict crowd behavior (Moore, Ali, Mehran, & Shah, 2011).

1.4 Thesis

Based in the information provided in this introduction, the purpose of this thesis is to develop a full-automatic behavior analysis system in crowded scenes from a distance. As there still exist problems in each behavior analysis module, the specific objectives of this thesis are the following:

1. Introduce new methodologies for multiple-target tracking that can solve, totally or partially, the main issues related to these systems: that is, the relation between accuracy and computational cost.

2. Explanation of what is defined as a pedestrian 'usual' behavior, according to trajectory analysis.

3. Definition, based in the previous explanation, of a novel pedestrian trajectory analysis model that can distinguish whether a person is having an abnormal behavior or not by introducing information about the environment.

4. Improvement of the model to be able to deal with a more general object, making the algorithm suitable to be used with any kind of target.

5. Improvement of the model in terms of computational cost, enabling the model to be used in crowded scenes.

To create a fully-automatic behaviour analysis system it is needed to deal with all the question arose in previous sections. None of the state-of-the-art approaches in each step have completely solved its problematic. To that end, the proposal of this research is to design an automatic system to perform a human behaviour analysis. Contrary to state-or-the-art techniques, the proposed framework is based in a set of human behaviour hypothesis that are evaluated by the experimental results.

As it is very hard to achieve a real-time behaviour analysis system according to the current state-of-the art techniques, this research was focused in the improvement of every step in a classic behaviour analysis system. To this end, this section contains a brief introduction of the contributions included in this paper. To a better comprehension, this section is divided into three different blocks, related with the classic behaviour analysis system boxes, that is, detection, tracking, and human behaviour analysis.

1.4.1 Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios

As previously exposed, it is very difficult to achieve a balance between efficiency and accuracy. During the detection procedure, low-level techniques like optic flow or background subtraction achieve real-time performance when using regular computers (one core, less than 4GB of RAM memory, no GPU acceleration, ...) but fails when dealing with crowded scenes, being unable to distinguish between people that are close enough. On the other hand, high-level approaches like the Viola-Jones or the HOG can solve the crowd issue. However, the computational cost make them unable to be used in real time scenarios. In a similar way, low-level tracking techniques like the Kalman filter cannot cope with complex human movements, whereas associated-based tracking methods requires too much processing time.

In our first contribution to this research, a real-time multiple-target tracking was developed (Cancela, Ortega, Penedo, & Fernández, 2011; Cancela, Ortega, Fernández, & Penedo, 2012). The main idea behind this project was to obtain a human tracking model that could be used in a regular computer, without any kind of acceleration hardware. Thus, this real-time tracking methodology is focused in the simplicity of its components, whereas the complexity of the model lies in how the components are merged.

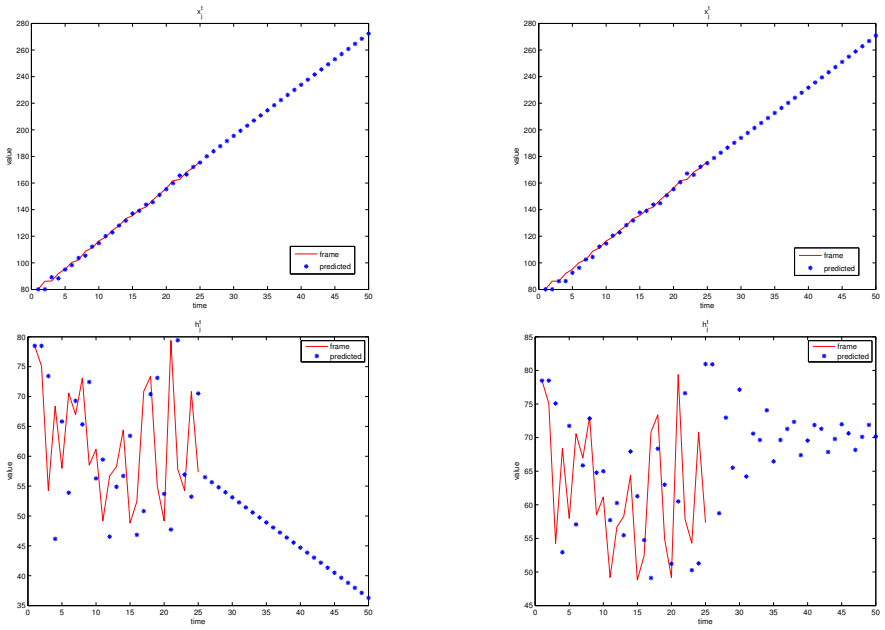
Object Detection: A background subtraction technique is used to detect every target in the scene. Three techniques were tested. A few considerations have to be taken into account when dealing with each technique. In first place, the method used by Horprasert (Horprasert et al., 2000) has a manual parameter, causing the system to behave incorrectly if it is not chosen correctly. On the contrary, both Mixture of Gaussians (MoG) (Stauffer & Grimson, 1999) and the Codebook model (Kim et al., 2005) achieve similar results. The Codebook model (using the YCbCr-color space) is selected because of the computational cost. A shadow suppression technique over the HSV-color space is also used (Cucchiara et al., 2001). Finally, every isolated blob j at time t is marked as a new detection and it is encapsulated as an ellipse representation

$$z_j^t = (x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t), \tag{1.10}$$

that includes the center (x_j^t, y_j^t), axis lengths (h_j^t, w_j^t) and its orientation (θ_j^t).

Low-level Tracker: When there is no large variations between the speed along successive frames (little noise), the Kalman filter achieve good results when tracking isolated targets in the scene. However, when dealing with a large amount of noise, as it occurs with the axes size h_j and w_j more problems arise. This usually happens because both the legs and the head are too thick, causing the background subtraction to remove them if the noise is high. The consequence of this problem is a high variation of the ellipse size between frames. When a target is lost, instead of maintain the values between a certain range, it starts to decrease quickly (Fig. 1.3-(a)), which is not desirable when trying to recover the tracking after a few iterations. This issue becomes critical when a target is lost and its position has to be predicted. Thus, a different low-level tracker is proposed, based in a classic linear filter (Adaline). The prediction techniques behaves similar to the Kalman filter as the noise remains low. However, when dealing with a large amount of noise, it is able to stabilize the value near to the average, causing the model to perform better than the Kalman filter under occlusions (Fig. 1.3-(b)).

The linear filter is used to predict the position of a target in a new frame. To



(a) (b)

Figure 1.3: (a) Kalman filter predictions (x_j^t and h_j^t component). (b) Adaline filter predictions (x_j^t and h_j^t component).

establish the similarity between a new detection and any given target, the methodology checks if the detection ellipse center fits within the target prediction ellipse.

Appearance tracker: Using the background subtraction information, an appearance tracker is performed in order to distinguish between different targets when the low-level tracker cannot. A pool of histograms are used as target representation. The proposed methodology uses L*a*b, which is a color-opponent space with dimension L for lightness and a and b for the color-opponent dimensions. Illumination is isolated into one unique component, being easier to distinguish every target within the scene by using the other two. Thus, the selected pool of histograms are the following:

$$h = \omega_1 L + \omega_2 a + \omega_3 b, \tag{1.11}$$
$$(\omega_1, \omega_2, \omega_3) \in \{(1,0,0), (0,1,0), (0,0,1), (0,1,1), (0,1,-1)\}.$$

The illumination component is kept as a backup in cases there is no enough evidence to make a decision with the other histograms. All histograms are normalized and discretized into 64 bins to perform the computation, forming a 320 feature vector. The *Bhattacharyya distance* (Bhattacharyya, 1946) is used to compute the similarity between two different pool of histograms.

Collision Detection: The background subtraction technique cannot distinguish between different targets that are close together. Thus, a different module is needed in order to detect whether one blob contains more than one target. To solve that, two different cases are considered

- *Grouping event:* A *grouping event* occurs whenever two or more target predicted ellipse centroids fit within an ellipse detection in the new frame. Note that it is possible that one centroid of a tracking object could be within more than one different predicted ellipses. In that case, only the ellipse which centroid is closer to its centroid is considered.

- *Splitting event:* A *splitting event* occurs whenever two or more new ellipse detections in the new frame fit within a group predicted ellipse centroid. The same case mentioned on the grouping detection about centroids that fit within two or more ellipses has to be considered.

After a splitting event occurs, the model has to re-identify every target involved within the erased group. A re-identification technique must also be used after a occlusion event. Thus, the high-level tracker is used to cope with this situation. A recover id is confirmed whenever every color histograms pass the test using the Bhattacharyya Distance.

Experimental Results: This approach was tested in two different datasets: CANDELA Intersection Scenarios (*CANDELA, Content Analysis and Networked DELivery Architectures*, Date accessed: february, 2015) and CAVIAR (*CAVIAR, Context Aware Vision using Image-based Active Recognition*, Date accessed: february, 2015). The CANDELA dataset videos take place outdoors, in an intersection. Although the light conditions in these videos are good, these scenarios have a lot of complexity. There are multiple object interactions in a short space of time, which implies the system has to be robust against collision events. In Fig. 1.4 it is shown how our algorithm can detect target collisions, and how it can successfully recover the correct identifications after that.

On the other hand, the CAVIAR dataset contains 26 different videos using two different cameras. Although state-of-the-art papers use the corridor camera to test their algorithms, there is no frame without moving objects in the scene, so the background subtraction algorithm cannot be successfully trained. Instead, the front camera is used. Objects that are tracked within a group are considered as a correct match. Table 1.1 shows the obtained results. Our method can successfully track multiple target in non-crowded environments. Furthermore, our method can process

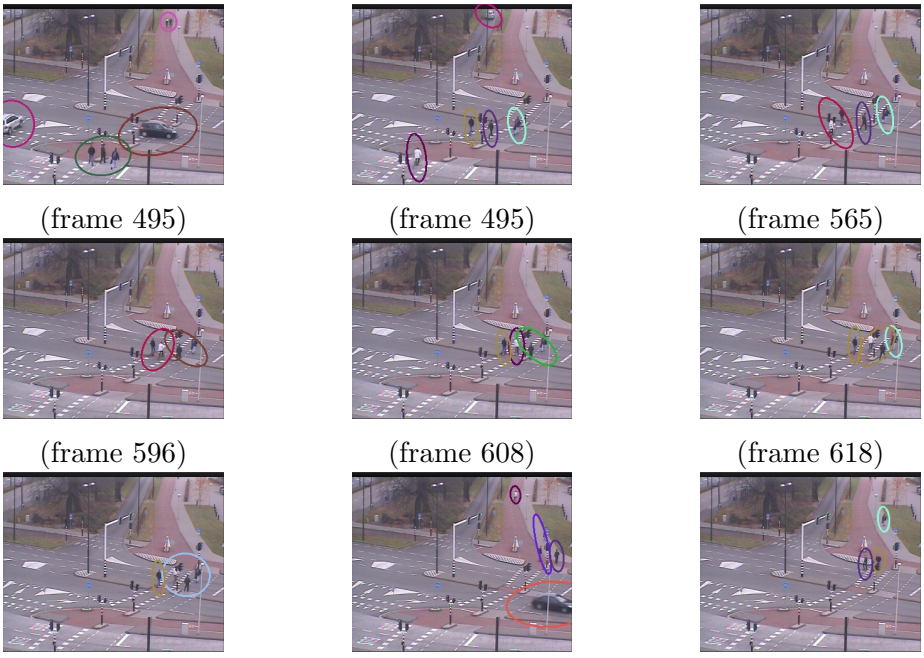


Figure 1.4: Quick succession of people collision events. The system is able to recover the previous id of all the four people involved in this collision.

more than 50 fps using a regular computer (Pentium Quad Core running at 2.40GHz with 4 RAM GB), making it suitable to be used in real-time scenarios.

1.4.2 Multiple Human Tracking System for Unpredictable Trajectories

As mentioned in section 1.1.2, it is very difficult to segment a pair of targets when they are overlapped by using background subtraction techniques. Although the frame rate of the previous approach is high (it can process more than 50 frames per second), it is not a suitable methodology to deal with crowded scenarios. Thus, a different multiple-target tracking approach, based in high-level detection techniques, was developed (Cancela, Ortega, & Penedo, 2014).

The main idea of this research is the development of a pedestrian multiple-target tracking system that can work under crowded scenarios, where it is not possible to achieve a good background reference. At the same time, it has to be able to distinguish between pedestrians that are close together. Finally, similar to the previous approach, the computational cost is a key factor. Thus, our research are focused in reducing, as long as possible, the complexity of the methodology.

A combination between two different detection techniques (Viola-Jones and HoG

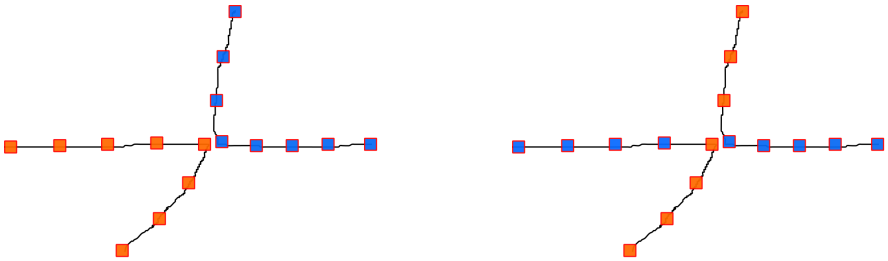
Table 1.1: CAVIAR dataset results. Good results are achieved, having in mind our system can be used in real-time scenarios.

	GT	MT	PT	ML	IDS	FRAG
Wu et al. (Wu & Nevatia, 2006)	144	72, 22%	23, 61%	4, 15%	13	42
Wu et al. (Wu & Nevatia, 2007)	189	74, 07%	21, 69%	4, 24%	19	40
Zhang et al. (Zhang, Li, & Nevatia, 2008)	140	85, 71%	10, 71%	3, 58%	15	20
Huang et al. (Huang, Wu, & Nevatia, 2008)	143	78, 30%	14, 70%	7, 00%	12	54
Xing et al. (Xing et al., 2009)	140	84, 28%	12, 14%	3, 58%	14	24
Li et al. (Y. Li, Huang, & Nevatia, 2009)	143	84, 60%	14, 00%	1, 40%	11	17
Song et al. (Song et al., 2010)	75	84, 00%	12, 00%	4, 00%	8	6
Ours [1]	110	90, 00%	7, 26%	2, 74%	8	16

type-classifiers) are used to increase both the speed and the accuracy of the target head detection. A particle filter is used to track every target in the scene, whereas an appearance tracker is used to recover the identification of occluded targets. This approach is particularly appropriate for uncontrolled scenarios like sport events, where it is difficult to predict the targets' behaviour. Assume, for instance, a basketball video. When an offensive player makes a crossover, or a fake movement, and the defender falls into the trap, an associated based tracking would swap the identifications, as Fig. 1.5 shows. Associated based tracking techniques assume target motions are stable, i.e., linear and constant speed in a short period, being incapable of dealing with big and unpredictable movements like that in sports. It is very difficult anticipate that kind of movements.

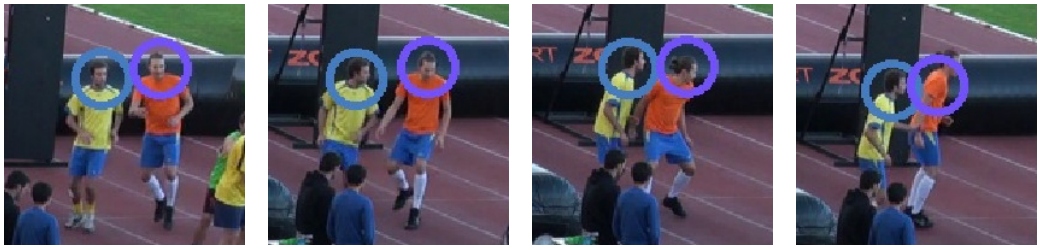
Object Detection: To detect every person in the scene, this architecture follows a similar approach than exposed in (M. Li et al., 2009): a Viola-Jones detector is combined with a HOG featured based SVM in order to obtain a good balance between accuracy and speed. The reason about combining two different classifiers is related with the efficiency: as the HoG computational cost is high, a less expensive technique is used to limit the positions where the HoG is computed. In this case, a Haar-like method, the Viola-Jones type classifier, is used to detect head-shoulders omega shape feature. This method reduces the computational cost. However, the classification performance is poor. Thus, a HoG-feature SVM is used whenever the Viola-Jones type-classifier obtains a positive match, whether to reinforce the hypothesis or to refute it.

The target detection is performed as follows: First, a background subtraction technique is used in order to restrict the regions in the scene where to perform the next steps. Later, a Viola-Jones type classifier is applied in that regions, that is,



(a) Real trajectories (b) Associated based tracking output

Figure 1.5: Associated based tracking issue. When dealing with sudden orientation changes, associated based tracking swap the identifications.



(a) (b) (c) (d)

Figure 1.6: Particle filter error. Along successive frames, purple target is losing the quality of the detection.

where movement is detected. Finally, the HOG feature based histogram is responsible for evaluating the Viola-Jones positive detections. To reduce the computational cost of the Viola-Jones type classifier, only one patch size per pixel is taken into account. To establish that, the object height h at the position i follows the equation

$$h_i = \frac{y_i}{y_c}(v_i - v_0), \tag{1.12}$$

being y_i the 3D object size, y_c the camera height, v_i the position in the image that is considered and v_0 the horizon point.

Tracking System: Following the same reasoning explained in the previous approach, a bunch of adalines is used to predict the velocity of each target. Contrary to the previously explained approach, only the velocity of the parameters related with their position are detected, since the size of the patch is determined by using Eq. 1.12, reducing the computational cost.

Prior to use any appearance tracker, the target detection technique explained before is also used as part of the detection technique. The use of the detection system can correct the error caused by the appearance tracker. For instance, in Fig. 1.6-(d), the purple target head is lost after using the appearance tracker during a

few iterations, resulting in a bad performance. However, if the detection system can successfully recover the head position, the error caused by the appearance tracker can be reduced. The hungarian method (Kuhn, 1955) is used to perform the detection assignment. Targets that are not associated with any patch detected by the Viola-Jones type classifier are tracked by using the appearance tracker.

The combination between the Viola-Jones type classifier with the HOG-based SVM achieves a high precision. However, the recall is low, causing the detections module to often lose every target within the scene. A particle filter is proposed to deal with this issue. The extracted local HOG features are used as object representation, whereas the Bhattacharyya coefficient compares a pair of appearance vectors, which has been proved as a good method for tracking non-rigid objects (Comaniciu, Ramesh, & Meer, 2000).

A particle filter is deployed for every tracked target. Let z_j be a target previously tracked, a number of particles are thrown following the equation

$$\breve{z}_j^{t+1} = \tilde{z}_j^{t+1} + \omega, \tag{1.13}$$

where $\tilde{z}_j^{t+1}$ the predicted position of the target z_j at time $t+1$ and $\omega \sim N(0, \Sigma)$ is Gaussian noise. By adding the latter to the equation, a set of different particles are obtained. The new target position is selected as the patch that maximizes the Bhattacharyya coefficient between that particle and the target appearance.

Re-Identification Procedure: It is very difficult to re-identify a person by using the head appearance. The head pose is critical: the appearance when a person is looking to the camera is totally different compared with the pedestrian looking in the opposite direction (skin vs hair). Thus, this approach proposes to define a rectangle in the bottom of the head-shoulder detection as a simple body estimation. This body estimation is split into different horizontal stripes, each one with its own pool of histograms. Each histogram is normalized and discretized into N bins. A modified Bhattacharyya coefficient is used to compare two different appearances, since the metric has to take into account that some body regions may be occluded and have to be avoided.

This approach takes advantage of one of the features explained in the previous approach. Every time a target is lost, an ellipse representation of the blob that contained the person is instantiated. This blob is followed over the entire scene. Thus, the re-identification technique is only executed within that ellipse representations. This restriction reduces the computational cost of the tracking system by delimiting the number of re-identification procedure, as long as the possible wrong matches.

Experimental Results: A full sport event was recorded in order to test this

Table 1.2: Tracking performance on the basket sequence for our tracking system. Different configurations are shown, depending on the modules enabled.

Method	MOTP	MOTA	Prec	Rec
HOG head detections	75.6%		27.5%	55.3%
Our tracking with no Collision/Recovery System	77.5%	43.4%	48.8%	79.7%
Our tracking with no Background Information	84.5%	48.1%	50.6%	92.0%
Our tracking with no HOG assist	80.7%	64.0%	85.0%	73.9%
Our tracking	84.5%	73.5%	88.2%	82.0%

methodology, consisting in a 3×3 basket match. The video used in this test have a 640×368 resolution running at 25 frames per second. Although it is not a heavily crowded scene, it is a very challenging environment, since multiple situations occur that usually do not happen under usual scenarios, including collisions, crossings, sudden orientation changes, jumps or squatting people.

Both Viola-Jones type classifier and the SVM are trained by using the generic dataset introduced in (M. Li et al., 2008). Head-shoulder detections in the scene are assumed to have a 32×32 size, which is the same size used in the training dataset. To perform the HOG feature extraction, each sample is divided into 64 cells. Defining 4 adjacent cells as a block, and using a one cell stride, 49 different blocks are obtained. For each cell, a histogram of 8 different orientation bins is computed and stored. Each block is also normalized.

In order to evaluate our system in a quantitative way, CLEAR standard evaluations tools are used (Stiefelhagen et al., 2007): MOTA (Multiple Object Tracking Accuracy), which take into account false positives, missed targets and identity switches, and MOTP (Multiple Object Tracking Accuracy), which measures the average distance between true and estimated targets. A predicted bounding box is considered correct if it overlaps more than 25% with a ground-truth bounding box.

A ground truth data is created, containing more than 10000 annotated heads. Table 1.2 shows the obtained results. The combination of all the different techniques used in this approach can increase the accuracy of the tracking system. Although the number of switch identifications is similar in all different configurations, the recovery system is able to revert the wrong matches, causing both MOTA and accuracy of the system to be improved.

Contrary to our previous approach, this technique can only run 6 fps in a regular machine (Pentium Quad Core running at 2.40GHz with 4 RAM GB). Both back-

ground subtraction, the Viola-Jones type-classifier and the HOG-based SVM takes near the 95% of the computational cost. However, these three methods are east to parallelize by using GPUs, making this algorithm a solid starting point to achieve a high-level real-time multiple-target tracking system.

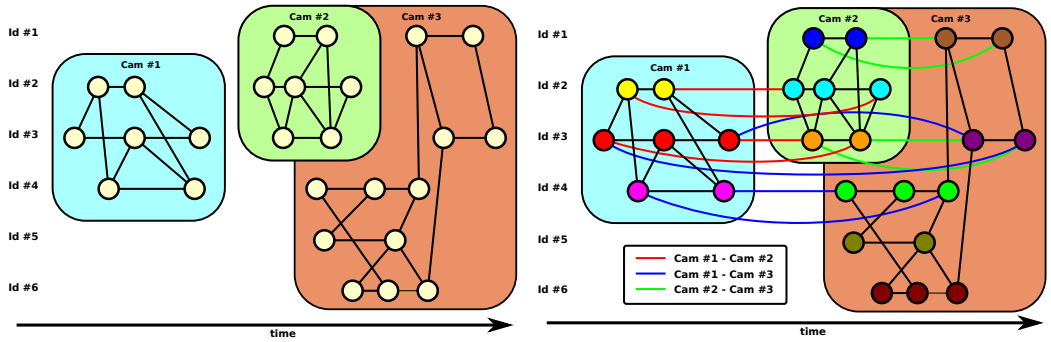
1.4.3 Open-world person re-identification by multi-label assignment Inference

One of the major drawbacks related with the previous approaches presented in this research relies in how the multiple-target tracking systems can recover a pedestrian identification after the target is lost during a large amount of time (seconds). This problem is crucial if you plan to use multiple non-overlapping cameras, where a target is lost right after leaving one camera and it cannot be recovered until reappearing in another one. To solve this problem, our next contribution (Cancela, Hospedales, & Gong, 2014) is focused in the re-identification field, introducing a new field of action into the issue.

Classic Re-identification problem can be seen as a retrieval problem: given a predefined 'gallery' set of known individuals, systems try to label each new 'probe' detection with the identity of the matching gallery individual. This is a good starting point, but relies on two very strong assumptions: the total number of people in the scene is known a priori, and it is always assumed that every person that is lost reappears at some point. These constraints make the classic re-identification problem to be unsuitable for real-world scenarios. We refer to this unconstrained setting as the 'open world' ReID problem. The open-world problem is more challenging for two reasons: (i) the total number of unique people in the scene is unknown, and (ii) each subject may reappear or not in some unknown subset of the cameras.

In this approach we consider for the first time the most general open-world re-identification problem, where there is no prior information about the number of people. Our framework can answer qualitatively more general queries than existing re-identification systems such as: How many people are in the scene?, If a person leaves a camera, which other cameras did he appear in, or did he simply disappear?. A new Conditional Random Field (CRF) model is introduced to overcome all the issues presented in this open-world re-identification task.

Target appearance: To describe each pedestrian, an ensemble of localized features (ELF) is selected: each target detection is split into six non-overlapped horizontal stripes. For each stripe, 8 color histograms are computed and normalizes (RGB, YCbCr and HSV (V is removed)) and 21 texture filters (Gabor, Schmid) de-



(a) Step 1 (b) Step 2

Figure 1.7: CRF illustration. In the first step, only detections within the same camera are connected. In the second step, a restricted connection between cameras is allowed.

rived from the luminance channel, resulting into a 2784-dimensional low-level color and texture feature vector.

Re-Identification Procedure: A novel two-step Conditional Random Field is presented. Fig. 1.7 shows an illustration about how the CRF model works. During the first step, only connections between detections recorded with the same camera are allowed. The output is used as input within the second-step, where a limited number of connections are allowed.

The CRF is a network with N nodes and N different states, being N the number of detections. The number of states and nodes are the same because it has to be taken into account the extreme case where there is only one detection per person. Each detections is defined as combination of five different properties: target appearance, camera that makes the detection, position within that camera, velocity of displacement and time where the detection was made. A combination between all this information is used to estimate whether two different detections belongs to the same person or not. Two different distance metrics were tested: RankSVM (Prosser et al., 2010) and KISSME (Kostinger, Hirzer, Wohlhart, Roth, & Bischof, 2012). Both metrics were normalized into 0 and 1, in order to have a probability that the two detections belongs to the same person.

Experimental Results: Our methodology is evaluated on the SAIVT-SoftBio dataset (Bialkowski et al., 2012). 3 different cameras were considered, including disjoint labeling (a person that appears in one camera may or may not appear in the others). Fig. 1.3 shows the F1-Score obtained by our methodology, compared against two different naive methods. Although the baseline methods obtain somewhat better recall, it is based on their non-conservative nature: the number of

Table 1.3: Re-identification F_1-Score among three cameras from SAIVT. The last column shows the global performance. Other columns show local performance. E.g., C3-C8 shows the quality of the connections between camera 3 and camera 8 when the whole CRF model is computed.

	C3	C5	C8	C3 - C5	C3 - C8	C5 - C8	Whole model
Naive RankSVM	31.7%	34.1%	27.1%	15.9%	20.1%	24.6%	26.2%
Naive KISS	32.6%	29.4%	34.7%	23.4%	31.0%	29.6%	29.5%
RankSVM+CRF	50.1%	41.1%	**73.2**%	18.2%	43.4%	32.4%	42.0%
KISS+CRF	**57.3**%	**52.0**%	70.0%	**30.3**%	**47.6**%	**43.7**%	**48.3**%

Table 1.4: Inferring the number of distinct people in the dataset.

Ground truth	Naive RankSVM	Naive KISS	RankSVM+CRF	KISS+CRF
48	61 ± 17.6	57.8 ± 11.2	65 ± 13.2	$\mathbf{54.1 \pm 7.9}$

different labels in the output is reduced, resulting in a huge increment in the number of false positives, resulting in significantly worse precision. Our CRF model is more robust, as evidenced by its maintenance of high precision values. Moreover, it improves both of the base methods it is paired with.

Our model it is also capable to successfully infer the number of distinct people within the dataset. Fig. 1.4 shows how our CRF can overcome the baseline.

1.4.4 On the Use of a Minimal Path Approach for Target Trajectory Analysis

As early mentioned, the tracking system is the base to perform a fully automatic behavior analysis system. Then, a knowledge layer is added to the tracking system to analyze the trajectory of each target. State-of-the-art path analysis techniques were discussed in section 1.3.1. To summarize the drawbacks in these systems:

- *Information a priori:* A previous training is needed in order to establish what is a normal behavior.
- *Online updating:* It is very difficult to adapt the models 'on the fly'. At some point, one usual path may be changed due to some problem (for instance, an accident), causing a new usual path to appear. However, how can the model realize that the initial path is no longer valid? The two trajectories tend to coexist at the same time.
- *Memory requirement:* Related to the previous issue, the addition of new trajectories causes the memory requirements to grow exponentially.

- *Occlusion impact:* It is very difficult to achieve a perfect tracking system. Consequently, small trajectory frames tend to appear in the model as complete trajectories. Tracking failures highly impact in the quality of the system response.

- *Path length:* State-of-the-art techniques require a target trajectory to be finished in order to establish whether its behavior is normal or not.

Thus, our next contribution to this research is a novel behavior analysis approach (Cancela, Ortega, Penedo, Novo, & Barreira, 2013) that can solve all the mentioned problems. Contrary to classic clustering-based techniques, our contribution follows a different approach. The key point is simple: to try to understand what is a normal behavior. However, the solution is complicated to solve, because there exists some degree of subjectivity in the answer. Thus, the definition of a normal behavior has to be simple, using a minimum number of rules.

Minimal Path: Having this idea in mind, two hypotheses were made to establish the common normal behavior:

Hypothesis 1 *Each person tries to reach a geographic goal.*

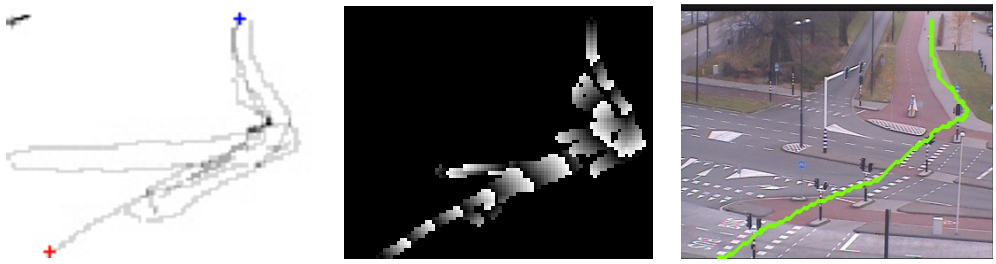
It is assumed that targets have the intention to reach some goal within or outside the scene. As a consequence, people that are stopped or start to move erratic are considered as abnormal movements.

Hypothesis 2 *The trajectory used to reach the goal is ruled by the common pedestrian behavior.*

This is the crucial point of the algorithm. In other words, this means that, when a pedestrian tries to go to some location, it follows the same path that other people used to reach the same goal. Note that this definition can also be successfully applied to describe classic clustering-based techniques.

Thus, the goal of this methodology is to find the most used path that connects the starting and the ending point of any given pedestrian. Mathematically, it is equivalent either to solve this problem or its dual, that is, to find the path that minimizes the inverse of the frequency of the targets. And this is a very interesting feature, because this dual approach is the classic definition of the minimal path search.

Instead of using classic graph-search algorithms which suffers from 'metrication error' (Cohen & Kimmel, 1997), our solution is based in the use of geodesic active



(a) (b) (c)

Figure 1.8: Fast Marching Method example. (a) Potential image. In red, initial point. In blue, ending point. (b) FMM minimal action surface. (c) Minimal path, as a result of applying the Heun's method in the back propagation over the minimal action surface.

contours. Specifically, the Fast Marching Method (FMM) (Sethian, 1996) was used, achieving a higher accuracy than the graph-search algorithms like A* or F*, while maintaining the complexity of the algorithm ($\mathcal{O}(N \log N)$).

Discussion on the Potential: The FMM requires the definition of a potential field in order to be used to establish the minimal path between two different points. As defined in our hypotheses, our potential contribution is the inverse of the pedestrian frequency at each point of the image. However, different potential approaches can be used. For instance, it is possible to introduce prior knowledge about the scene by restricting some forbidden areas.

Our model is also capable to use multiple potential functions, depending on either where is the initial point located or which kind of object is tracking (pedestrians, cars, ...). The advantages of this technique is that the potential function is easily updated: it only needs to take the output of the tracking method in order to update the frequency of each scene point. Using the same reasoning, it is possible to initialize the model with no information about the environment. That makes our model to be used without any previous training step. The memory requirement is stable: this technique only requires the definition of the potential function. The occlusion impact is reduced, because even if the tracking system returns multiple fragmented trajectories, the combination of all of them can be used to predict any possible combination of these short paths. Finally, this technique do not require a complete trajectory to analyze its behavior: it only requires an initial and an ending point.

Abnormal Behavior Detection: The FMM output, starting in any given initial point, is a monotone surface. It only contains one local, hence, global minimum, which is located at that point. To obtain the usual path between the starting point and any other given point in the scene, only a back-propagation from the ending

(a) (b) (c)

Figure 1.9: Matching trajectory methods. (a) Target route. (b) Minimal path route. (f) Target route fitted into the distance map image. This route is really close to the minimal path solution.

point is needed. That back-propagation follows the FMM surface maximum gradient descendant. Fig. 1.8 shows an example about how to obtain an 'usual path'.

Once the 'usual path' is computed, it has to be checked against the real path performed by the target. In first place, a register technique was proposed to establish a similarity between the real and the 'usual' path (Cancela, Ortega, Fernández, & Penedo, 2011). However, this technique requires a complete path to evaluate the target behavior. To avoid this limitation, a distance map is computed, containing the minimum distance from any given point in the scene to the 'usual path'. Thus, two new metrics are introduced, the Distance Map (DM) and the Weighted Distance Map (WDM), that computes the average distance of the real trajectory against the 'usual' one. The difference between both metrics rely in the FMM procedure: in the second one, the FMM surface is not computed over the positions in the scene that were not reached for any target.

Experimental Results: There is a huge drawback when testing any behavior analysis methodology: there is no annotated dataset. Thus, it is very difficult to compare any methodology against the state-of-the-art. In order to solve this issue, a new dataset, Behavioral Analysis and Recognition Dataset (BARD) (*BARD, Behavioral Analysis and Recognition Dataset*, Date accessed: february, 2015) was created to test this methodology. This dataset contains human movements over a crossroad. Usual movements cross the scene along the pavement, while abnormal movements cross the grass. Different videos were used with a duration between one and two minutes, resulting in more than 5000 samples. Fig. 1.10 shows the ROC curve of our new metrics, compared against state-of-the-art vector distance methods. Our methods clearly outperforms the baseline.

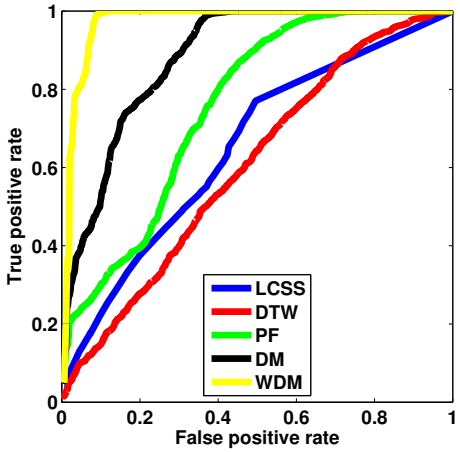


Figure 1.10: ROC curve for the pedestrian trajectory analysis. Both DM and WDM outperforms classic vector distance methods.

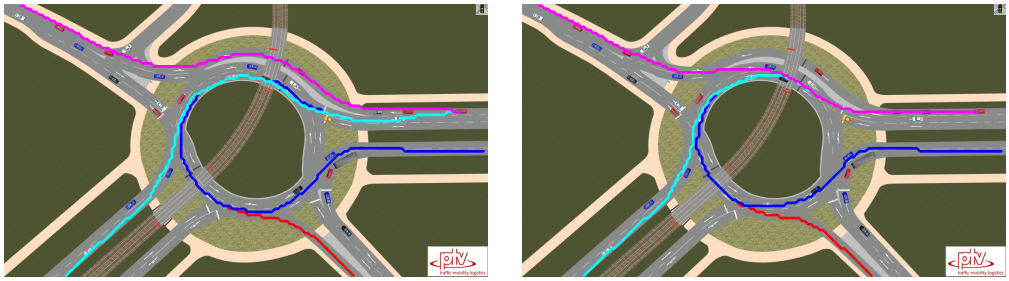
1.4.5 Path Analysis Using Directional Forces. A Practical Case: Traffic Scenes

The use of a minimal path approach to detect abnormal behavior can successfully detect whether a pedestrian trajectory is abnormal or not. However, there still exist some limitations about where to use the model. One of the most critical issue is related with the FMM nature: it does not take into account the orientation of the motion. In other words, when potential is a scalar function, in does not depend on the direction. That means the cost of reaching any point in the scene is not influenced by the direction of the FMM wavefront expansion.

Although it is not a huge problem when dealing with people tracking, it is a huge drawback in other scenarios, like the traffic analysis. The computation of the minimal path surface has to take into account the orientation of the motion, the lanes, etc. Thus, a more powerful technique is needed that can cope with the FMM limitations.

In this approach, the use of the Ordered Upwind Method (OUM) (Sethian & Vladimirsky, 2003) is proposed. This algorithm can deal with more complex Static Hamilton-Jacobi equations, where information about the orientation of the motion is included. In essence, the structure of the algorithm is somehow similar to the FMM. It differs in two parts: (i) the number of nodes that have to be updated; and (ii) the updating equation.

Discussion on the Potential: In this approach a two channel potential is developed, containing the average velocity vector at each point of the scene. The



Frame 500 Frame 1200

Figure 1.11: Minimal paths obtained in different times. The paths start from the upper-right roundabout entrance, reaching all the departures. All the routes are updated at every moment without needing to store all the different paths.

scene properties are also taken into account. Since this algorithm is focused in tracking vehicles, the propagation technique is only allowed over the asphalt. Once the surface of minimal action is computed, the 'usual' path is obtained by following the same approach used with the FMM algorithm.

Experimental Results: To test this methodology,he traffic simulator PTV Vissim provided by the PTV group was used. A turnaround scenario was chosen due to its complexity: there is only one direction where to move within a turnaround, making this a perfect scenario where to test this new methodology. Fig. 1.11 shows the main routes obtained, in different times, by using our method. Two conclusions can be derived from these results: (i) the OUM is able to take into account the orientation of the movement when computing the minimal path approach; and (ii), the method can detect changes in the usual behavior, modifying the path of the usual routes.

1.4.6 Trajectory Similarity Measures Using Minimal Paths

The use of the OUM instead of the FMM increases the quality of the solution. However, the computational cost is also increases ($\mathcal{O}(N \log N)$ in the case of FMM and $\mathcal{O}(\Upsilon N \log N)$ the OUM). When dealing with pedestrian behavior, the results do not differ between these two techniques. Thus, FMM is going to be used in the successive improvement of this pedestrian abnormal behavior approach.

Two huge drawbacks remain in our minimal path behavior algorithm:

- *Distance Map computational cost:* The computation cost of the Distance Map mentioned in section 1.4.4 is similar to the FMM ($\mathcal{O}(N \log N)$), which is really high. The FMM surface is computed only once per target, whereas the Distance Map has to be computed every time a behavior analysis is launched.

- *Isolated obstacles:* If, for instance, there is some isolated obstacle in the middle of a possible 'usual' track, the target has to avoid it. However, which side is going to take to do that? Left or right? Both decisions results in an 'usual' path. However, the initial model only returns one option.

In our next research (Cancela, Ortega, Fernández, & Penedo, 2013), a new behavioral metric function is defined that can cope, at the same time, with these two issues. A new technique is developed by using the minimal path algorithm properties, obtaining a metric without increasing the computational time, solving the distance map computational disadvantage.

Discussion on the Potential: Two different potential images are used in parallel. The first one is the same potential image used in previous approaches. This potential is used to determine the order in the front propagation procedure. Every time the FMM surface is update, another surface is also computed using a constant potential. If a constant potential is used in a FMM approach, a distance map is obtained. But this distance map is different from the one previously used: instead of having the distance between any point with respect to the 'usual' path, it contains the distance between any given point to the initial point used to compute the FMM.

How can this property be relevant? A new hypothesis is established in order to take advantage of this new approach.

Hypothesis 3 *Having an 'usual' behavior, the difference between the target real length and the minimal path trajectory length tends to zero.*

Having this in mind, different metrics are presented for detecting abnormal behavior. All these metrics are based in two different equations. The first one tries to obtain the relation between the target route and its associated minimal path. It is called the Minpath Relation (MR), and is defined by

$$MR(p_N) = \left(\frac{\sum_{i=2}^{N} d(p_{i-1}, p_i)}{D(p_N)} - 1 \right)^2, \tag{1.14}$$

where $\mathbf{p} = \{p_1, \ldots, p_N\}$ is the real target trajectory and $D(p_N)$ is the distance map value at the end of the trajectory. The second one tries to detect local variations in the MR metric. It is called Local Minpath Relation (LMR), and is defined by

$$LMR(p_N) = \left(\frac{d(p_{N-1}, p_N)}{D(p_N) - D(p_{N-1})} - 1 \right)^2. \tag{1.15}$$

In both metrics, values close to 0 mean the path is correct, while higher values could indicate an abnormal behavior.

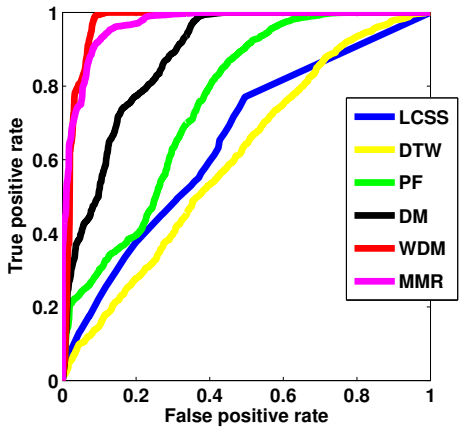


Figure 1.12: ROC curve. Our new metric outperforms the baseline methods, with the exception of the Weighted Distance Map. However, the computational cost allows our new method to be more suitable to be used in real-time environments.

Experimental Results: Our Behavioral Analysis and Recognition Dataset (BARD) (*BARD, Behavioral Analysis and Recognition Dataset*, Date accessed: february, 2015) was used again to test these new metrics. Fig. 1.12 shows the new results compared against the results obtained using both state-of-the-art and our previous metrics. Different comparisons show that the Mean Minpath Relation achieves the better results of our new metrics. Compared with previous results, it is found that the Weighted Distance Map developed in our first minimal path approach achieves the higher results. However, the results are pretty similar. What is more important, the new metric reduces the computational cost from $\mathcal{O}(N \log N)$ to $\mathcal{O}(1)$, meaning this new method is more suitable for being used in real-time applications.

1.4.7 Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths

All the methodologies presented in this research were focused in detecting trajectory 'abnormal' behavior. However, only the shape of the path are taken into account. There is no information about how the trajectory was performed, that is, the velocity of the target is not taken into account.

Knowing the distance map from the beginning presented in the previous subsection results in a good metric to model pedestrian trajectory behavior, it is possible to use the same approach but introducing extra information about the environment. In this contribution (Cancela, Iglesias, Ortega, & Penedo, 2014), a temporal surface is introducing, enabling our minimal path technique to include the velocity into the pedestrian behavior analysis. Using the same parallelization exposed in the distance

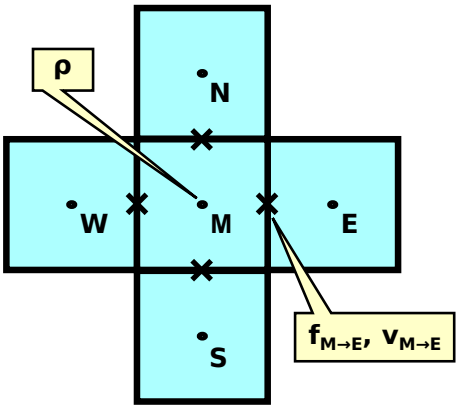


Figure 1.13: Discretized grid structure. ρ is the number of times a target reaches the point, whilst f and v are both the number of times a target crosses the path in each direction, and their most common speed.

map from the beginning technique, the inclusion of this new surface does not affect the complexity of the algorithm.

Discussion on the Potential: Contrary to our last contribution, a new potential is provided to introduce orientation information in the front propagation procedure. In early attempts, each point in the scene only stores the number of people that reaches it. In this new approach, extra information is added:

- *Orientation information:* Four different directions are taken into account, as depicted in Fig. 1.13 (north, south, east, west). The potential includes, for any given point, the number of people who, after reaching it, follows any of the four different directions.

- *Velocity information:* Following the same idea developed in subsection 1.4.5, for each one of the four predefined directions, the most probable velocity is stored.

To compute the time surface, the same technique explained in section 1.4.6: Every time the FMM surface is update, the time surface is also computed by using the velocity information as potential. Only the velocity stored in the direction of the motion is used; the others are discarded.

Again, different metrics were performed to establish if a target behavior is 'usual' or not. The metrics follows the hypothesis

Hypothesis 4 *If a path* $\mathbf{p} = \{p_1, \ldots, p_N\}$ *have an usual behaviour, then* $\forall p\mathbf{p} \in , T_p \approx t_p$,

where t_p is the real time trajectory at point p and T_p is the expected one. In other words, it says that the relation between the real and the expected time is $\frac{T_p}{P_t(p)} \approx 1$. Having this idea in mind, a new metric called it Time Log-Likelihood (TL) was created, defined by

$$TL(p_N) = \| \log(T_{p_N}) - \log(t_{p_N} - t_{p_0}) \|. \tag{1.16}$$

Values close to 0 mean the path is correct, while higher values could indicate an abnormal behaviour. As in early attempts, this final point mentioned in this metric is not necessarily the moment when the target leaves the scene. It is only a moment when the trajectory is evaluated. This strong advantage allows this method to be used in real-time systems, since the computational complexity of the metric, once the surface T is computed, is $\mathcal{O}(1)$.

Experimental Results: Intuitively, the idea about using a time surface field to model human trajectory behavior seems plausible. However, what happens in crowded scene, when some people may affect the behavior of the others. In order to have an answer to that question, this methodology was tested in two different scenes: a parking lot and a train station.

When trying to test any trajectory analysis, the same problem arises: there is a total absence of ground truth information. It is very hard to define whether a trajectory is normal or not. In related papers, they use some visual information to probe its effectiveness (Wang, Ma, Ng, & Grimson, 2008), (Wang et al., 2009), (Wang et al., 2011), (B. Zhou, Wang, & Tang, 2012). However, a different statistical approach was used in this contribution. Since all the earlier attempts to model the human trajectory behaviour have used some learning methods to determine the usual behaviour, we extract the idea that, having no information about the environment, every method consider the most usual paths as normal movements, being the outliers the abnormal ones.

Ideally, we expect an abnormal behaviour measure to have an asymptotic curve, like $\frac{1}{x}$, where the most part of the trajectories are normal, with a few abnormal movements. That is, the more erratic a trajectory is, the lower frequency it has. Fig. 1.14 shows how our method, even under crowded scenes, obtains the expected solution. We can conclude that the effect caused in a target by the rest of the people is not as huge as one may think. And with this hypothesis, we can pre-compute the time surface at the beginning without significantly increasing the metric error. Having this in mind, computing the behaviour at each position can be done in constant time.

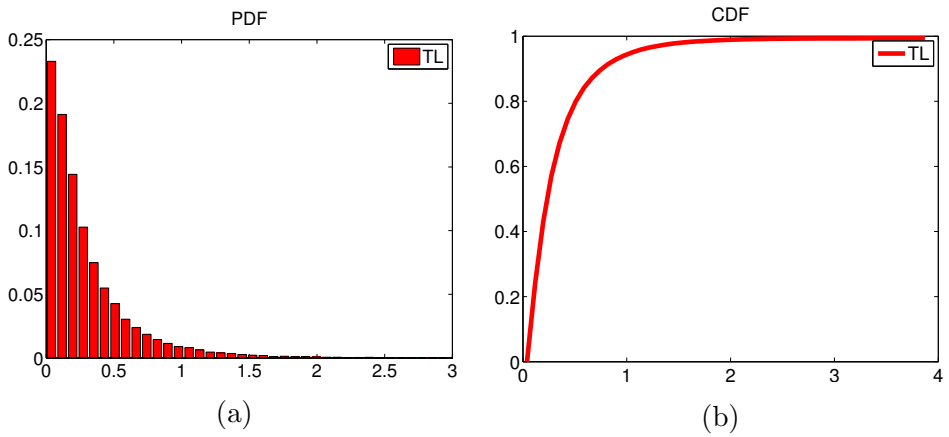


Figure 1.14: Train Station Dataset statistical results. The results suggest the effect of the rest of the people in a crowded scene has little impact in a target behaviour.

1.5 Conclusions

In this research, different modules for a fully-automatic pedestrian behavior analysis system are presented. The combination between these proposals led to a novel behavioral system based in information about the environment.

Tracking: Two different multiple-target tracking approaches are presented. The first one introduces the use of a hierarchical structure. A background subtraction in combination with two different trackers are used to track every possible target within a scene. A low-level tracker based in a velocity prediction using Adalines is used to track every isolated object, while the high-level tracker, which stores every object appearance using a fixed pool of L*a*b-space color histograms, is used to manage the occlusion and collision events. Experimental results in a public dataset (CAVIAR) proves both the accuracy and the low computational cost of this model (it can be used in a regular machine) under non-crowded scenarios.

Additionally, a high-level multiple-target tracking is presented to track people under uncontrolled scenarios. A combination of two different classifiers, a Viola-Jones and a SVM, are used to detect every person in the scene because of the head-shoulder omega-shape feature. A background subtraction technique is used to restrict the image location in which the classifiers are used, and also to perform a torso estimation to confirm the positive solutions. Two different trackers are used. A feature-based particle filter system, in combination with a velocity prediction system based in linear filters, is used to track the head-shoulder shape along the frames. A high-level tracker, which stores every people appearance into a fixed pool of L*a*b-space color histograms, is used to recover identifications which are previously lost.

show the improvement of this methodology with respect to other similar people detection techniques. The procedure speed indicates that is possible to obtain a real-system framework using parallelism techniques. This approach highly improves processing times of previous approaches to this topic. Since the techniques used in this methodology are mainly computed pixel per pixel, the inclusion of GPU programing techniques could derive in a real-time system for tracking people under crowded scenes.

Behavior Analysis: Contrary to classic state-of-the-art approaches, a novel methodology to detect abnormal behavior is presented. Based in a set of hypothesis, a new technique based in information about the environment is provided. Requiring only the trajectory of each target, a minimal path approach is used to successfully determine whether a path is abnormal or not. Our system is able to operate with partial information, that is, multiple fragments can be used to model a complete path. It requires constant memory, and it can be easily updated with the information that is constantly provided by the tracking system. Different potential images were defined to be used in our minimal path algorithm, proving that our system can be either run with or without any information a priori about the environment.

Although our first approach contains different limitations, it was improved with successive proposals. Thus, the model was extended to work under directional forces like traffic analysis, where the orientation of the motion is also important. The complexity of the distance metric was reduced from $\mathcal{O}(N \log N)$ to $\mathcal{O}(1)$ by introducing a novel distance map that can be computed in parallel with the minimal path surface. Finally, velocity was also introduced in order to create a temporal surface, enabling our model to work under crowded scenes.

This thesis demonstrates our new context-based behavioral model introduces simplicity in the definition of trajectory behavior without reducing its quality results. The improvements show this model is a fast an easily scalable approach to model pedestrian behavior.

1.5.1 Future Work

Related to the tracking improvements, we propose to parallelize our high-level multiple-target algorithm (section 1.4.2) in order to it them useful in real-time scenarios. Additionally, we propose to use the most recent deep learning techniques to create a new target appearance vector for re-identification.

As mentioned in section 1.4.5, the inclusion of directional forces resulted in an increase in the computational cost of the algorithm (from $\mathcal{O}(N \log N)$ to $\mathcal{O}(\Upsilon N \log N)$).

Thus, we propose to develop a new one-loop algorithm to solve the Hamilton-Jacobi equation, while maintaining the FMM complexity.

What is more interesting is what was achieved in the last contribution on this thesis (section 1.4.7). We conclude that, even in crowded scenes, the model behaves correctly. What is also important, the model is able to detect failures in the tracking system. Thus, we propose to change the established state-of-the-art behavior system structure to an hybrid one. That is, instead of using the tracking information to model the pedestrian behavior, we propose to use the context-based metrics to improve the tracking system.

Related to the latter, we also propose to use the minimal path model to build a long-time prediction system. The new architecture will try to answer the question: if a target is lost during the tracking procedure, where can it reappear after a certain number of frames?

Chapter 2

Tracking Published Papers

2.1 Journal Paper: Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Marcos Ortega Hortas
Affiliation: Universidade da Coruña, Spain

Co-author #3: Alba Fernández Arias
Affiliation: Universidade da Coruña, Spain

Co-author #4: Manuel Francisco González Penedo
Affiliation: Universidade da Coruña, Spain

Article title: Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios
Journal: Expert Systems with Applications
Volume: 40(4)
Pages: 1116–1131
Editorial: Elsevier
ISSN: 0957-4174
Year: 2013

Expert Systems with Applications 40 (2013) 1116–1131

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios

B. Cancela, M. Ortega *, A. Fernández, Manuel G. Penedo

Varpa Group, Department of Computer Science, University of A Coruña, Spain

ARTICLE INFO

Keywords:
Multiple-target tracking
Adaptive filters
Collision detection and management
Appearance computation

ABSTRACT

Multiple-target tracking is a challenging field specially when dealing with uncontrolled scenarios. Two common approaches are often used, one based on low-level techniques to detect each object size, position and velocity, and other based on high-level techniques that deal with object appearance. None of these methods can deal with all possible problems in multiple-target tracking: environment occlusions, both total and partial, and collisions, such as grouping and splitting events. So one solution is to merge these techniques to improve their performance. Based on an existing hierarchical architecture, we present a novel technique that can deal with all the mentioned problems in multiple tracking targets. Blob detection, low-level tracking using adaptive filters, high-level tracking based on a fixed pool of histograms and an event management that can detect every collision event and performs occlusion recovery are used to be able to track every object during the time they appear within the scene. Experimental results show the performance of this technique under multiple situations, being able to track every object in the scene without losing their initial identification. The speed processing is higher than 50 frames, which allows it to be used under real-time scenarios.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays, object behavior detection is one of the most promising techniques in the field of automatic video-surveillance systems. These techniques have a huge variety of applications like detecting abnormal movement over crowded scenes, or monitoring isolated objects, within a scene, to study their behavior.

The quality of these techniques largely depends on the tracking algorithm used. It is crucial to detect every object of interest within the scene. Target tracking is a problem far from being solved. Although there are many techniques that are able to detect isolated objects, when they have to manage multiple objects their performance decreases. There are two huge problems which have to be solved in order to obtain a good multiple target-tracking algorithm: long time occlusions and collisions.

A collision occurs when two or more objects interact within the scene. The most usual is a trajectory cross, which often causes the objects to be overlapped. This is a hard problem to be solved, because tracking systems tend to detect only one object during the cross. These algorithms have to develop a collision routine in order to detect every object involved in the event and be able to, after the collision expires, to successfully identify all of them.

On the other hand, long-time occlusions could happen in multiple situations, such as an object leaving the scene and re-entering later or a partial occlusion caused by fixed objects between the camera and the moving object, like a traffic light, for instance. Also a bad blob detection technique, could cause this problem. The difficulty of this issue is to identify the same object again after the occlusion.

There are many different approaches in the literature for object tracking. They can be classified into two big groups: low-level approaches and high-level approaches. Low-level approaches track every object within the scene without taking any appearance properties. They track the position of each object and match every tracking object between frames based in prediction techniques.

In Haritaoglu, Harwood, and Davis (1998), the W^4 algorithm employs second order motion models to predict the position of each tracking object between frames. It is based in the fact that, during people tracking, the motion of each person is relatively small with respect to the frame rate, which causes little changes in the silhouettes between frames. The estimation involves a previous median filter to avoid the noise. The system can track every person even under occlusion events, but fails when it has to deal with collision events. Other similar methods, like Pfinder (Wren, Azarbayejani, Darrell, & Pentland, 1997) are based on Gaussian distributions. This representation has multiple limitations when dealing with occlusions and collisions because, during a collision event, the representation of two different tracking objects are merged,

* Corresponding author.
E-mail addresses: braiscancela@udc.es (B. Cancela), mortega@udc.es (M. Ortega), alba.fernandez@udc.es (A. Fernández), mgpenedo@udc.es (M.G. Penedo).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.08.025

making it very difficult to identify each object after the occlusion using only position properties. In objects with low changes of movement, like cars,it is possible to obtain a correct identification using a prediction technique, but this is not feasible when dealing with objects quickly changing their orientation. One example of this situation are the human beings.

More complex methods are based on Mixture Models. In Stauffer and Grimson (2000), a technique based in a Gaussian Mixture Model is used to make a blob subtraction. The advantage of this method is that it is updated at every time step, so no time to train is needed. However, this update causes a bad estimation if the tracking object stops. After a few iterations, the tracking object will be set as part of the background. These problems also occur in Jepson, Fleet, and El-Maraghi (2003), where a recursive update of a Gaussian Mixture is used to track human faces. It can cope with partial occlusions, but fails when dealing with both total occlusions and collisions, since it is implemented to track isolated people.

Techniques based on predicted position are used in this topic too. Rohr (1994) introduces a bunch of Kalman filters in order to predict the target position under noise conditions. Huang et al. (1994) use an optical flow detection technique to detect vehicles in an automatic surveillance system. These methods fail when dealing with object collision, and when the tracking object stops, causing a missing target.

Kernel density estimation can track both occlusions and collisions (Elgammal, Duraiswami, Harwood, & Davis, 2002). However, it has to make some assumptions and restrictions to obtain good results. They assume that all the tracking objects are isolated the first time they appear on the scene, something that does not always happen in real surveillance systems. Also the high memory requirements and the computational complexity makes the algorithm useless under real-time tracking. An approximation to this technique can deal with real-time systems (Han, Comaniciu, Zhu, & Davis, 2008). However, this particular approach does not take into account multiple targets.

As we can see, low-level trackers are a good choice to track an isolated target. They can deal with partial occlusions, but fail when total occlusions or collision occurs. On the other hand, high-level approaches try to learn complex templates a priori in order to do pattern matching. They are mainly focused on target appearance. Collins, Liu, and Leordeanu (2005) use a pool of 'best feature' histograms to identify each tracking object. Each histogram is a linear combination in the RGB-space color, and the quality is obtained with a comparison against the near background. This method only takes into account single targets, and the comparison against the background makes the pool change quickly under heterogeneous background, which is an undesirable effect.

In Comaniciu, Ramesh, and Meer (2003), the mean shift is presented. It is a powerful technique that can locate each target by performing a gradient-descent search on an image region of interest. However, this method does not take into account multiple targets, and the initialization is not automatic. Also the appearance is never updated. Nummiaro, Koller-Meierb, and Van Gool (2003)introduce a particle filter based on color-histogram. No multiple target is considered, and the observation is computed using the predicted position, which introduces errors in the matching. On the other hand, occlusions are considered and solved under isolated tracking conditions. In Xing, Ai, and Lao (2009), a particle-filter is used to track every moving object, while it is stored in a temporal sliding window. Occlusion problem is solved using a detection response, which creates a set of potential tracklets. Tracklets from the particle-filter are associated with these potential tracklets by the Hungarian algorithm. Speed algorithm is low to be used in real-time applications.

Li, Zhang, Huang, and Tan (2009)track every person within a scene locating the omega-shape as a result of the head and shoulders pattern. A Viola–Jones system is trained and the system obtains good results in tracking isolated people. However, it has problems dealing with occlusions and collisions. In some cases, the mean-shift algorithm used cannot show differences between tracking objects. More generally, MacCormick and Blake (1999) use probabilistic methods to track any shape they want, but do not take into account collision scenarios.

Wu and Nevatia (2006, 2007) use a body-part scheme to locate every human in the scene. This method can deal with total and partial occlusions, but the algorithm speed is very low to be used in real-time scenarios. In Zhang, Li, and Nevatia (2008), a network flow augmented with an Explicit Occlusion Model is used. Although the algorithm speed is fast, a frame window is needed, which demands high memory requirements in real-time scenarios. In this case, a sliding window is used to obtain good results. A similar idea is performed by Song, Jeng, Staudt, and Roy-Chowdhury (2010).

Both low-level and high-level trackers have advantages and disadvantages. There is no method that can deal with all the problems we mentioned before. So, a proper mixture of these techniques is one option. Rowe, Reid, González, and Villanueva (2006) introduced a hierarchical architecture to cope with all problems, as in Huang, Wu, and Nevatia (2008) and Li, Huang, and Nevatia (2009). A bunch of Kalman filters are used to predict each target position, while a bunch of histograms model its appearance, using the method explained in Collins et al. (2005). Both trackers work in parallel and they are needed to perform a match. An event management is also used to detect the collisions produced by the targets. The algorithm can deal with occlusions and collisions in homogeneous scenarios with the absence of noise, but the id recovery after a collision event in heterogeneous scenarios performs bad results.

In this work we present a new strategy for multiple-target tracking that can solve part of the problems presented in previous hierarchical approaches. In first place, a blob detection is performed. Different background subtraction algorithms were tested, and their application depends on the scenario conditions. With the blobs detected, they are modeled using both an ellipse representation to perform the position and size, and a pool of histograms to model its appearance. Contrary to the method explained by Collins et al., we use a fixed pool of histograms to improve the results under non-homogeneous background scenarios. The use of alternative space color models is needed.

Two different trackers are used, but not in a parallel way. Low-level tracker is used when dealing with isolated targets, which obtain better results than other techniques. A new prediction technique based on Adalines is presented, which is more stable than Kalman filters in occlusion cases. Additionally, the pool of histograms is updated every iteration.

A new collision technique based in the ellipse properties is presented in order to detect target collisions. During these events, the high-level tracker is used to cope with target id recovery. Once the collision finishes, the tracker can recover every id using the target appearance.

This system is able to detect every tracking object in the scene, and preserve its identification until it leaves the scenario. Our approach is particularly appropriate for real-time systems such as surveillance applications.

This paper is organized as follows: Section 1 shows different approaches to solve this problem, introducing our solution, whereas Section 2 describes the method used to detect every tracking object in the scene; Section 3 explains the trackers used, Section 4 explains the algorithm used to combine each tracker and to detect occlusion and collision events; finally, Section 5 shows some experimental results and Section 6 offers conclusions and future work.

2. Blob detection

First, we have to perform a technique to detect every moving blob within the scene. We need a technique that can enable us to obtain, for each blob, both spatial values like position and size, and appearance properties.

Hence, we decided to perform a background subtraction in order to detect all the moving blobs within the scene. We consider as moving blobs all the objects that are not considered as part of the background, even if they are stopped. Due to this condition, we decided to discard optical flow techniques. Different approaches to background subtraction are presented in the literature.

Horprasert, Hardwood, and Davis (2000) used a method based on a color background-subtraction. For each pixel, two different parameters, chromaticity (CD) and brightness (α), are computed and normalized using linear combinations of RGB color-space statistical properties. Using four different thresholds, a mask image M indicates the type of each pixel. Four different categories are presented: original background, highlighted background, shadow or moving foreground. A pixel is classified as moving foreground if the chromaticity is different from the expected or if the brightness is lower than a threshold $\tau_{\alpha lo}$. Although the other thresholds can be selected using an automatic method, $\tau_{\alpha lo}$ needs to be chosen manually. Unfortunately, this parameter is very dependent on the lighting condition, which varies in every scene. Thus, a bad calibration of this threshold introduces a lot of noise.

Maddalena and Petrosino (2008) use a self-organizing map to model the background. Each background pixel is represented as a set of $n \times n$ weight vectors, typically $n = 3$. The method obtains good results, but the processing time is too high to be used in real-time systems.

Stauffer and Grimson (1999) store the background as a Mixture of Gaussians (MoG). A bunch of Gaussians, typically three, are used to define each background pixel. If a pixel value does not fit into any of the background distributions, it is considered as foreground until it is included into a Gaussian with enough evidence. Thus, this is a dynamic model which is updated at every frame using a parameter α. In order to maintain stopped objects as part of the background, α value must be very low. Even the initial model runs in the RGB-color space, it can be used in others, like L^*a^*b, obtaining good results under sudden illumination changes. However, a lot of problems are introduced when dealing with objects whose colors are similar to the background. It cannot handle them.

Kim, Chalidabhongse, Harwood, and Davis (2005) use a codebook. Each background pixel consists of one or more codewords. Not all pixels have the same number of codewords. The brightness, frequency of occurrence, the longest interval during the training period in which the codeword has not occurred and the first and last access time are stored into each codeword. Testing the difference between the current image and the codebook we can detect the foreground pixels. This method can be used in a dynamic or a static way, depending on if we want to update the background model or not, respectively. In our case, as we explained before, we choose the static model. Doshi and Trivedi (2006) combine the codebook model with the shadow suppression in HSV-color space (Cucchiara, Grana, Piccardi, Prati, & Sirotti, 2001) to improve the methodology.

Each method has been tested. As we explained before, the self-organizing map was discarded. The method used by Horprasert needs to tune manually an extra parameter, which produces poor results if it is not chosen correctly (Fig. 1(b)). Both MoG and codebook have good results, as we see in Fig. 1(c) and (d). Since we can use the codebook in a static way and it is faster than the MoG, we selected it as our background subtraction method. Contrary to the Doshi et al. approach, we perform the codebook in the YCbCr-color space and, after that, we perform the shadow detection performed by Cucchiara et al. We proceed that way because our tests show that the background subtraction obtains better results in the YCbCr-color space rather than the HSV-color space, but the shadow detection is easily detected in HSV. So, we combine these color spaces to obtain a better result.

In order to remove noise in the background subtraction, we perform the following procedure: first, we remove isolated pixel. Having a foreground pixel, we check the neighborhood. If there is no pixel marked as foreground in the top and in the bottom or in the left and in the right, we remove it. Basically, we remove every pixel which is part of a 1-pixel thick object. Second, an opening morphological operator is used in order to fill the blobs. Finally, a minimum-area filter is used. Fig. 2 shows the results after applying noise removal.

3. Tracker pool

As mentioned before, two different trackers are used to cope with the problems explained in Section 1: a low-level tracker, which is going to be used to track every isolated object, and a high-level tracker, which is necessary when dealing with occlusion or collision events. This section introduces our proposal.

3.1. Low-level tracking

Once the background subtraction is made, we group every pixel into blobs and we represent it as an ellipse. There are many other possibilities to define each blob, however, we find this representation good enough to identify each object. Moreover, the ellipse properties will be useful when dealing with object collision. Thus, for every j-observed blob at the time t is represented as

$$z_j^t = \left(x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t\right), \tag{1}$$

where x_j^t and y_j^t are the ellipse centroid coordinates, h_j^t and w_j^t are the size of the major and minor axes and θ_j^t represents the ellipse orientation.

Having five components identifying each blob, each object can be tracked around the scene. Our next goal is to identify the same object between frames. In this scenario the blob detection could lose a blob under a few frames, due to bad background subtraction or because the blob does not have the minimum-area required to be considered. Hence, we propose a model to be able to predict a blob position, knowing its previous ones, without any information about the blob appearance, only the position over the frames. One of the most popular methods for prediction is the linear filters, particularly the Kalman filter (Kalman, 1960).

The Kalman filter is a recursive algorithm which is able to predict the position under noise conditions. It uses a target state and a transition model to obtain the expected position, and it is updated using the correction between that position and the real measurement obtained. It works in two steps: first, a prediction is made and second, the observed measurement is used to correct the filter. This method is used in many tracking systems. For instance, in Rowe et al. (2006), the target-state for an ellipse representation is defined by $z_j^t = \left(x_j^t, \dot{x}_j^t, y_j^t, \dot{y}_j^t, h_j^t, \dot{h}_j^t, w_j^t, \dot{w}_j^t, \theta_j^t\right)$, where $\dot{e}_j^t$, $e \in \{x, y, h, w\}$ represents the velocity of each component. The velocity of θ_j^t is not computed because it is considered as noise. Using a transition matrix which adds the velocity of each component to predict the new position and using both measure and process diagonal covariance matrices we obtain accurate results.

Fig. 3(a) shows the prediction in an increasing function with little noise. As we see after the time 25, the target is lost and the Kalman filter obtains goods results predicting the position of the

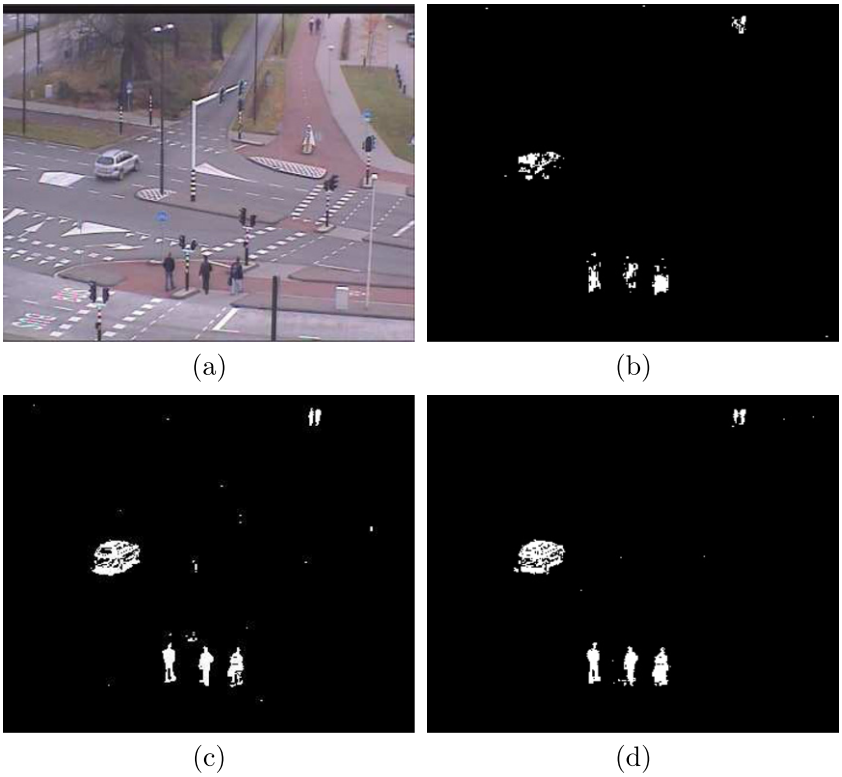


Fig. 1. Background subtraction: (a) frame, (b) Horprasert method, (c) MoG method, and (d) codebook method with shadow suppression.

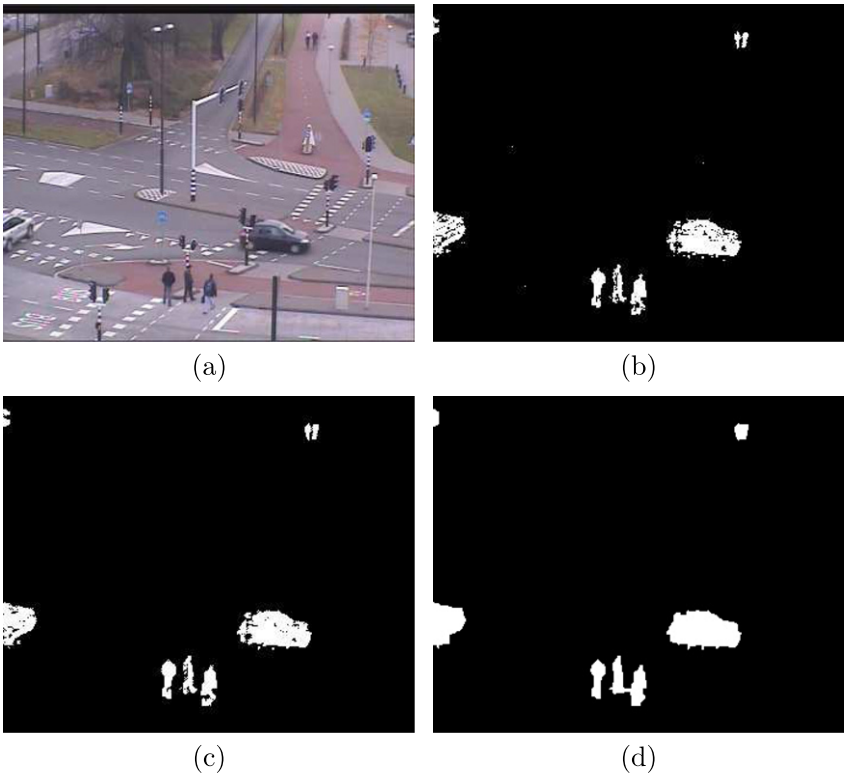


Fig. 2. Background subtraction processing: (a) frame, (b) codebook background detection, (c) 1-pixel thick removing, and (d) opening morphological operator.

x_j components. However, when dealing with the axes size h_j and w_j more problems arise. Usually, the size is always the same between frames, or it is increasing or decreasing slowly, if the blob moves toward or away, respectively. The problem is that both the head and the legs are too thick, which means that they can be erased during the background subtraction. This means that the size of

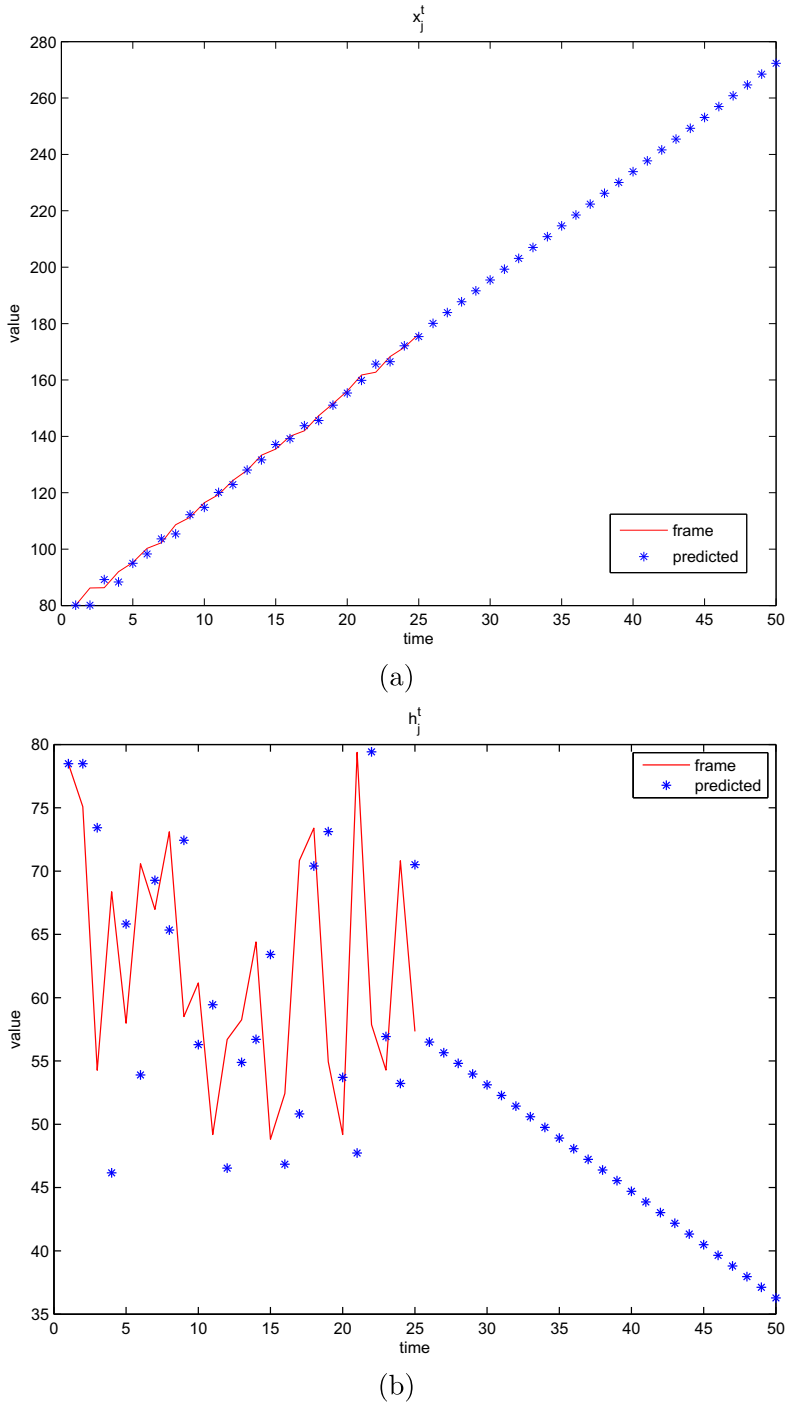


Fig. 3. Kalman filter predictions: (a) x_j^t component and (b) h_j^t component.

the ellipse could vary between frames, making the prediction really difficult to achieve. Fig. 3(b) shows an artificial example of the ellipse size. The average is 60, but the values vary quickly. When the target is lost, instead of maintaining the values between the range, it starts to decrease quickly, which is not desirable if we want to recover the tracking after a few iterations. So, our goal is to develop a new model to be able to cope with this problem.

Our approach also involves linear filters. However, instead of using the Kalman filter, we propose the use of adaptive filters (Adalines). An Adaline is a basic neural network consisting in M different inlets and only one outlet. Each Adaline has a weight vector $W = \{w_k\}$, $k = 0 \cdots (M-1)$ and the output is defined by:

$$O^{t+1} = W^t * I^t, \tag{2}$$

where O^{t+1} is the output value at time $t+1$, W^t is the $1 \times M$ weight vector and I^t the $M \times 1$ inlet vector at time t. The output value corresponds with the predicted position. As the Kalman filter, the weight vector is updated using the difference between the expected position and the new measurement. In our case, instead of trying to predict the position of each component, we only try to predict their velocities, adding them to the previous position in order to obtain the prediction. We also make the assumption that the velocity of the ellipse orientation θ_j^t is mostly due to noise. This is so because

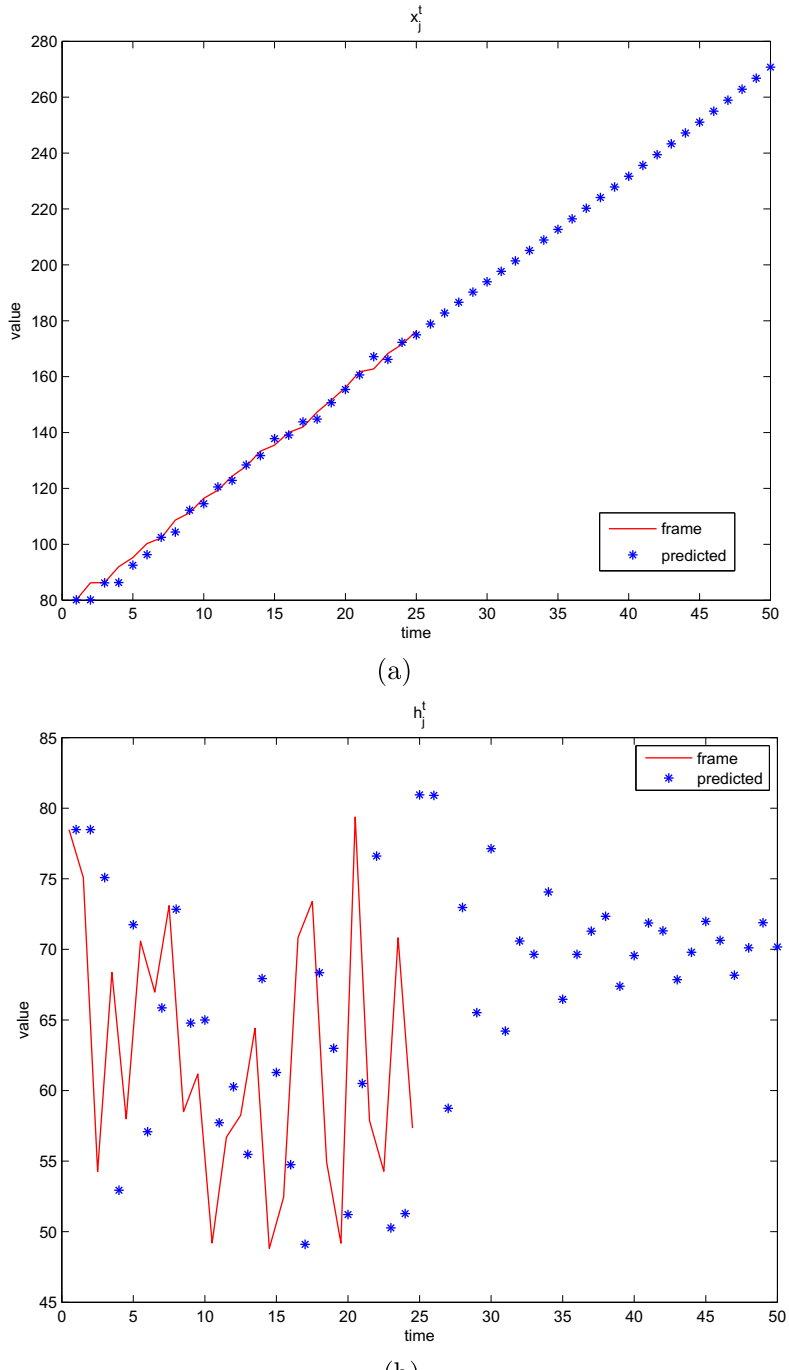


Fig. 4. Adaline filter predictions using a window $M = 3$: (a) x_j^t component and (b) h_j^t component.

little changes in the blob could cause huge changes in the orientation. Thus, we are not going to compute it.

Hence, we have four different Adalines, one for each velocity component. To calculate the vector I^t we have to store the $M + 3$ previous positions. Thus, the j-observed inlet velocity vector at time t is given by:

$$\dot{e}_j^t = \frac{e_j^{\theta} - e_j^{\theta-3}}{3}, \quad e \in \{x, y, h, w\}, \quad \theta = (t - M + 1) \cdots t, \tag{3}$$

where e represents each ellipse component except the orientation. Basically, what we are doing is a simple upwind finite difference scheme followed by a mean smoothing, which results in this equation. Then, the predicted position of the j-observed blob is given by

$$\tilde{e}_j^{t+1} = e_j^t + We_j^t * \dot{e}_j^t, \quad e \in \{x, y, h, w\}, \tag{4}$$

and, once we know the new measurement z_j^{t+1}, the equation to update the weights is

$$We_j^{t+1} = We_j^t + \alpha * \left(e_j^{t+1} - \tilde{e}_j^{t+1}\right) * \dot{e}_j^t, \quad e \in \{x, y, h, w\}. \tag{5}$$

Using this method we obtain a better approach to our goal than the Kalman filter. Fig. 4(a) shows that the response to an increasing function is similar to the Kalman filter. However, in Fig. 4(b) we can see that the prediction becomes stable after the target is lost, which will be very useful in the next steps.

Once we develop a method to predict the position of each blob, we need a metric in order to determine when a blob in a new frame

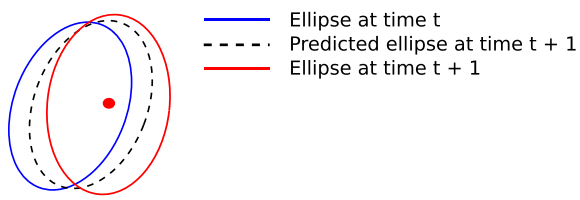


Fig. 5. A blob in a new frame is related to an early tracking object if its ellipse centroid fits within the predicted ellipse of the tracking object.

corresponds with an existing object which has been tracked before. There are different metrics that we can use to solve this question. For instance, the euclidean distance. In Rowe et al. (2006), they use a property of the Kalman filters, the innovate covariance matrix S_k, to perform the matching. In our experiments, all the values of S_k^{-1} tend to zero quickly. This means that a blob can be matched with an existing tracking object even if they are not close together.

In our approach we make an assumption, that every object in the frame moves slowly enough compared to the frame rate. This means that the blob in the new frame associated with an existing tracker must be really close to the predicted one. As we calculated before in Eq. (4), we have a predicted position of the j-observed blob at time $t+1$, $\tilde{z}_j^{t+1} = \left(\tilde{x}_j^{t+1}, \tilde{y}_j^{t+1}, \tilde{h}_j^{t+1}, \tilde{w}_j^{t+1}, \theta_j^t\right)$. As we said, we consider the velocity of θ_j^t as noise, so the predicted orientation is the same as the previous one. With these assumptions, we made the following hypothesis: it is highly probable that the ellipse centroid of the blob in the new frame associated with the j-observed blob fits within the predicted ellipse $\tilde{z}_j^{t+1}$. This is the reason why we need the predicted size of every blob to remain stable under occlusion events. If the size of the predicted ellipse grows quickly, the method would be useless.

Fig. 5 illustrates the case of a positive match. We have a blob tracked at time t, and we calculate the predicted position at time $t+1$. At time $t+1$, the ellipse centroid that matched with the tracking object fits within the predicted ellipse.

To determine if a point is within an ellipse we are going to use the ellipse properties. The ellipse $z = (0,0,h,w,0)$ has the following equation that satisfies every point in the boundary:

$$\frac{x^2}{h^2} + \frac{y^2}{w^2} = 1, \quad (x,y) \in \text{boundary}. \tag{6}$$

Particularly, every point within the ellipse satisfies that

$$\frac{x^2}{h^2} + \frac{y^2}{w^2} < 1. \tag{7}$$

So, having an ellipse $z^{t+1} = (x^{t+1}, y^{t+1}, h^{t+1}, w^{t+1}, \theta^{t+1})$, if we want to determine if it is matched with the j-observed blob, we make the following transformation:

$$\begin{vmatrix} x_z \\ y_z \\ 1 \end{vmatrix} = \begin{vmatrix} cos\theta_j^t & -sin\theta_j^t \\ sin\theta_j^t & cos\theta_j^t \\ 0 & 0 \end{vmatrix} \begin{vmatrix} 1 & 0 & -\tilde{x}_j^{t+1} \\ 0 & 1 & -\tilde{y}_j^{t+1} \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x^{t+1} \\ y^{t+1} \\ 1 \end{vmatrix}, \tag{8}$$

where (x_z, y_z) represents (x^{t+1}, y^{t+1}) under $\tilde{z}_j^{t+1}$ coordinates. If $\frac{x_z^2}{\left(\tilde{h}_j^{t+1}\right)^2} + \frac{y_z^2}{\left(\tilde{w}_j^{t+1}\right)^2} \leqslant 1$, the centroid is within the ellipse.

This method will also be used to detect different object collisions, which will be explained later. Once we have the metric to assign new frames to each tracking object, we are able to start the low-level tracker to track every object within the scene. To distinguish every blob tracked in the scene, a unique color is assigned to each one, which is also used as the blob id. The method works as follows: when a new frame arrives, the background subtraction is made, detecting the foreground blobs in the scene. Then, we compare each blob with the tracking objects obtained in previous scenes using both the Adalines to predict the new position and the ellipse centroid metric to matching. When a match is confirmed, the track is updated with the new position, and also the Adaline weight vectors values.

Weights of every Adaline are initialized to zero. In the first N step, we consider that the tracking object is not trained, so, to perform the match, instead of using the predicted position, we use the position in the previous frame. If a tracking object is lost during the training period, it is removed. If the difference between the frames since the tracking object appeared for the first time and the times the tracking object appeared in the scene is high, the tracking object is also erased.

Fig. 6(a) and (b) shows how the method can track object along the frames. However, this method only works under individual objects which do not produce collisions, such as splitting or grouping events. The background subtraction under a grouping event merges every object into the group, so we cannot classified each object individually during that event. Also, when the splitting event occurs, we have to correctly classify each object which was involved into the group, giving to them the same id that they have before the grouping event. So, another tracker must be developed, containing appearance properties for every object.

3.2. High-level appearance tracker

As mentioned before, we need other kind of tracker to deal with collision problems. A tracker based only in shape and movement of an ellipse cannot manage the case of two different objects which make a group and, after a few iterations, they separate its paths. The low level tracker cannot distinguish which objects correspond with the previous ones. Since we perform a background

(a) (b)

Fig. 6. Low level tracking example using ellipse centroid metric. Each ellipse has a unique color which is used as object id. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

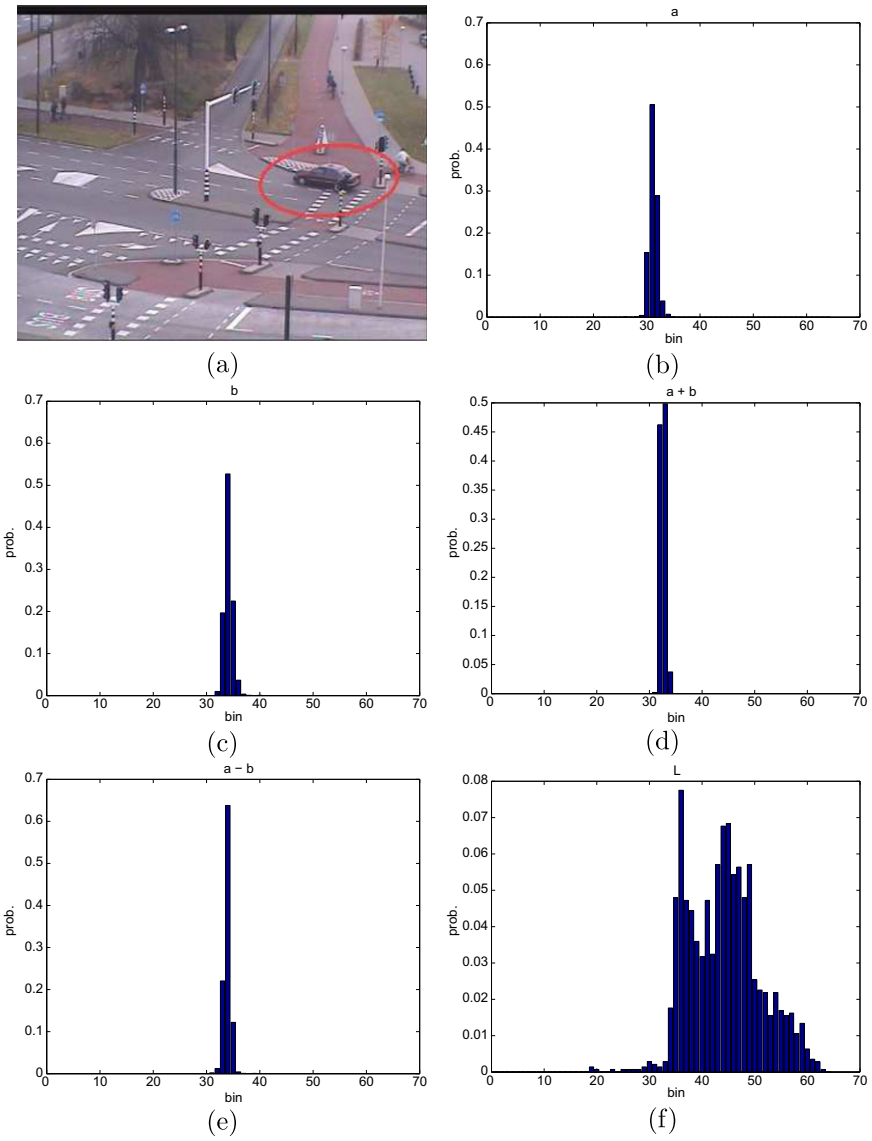


Fig. 7. Car feature selection: (a) object used, (b) a histogram, (c) b histogram, (d) $a + b$ histogram, (e) $a - b$ histogram, and (f) L histogram.

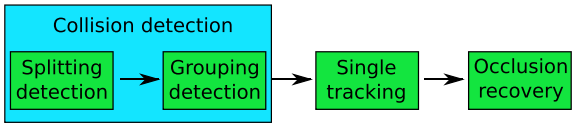


Fig. 8. Execution order.

subtraction, we are able to perform an appearance tracker in order to solve the situation mentioned before.

To deal with the problem mentioned before, Collins et al. (2005) proposed a method based on a pool of histograms. Using multiple combinations in the RGB-color space, they collect a pool of 49 different histograms. They choose a region containing the object to make the pool, and other region which wraps the object to perform a background histogram pool. Using a log-likelihood metric between the object and the background histograms, a smaller pool of best features is selected and compared to determine if the object corresponds to a previously tracked one. This method works because the background difference between frames is low enough to obtain, approximately, the same best histograms. Rowe et al. use the same system to perform their high-level appearance tracker. Additionally, a pool of historical 'best features' is maintained to use them in cases of object occlusion recovery.

This method obtains good results under homogeneous backgrounds with constant brightness. However, in the occlusion case we want to solve, this method does not work very well if the duration of a grouping event is high. As mentioned before, the best features are chosen comparing the object against the background. As the actual background we have after the splitting could be totally different than the previous one, the record of previous histograms may result to be useless. Also a sudden background change, i.e. from the road to the grass, could make a change in all the histograms in the best feature pool. To solve this, our goal is to obtain a fixed pool of histograms, which will be invariant to the background or the illumination. The first step is to define or validate a set of features enabling this invariance and the comparison between blob descriptors.

3.2.1. Feature selection

Under illumination changes, RGB-color space highly changes all its values, so we focus on perceptual color spaces which can isolate

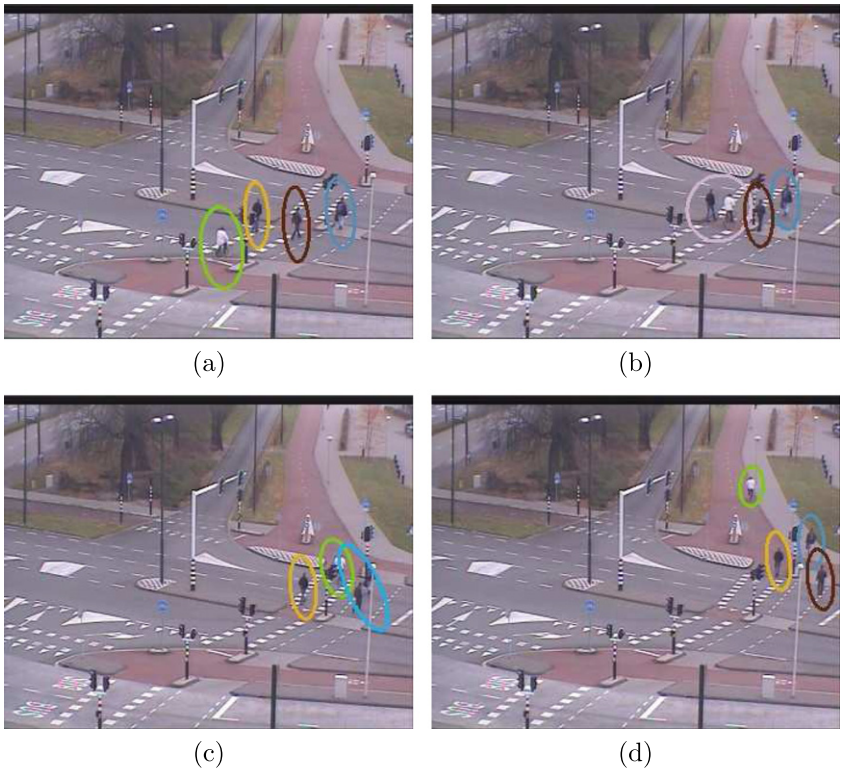


Fig. 9. Collision detection example. The system can detect grouping events end, once they finished, recover the previous identification of each tracking object.

the illumination into one component. So, we propose to use L^*a^*b, which is a color-opponent space with dimension L for lightness and a and b for the color-opponent dimensions. So we can isolate the illumination and work with the other components. Thus, the histograms we use are the following:

$$h = \omega_1 * L + \omega_2 * a + \omega_3 * b,$$
$$(\omega_1, \omega_2, \omega_3) \in \{(1,0,0),(0,1,0),(0,0,1),(0,1,1),(0,1,-1)\}.$$

We keep only one histogram with the illuminant component as a backup in caseswhere there is not enough evidence to make a decision with the other histograms. Features are normalized and discretized into 64 bins to perform the computation. A smaller number of bins could produce wrong matching, according to Rowe et al. (2006).

Fig. 7 shows an example of the histogram pool, specifically a car. For each ith-feature of the tracking object histogram is given by $\mathbf{p}^i = \{p_k^i;\ k = 1,\ldots,64\}$, and normalization ensures that $\sum_{k=1}^{64} p_k^i = 1$.

3.2.2. Appearance computation

Although L^*a^*b space guarantees small changes in the histograms under different illumination changes, every histogram must be updated in order to improve its quality. Hence, for each one of the five different histograms, its appearance is recursively computed. The mean appearance histogram of the ith-feature histogram in time t, $\mathbf{m}_t^i$, is

$$\mathbf{m}_t^i = \frac{n_i \mathbf{m}_{t-1}^i + \mathbf{p}_t^i}{n_i + 1}, \tag{9}$$

where n_i is the number of times the histogram has been computed. Then, we need another metric to establish the similarity between two different frames. In our case, we choose the *Hellinger distance*, which, like other f-divergence functions, is used to quantify the similarity between two probability distributions. The Hellinger distance between two different histograms p_k and q_k is

$$d_H = \sum_{k=1}^{64} \sqrt{p_k q_k}, \tag{10}$$

where a value of $d_H = 1$ indicates the histograms are the same. To establish an acceptance criteria, a threshold in the distance space could be established. However, it seems like a bad idea since the values change when dealing with different colors. Instead, we calculate both mean and variance of the Hellinger distance for each time step, and we use it as the acceptance criteria.

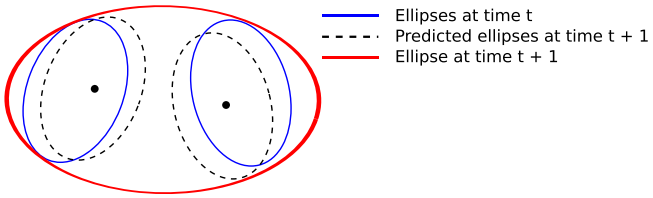


Fig. 10. A group in a new frame occurs when two or more predicted ellipse centroids of any tracking objects fit within the same ellipse in the new frame.

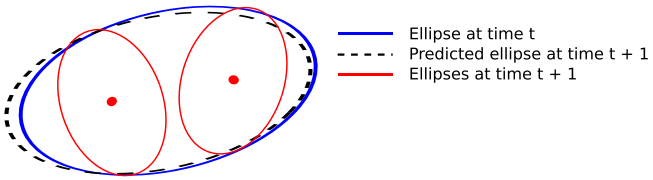


Fig. 11. A split in a new frame occurs when two or more ellipse centroid in the new frame of it within the same predicted ellipse of any tracking object.

4. Tracker combination for multiple-target tracking

As discussed in previous section, two different trackers are used in this methodology, both with advantages and disadvantages. Low-level tracker is the better choice when dealing with individual tracking objects. Using the ellipse centroid to make the comparison we obtain good results. However, it cannot deal with the problem of object collisions and/or occlusions. Moreover, high-level tracker can distinguish each tracking object due to its appearance. However, a high threshold value in the appearance computation could produce a high false negative rate. On the other hand, a low value will increase the false positive rate.

This makes that the high-level tracker to be a good choice in dealing with occlusions and/or object collisions. But the low-level tracker is better than the high-level tracker when dealing with isolated objects. Therefore, a fusion of these two trackers will be used to improve their individual performances. First, we need to detect every collision and occlusion over the scene to keep a correct record of objects in the scene and their locations.

Fig. 8 shows the order of the system execution. First, the collision detection module tries to find splitting and occlusion events. Later, the single tracking module finds matches for every single target with its corresponding object in the new object, only using the low-level tracker. Finally, the occlusion recovery module uses the high-level tracker to find the tracking objects which are occluded in previous frames.

4.1. Collision detection

The basic idea to detect collision events is to perform a collision map. However, the accuracy of this method is poor since it deals with a simple grid, decreasing the frame resolution. In our case, we use the low-level tracker properties to perform our collision event detector.

Two different cases are considered: grouping and splitting events. Eq. (7) offers a valid criteria to study the occurrence of these situations. Fig. 9 shows an example of how this method works. In the first image four different persons are detected into the scene. Later, group events occur during the video and, finally, the system is able to recover the id of each tracking object without any error. Although splitting detection runs before the grouping detection, the latter is going to be explained before because the splitting detection is based in the grouping detection algorithm.

4.1.1. Grouping detection

To avoid different noise conditions, we perform one restriction before we initiate the grouping detection: tracking objects which are not trained are ignored. Additionally, tracking objects which are occluded for a long time will also be ignored, because their predicted positions could highly vary from the real ones. Previously, we mentioned that every object moves slowly enough compared with the frame rate, so we can say that every predicted ellipse centroid will be within the ellipse that corresponds to each tracking object. Thus, we can make the same reasoning to say that, if two or more predicted ellipse centroids fit within a new ellipse in the scene, a new group is created containing all of the tracking objects involved. Fig. 10 shows a graphic explaining how this method works. More formally, if we have a blob in the new frame defined by the ellipse $z^{t+1} = (x^{t+1}, y^{t+1}, h^{t+1}, w^{t+1}, \theta^{t+1})$, we compute the grouping factor as

$$Gr_{z^{t+1}} = \left\{ \tilde{z}_j^{t+1} \in \Omega^t \left| \frac{\tilde{x}_j^2}{(h^{t+1})^2} + \frac{\tilde{y}_j^2}{(w^{t+1})^2} \leqslant 1 \right. \right\}, \tag{11}$$

where Ω^t is a set containing all the tracking objects at time t, except those we ignored, and $\tilde{x}_j^2$ and $\tilde{y}_j^2$ are the position of the $\tilde{z}_j^{t+1}$ predicted ellipse centroid under the z^{t+1} coordinates, using Eq. (8). So, if we have that $|Gr_{z^{t+1}}| \geqslant 2$ a new group is created and instantiated. Note

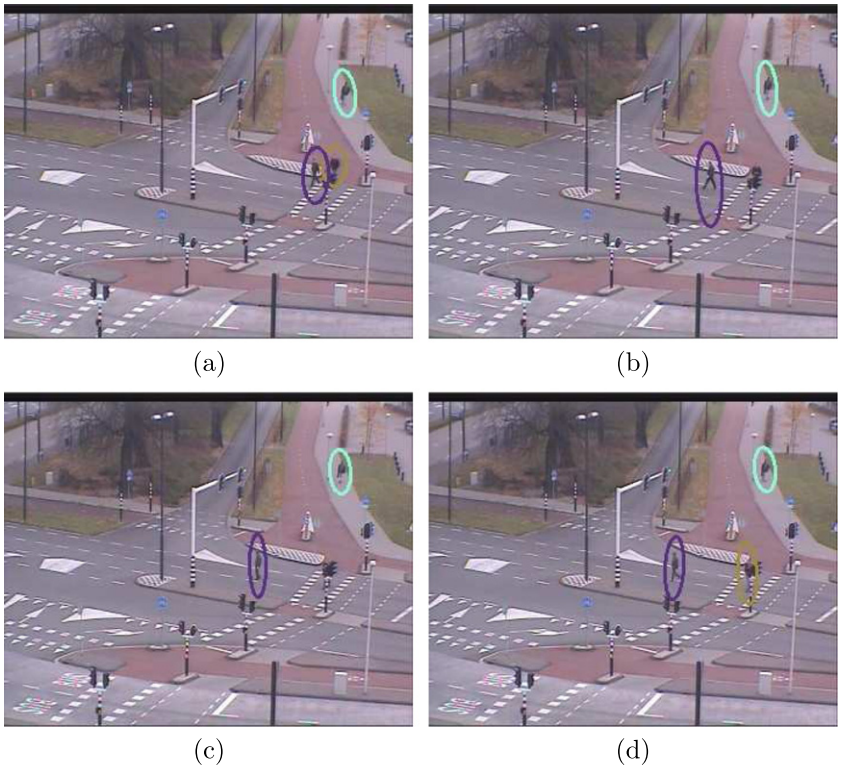


Fig. 12. Occlusion recovery. In (a), the person identified with the yellow ellipse is tracked. In (b) and (c), this person is occluded behind the traffic light, while in (d) is recovered again. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

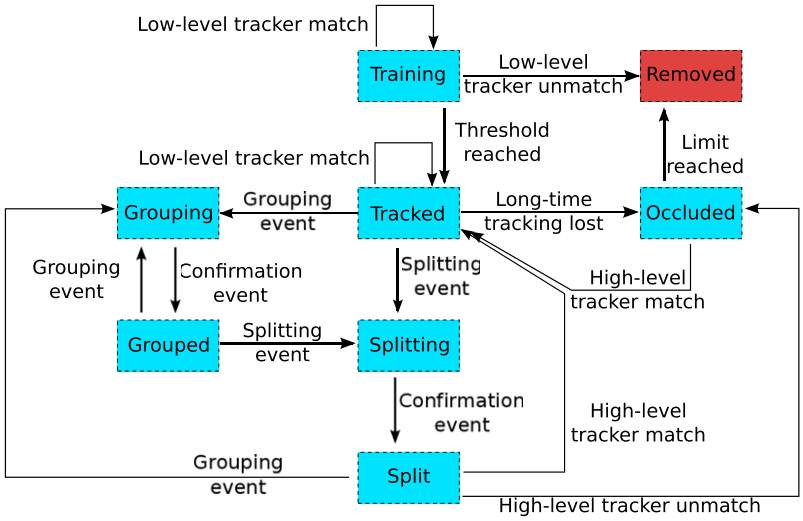


Fig. 13. State diagram of a tracking object which is not considered as a group.

that it is possible that one ellipse centroid of a tracking object could be within more than one different predicted ellipse. In that case, we only consider the ellipse in which the centroid is closer.

For every tracking object which is part of the group, we also store its size, which will be useful whenever the splitting event occurs.

4.1.2. Splitting detection

The behavior of the splitting detection works in the opposite way to the grouping event. Thus, if two or more new ellipse centroids fit within a tracking object predicted ellipse, a splitting event occurs. Fig. 11 shows a graphic explaining how this method works. More formally, if we have a tracking object with its predicted ellipse $\tilde{z}_j^{t+1} = \left(\tilde{x}_j^{t+1}, \tilde{y}_j^{t+1}, \tilde{h}_j^{t+1}, \tilde{w}_j^{t+1}, \tilde{\theta}_j^{t+1}\right)$ defined using Eq. (4), we compute the splitting factor as

$$Sp_{\tilde{z}_j^{t+1}} = \left\{ z^{t+1} \in \Omega'^{t+1} \left| \frac{x'^2}{\left(\tilde{h}_j^{t+1}\right)^2} + \frac{y'^2}{\left(\tilde{w}_j^{t+1}\right)^2} \leqslant 1 \right. \right\}, \tag{12}$$

where Ω'^{t+1} is a set containing all the ellipses in the scene at time $t+1$, and $x_j'^2$ and $y_j'^2$ are the position of the z^{t+1} ellipse centroid under the $\tilde{z}_j^{t+1}$ coordinates, using Eq. (8). Again, if we have that $\left|Sp_{\tilde{z}_j^{t+1}}\right| \geqslant 2$ a splitting event occurs and must be evaluated. The same case mentioned on the grouping detection about centroids that fit within two or more ellipses has to be considered.

When a splitting event occurs, two different possibilities have to be considered:if the tracking object involved is a group or not. If it is not a group, which happens when the group is created before the objects enter in the scene, a new tracking object is instantiated for every new ellipse involved in the splitting event. Note that it is possible that no splitting event occurred and the additional ellipses were due to noise. To solve this situation, we maintain these new tracking objects as provisional. If the split situation remains in the next frame, the split is confirmed.

If the tracking object involved is a group, then we remove it. As mentioned before, every group has a pool with all the tracking objects involved in the event. Thus, using the high-level tracker, we try to match every object in the new frame with the tracking objects stored in the group. We consider a match if three of the four color histograms pass the acceptance criteria based on Hellinger distance discussed in Section 3.2.2 and the size of the new object is similar to the tracking object size. If more than one tracking object passes the test, we consider those which have the best accuracy, including the luminance histogram.

If the size of one of the new objects is higher than the stored sizes, then we consider that object as a new group containing all the tracking objects which were impossible to match with the rest of the objects in the splitting event. On the other hand, if there are more new ellipses than tracking objects stored in the group, we create new tracking objects with the unmatched ones. There could be the case when a group splits in two or more different groups. While it is impossible using this method to determine in which group to fit the tracking objects stored in the previous group, we create the two groups including all the tracking objects. Then, a tracking object could also be removed if a split occurs in one of the groups and we could identify it. Thus, that tracking object is removed in both the groups.

Table 1
Test scenario statistics. Accuracy means how the target, being within the scene, is located. This includes both isolated tracking and tracking when it become as a part of a group.

Number of frames	2935 Frames
Number of tracked targets	33 Targets
Number of correct targets	32 Targets
Number of unallocated targets	0 Targets
Accuracy	98.09%
Processing speed	51.122 fps

4.2. Occlusion recovery

There are two different types of occlusion: total and partial. Partial occlusions are easy to solve since the tracking object quickly becomes visible again. Thus, using the low-level tracker to predict the position of the tracking object is enough to recover it. On the other hand, long-time occlusions make the low-level tracker useless, since the position is hard to achieve after a few frames.

However, low-level tracker will be a powerful tool in order to limit the possibilities. Once we recognize that a tracking object is occluded (after a few iterations), we use the velocity module to set a circular area centered in the last position in which the tracking object has been detected. This area will grow in time until we obtain a positive match. Thus, we consider all the blobs in which

the ellipse centroid in the new frame has not been matched with any of the visible tracking objects. Moreover, we consider all the blobs which could also be matched with a tracking object which is not trained yet. If an ellipse centroid of these blobs meets any of these conditions and fit within the circular area, the high-level tracker is activated. In this case, we have to reduce the possibilities of a wrong match, so we accept that the high-level tracker obtains a match when all the color histograms pass the test using the Hellinger distance. Fig. 12 shows an example of a tracking with occlusion recovery. In this sequence, a traffic light eventually hides the person, making the tracking impossible. Afterward, the tracking object reappears and the system is able to recover its previous id.

4.3. Object state management

Fig. 13 shows a tracking object state diagram, indicating how the transitions are made. Seven different states are defined: training, tracked, grouping, grouped, splitting, split and occluded. When a tracking object is detected for the first time, its state is set to *training*. In this state, the low-level tracker is used to detect the tracking object in successive frames. If, during this state, the low-level tracker could not find a positive match, the tracking object is removed. If the number of consecutive positive matches is higher than a threshold, the tracking object changes its state to *tracked*.

When the state of the tracking object is *tracked*, only the low-level tracker is used to perform the match. If a grouping or a splitting event occurs, the state is changed to *grouping* or *splitting*, respectively. On the other hand, if it is impossible to match the tracking object with any ellipse during a few iterations, we mark this tracking object as untraceable and we change its state to *occluded*. If no collision event occurs and the low-level tracking finds a match, its state remains the same.

If the state of the tracking object is *occluded* it means that we cannot locate it within the scene during a few iterations. Tracking objects marked as *occluded* are the last objects of being processed, because there are less possibilities to find them than the others. Thus, when all the other tracking objects are processed, we check the remaining ellipses in the frame which have not been matched. If the occlusion recovery module obtains a match using the high-level tracker, the Adalines are reset with the new position and its state is changed to *tracked*. If the total time the tracking object appears in the scene is much lower than the total time since it appeared for the first time, the tracking object is removed.

Both states *grouping* and *splitting* are transitory states. If, in the next frame, grouping or splitting events are confirmed, its state is changed to *grouped* or *split*, respectively. Otherwise, its state is changed to the state it had before the collision event occurred.

In *grouping*, the tracking object stores its size with respect to the group. Nothing happens in this state until another collision event occurs in the group that contains the tracking object. Two different possibilities may occur: another grouping event, which increases the number of members in the group and change every state to *grouping*. On the other hand, in a splitting event occurs, its state is changed to *splitting*. This could also happen if we find that the group becomes occluded. After the occlusion, it is impossible to predict if the group remains immovable, so it is preferable to remove it and mark all members as *occluded*.

Split is another transitory state. In these cases, the high-level tracker is activated and, if a positive match occurs, the Adaline weights are reset and its state is changed to *tracked*. If no positive match occurs, its state is changed to *occluded*. Finally, it is possible that, when the split event takes place, one member could form another group immediately. This is the reason why the splitting detection module always runs before the grouping detection module. If this happens, its state is changed to *grouping*.

Contrary to Rowe et al. (2006), in our case we do not use both trackers at same time. Low-level tracker is responsible for tracking every isolated object in the scene, while the high-level tracker is only used when the low-level tracker cannot be used, such as splitting events or occlusions.

5. Experimental results

In our experiments we have used the CANDELA Intersection Scenarios in order to test the methodology. Over 2900 frames were used, including occlusions and multiple collisions, such as

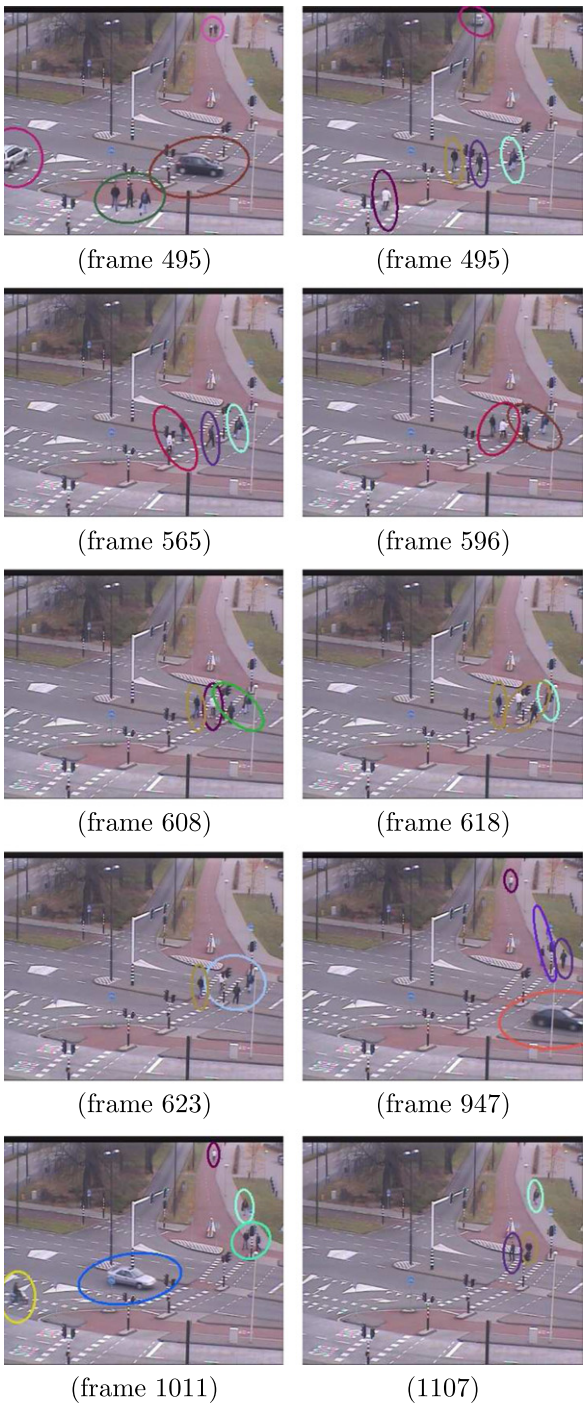


Fig. 14. Quick succession of people collision events. The system is able to recover the previous id of all the four people involved in this collision.

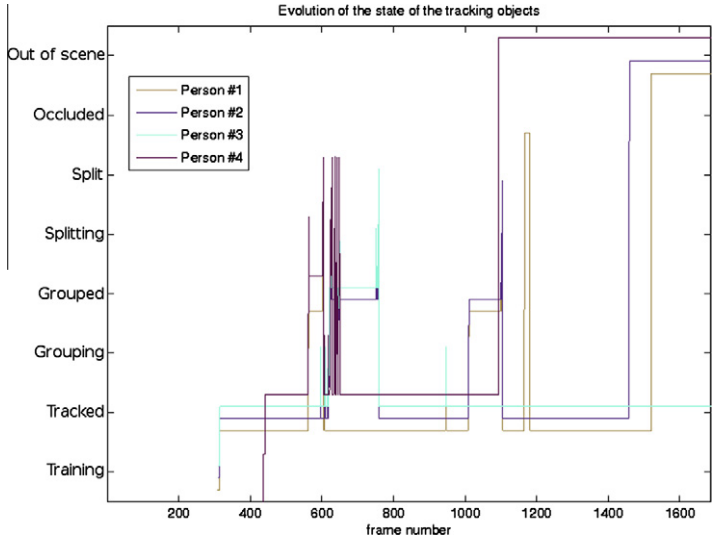


Fig. 15. State evolution of the four people showed in Fig. 14. Multiple grouping and splitting events occur around the frame 600. Also the person #1 becomes occluded near frame 1200, while the system is able to recover it when it appears again in the scene.

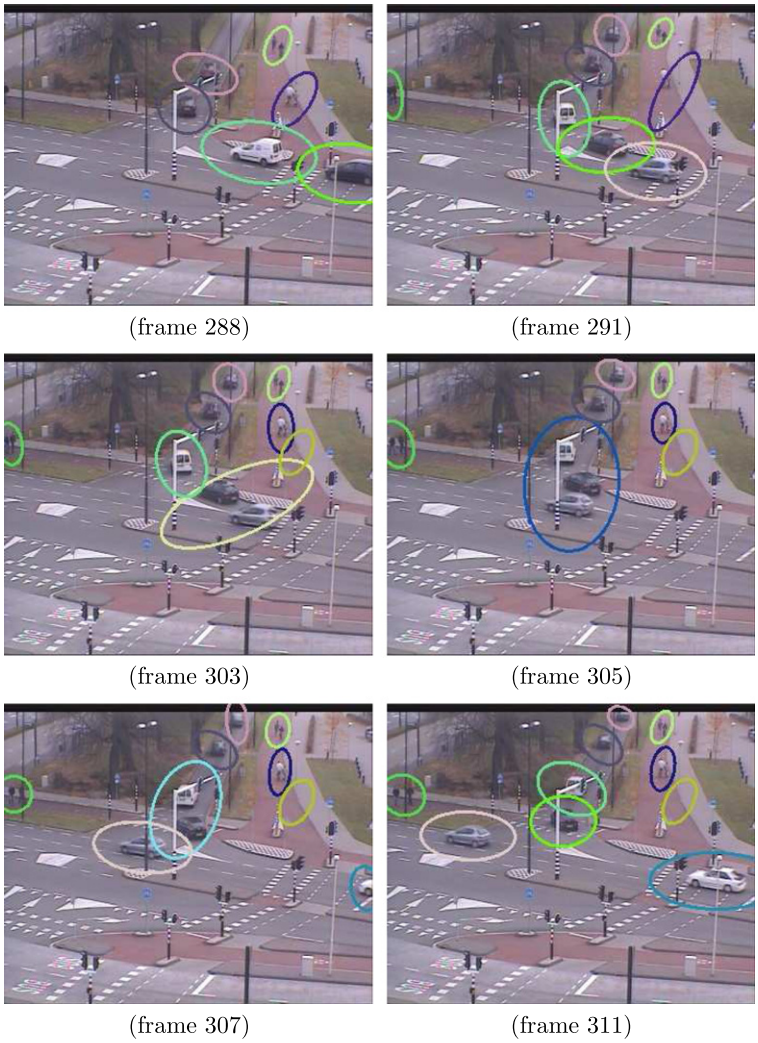


Fig. 16. Quick succession of car collision events. Every car in the event recover its identification when it become isolated.

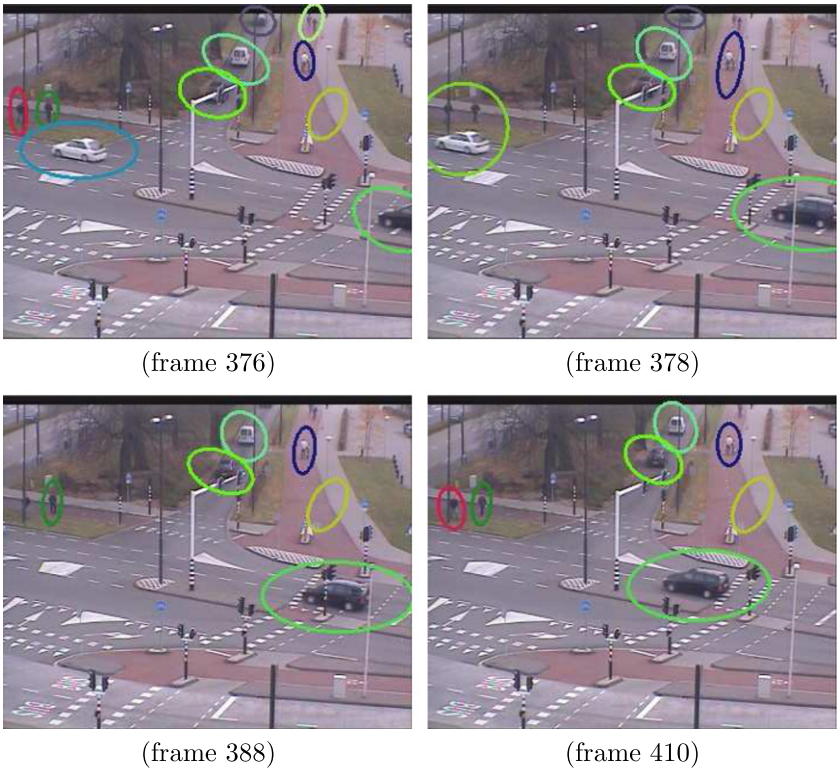


Fig. 17. Occlusion example after a grouping event. The person identified with the red ellipse becomes part of a group with a car. Before the group dissolves, the person is hidden. The algorithm dissolves the group and, after a few iterations, can recover the person identification. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Test scenario event statistics. The system can detect and process every grouping event. Splitting events are detected and processed in a correct way expect when the tracking objects are on the point of leaving the scene. The occlusion recovery system has the same problem.

	Total	Correct	%
Grouping events	17	17	100
Splitting events	19	18	94.74
Occlusion recovery events	5	4	80

Table 3
CAVIAR dataset results. Every paper uses corridor camera to obtain the results. However, no frames without persons exist, so it is impossible to train the background subtraction algorithm. Instead, front camera is used. Good results are achieved, having in mind our system can be used in real-time scenarios.

	GT	MT (%)	PT (%)	ML (%)	IDS	FRAG
Wu and Nevatia (2006)	144	72.22	23.61	4.15	13	42
Wu and Nevatia (2007)	189	74.07	21.69	4.24	19	40
Zhang et al. (2008)	140	85.71	10.71	3.58	15	20
Huang et al. (2008)	143	78.30	14.70	7.00	12	54
Xing et al. (2009)	140	84.28	12.14	3.58	14	24
Li, Huang, et al. (2009)	143	84.60	14.00	1.40	11	17
Song et al. (2010)	75	84.00	12.00	4.00	8	6
Ours[a]	110	90.00	7.26	2.74	8	16

[a] Front camera is used instead of corridor camera.

grouping and splitting events, as well as target entering and exiting the scene. These videos take place outdoors, in an intersection. Although the light conditions in these videos are good, these scenarios have a lot of complexity. There are multiple object interactions in a short space of time, which implies the system has to be robust against collision events. The first 15 frames of each video were used to train the codebook, which are enough to obtain a good background subtraction and blob detection.

The sequences used in this algorithm test involves isolated people, groups and vehicles. Table 1 shows the video statistics. The accuracy of the algorithm is high, since it is able to locate every target within the scene and, including total occlusions, the mean time in which each object is lost is lower than 2%. It is noted than an incorrect target is detected by the system because of the training process. In one of the videos there are no frames in which there are only background elements. So, when the codebook is trained, there is one position with a moving object. When the codebook is trained and the moving object leaves that position, a blob is detected there as part of the foreground. This error does not happen with enough background frames to train the codebook.

The problem of detecting collisions when two tracking object are moving in opposite direction can be easily solved, since there is only one grouping event and one splitting event. The system solves this situation rapidly. The challenging situation is dealing with tracking targets that are moving really close together in the same way. In this case, there is a quick succession of collision events, which implies that only one fail could ruin all the correct matching.

Fig. 14 shows an example of four people walking in the same way. The blob subtraction algorithm often merges two or more tracking objects. In Fig. 15 we can see the state of each person. Around frame 600 we clearly see that person #2, person #3 and person #4 quickly change their states to become isolated and as part of a group. Six grouping events and five splitting events occur in less than 100 frames, without counting the uncommitted changes. In frame 623 we can see a group that involves three different people. In the next image, at frame 947, one person leaves the group, and the system is able to detect which person abandons

(a) Frame 492 (b) Frame 589

(c) Frame 649 (d) Frame 1056

(e) Frame 1196 (f) Frame 1337

Fig. 18. CAVIAR dataset example. The system can detect both grouping and splitting events, and also long-time occlusions.

it. Also, the algorithm is able to detect the other blob as a new group that involves the rest of the people in the previous one. Also, in Fig. 14, the last image shows that the algorithm is able to detect every collision event in this group, so finally it can recover every previous target identification.

The occlusion case also occurs in that scene. As we can see in Fig. 15, around the frame 1200 the person #1 becomes occluded. The reason is that a traffic light interferes between the person and the camera. After the person reappears, the high-level tracking obtains a match between the tracking object and the new blob, restoring the previous identification.

Fig. 16 also shows a similar example involving cars. This time the collisions are easy to solve because the bigger a tracking object is, the harder to lose it during the blob detection. Also, when a splitting event occurs, it is easier to detect if one of the new blobs corresponds to another group, since the difference between the group size with respect to each car involved is high. In frame 307 we can see an example of the algorithm accuracy in these cases. After all the collision events occur, in frame 311, all the previous ids were recovered.

Another different case is shown in Fig. 17. In frame 378 a group is created, which involves a person, identified by a red ellipse, and a car identified by a blue ellipse. Before the splitting event occurs, the person becomes occluded behind a streetlight. Then, in frame 388 the car leaves the scene. So, our method removes the group and, in frame 410, is able to recover the previous identification for that person.

To summarize the whole set of events processed by the multiple-target tracking algorithm, Table 2 shows some statistics. All the grouping events are well detected and processed. With the splitting events the system makes one mistake. This mistake occurs when a splitting event over two persons happens just before these persons leave the scene. Since at least half of each person is out of the scenario, the high-level tracker has less information about their appearance, and the matching process fails. A similar case occurs in the occlusion recovery events. Two persons that are involved in a group leave the scene. So, the algorithm dissolves the group and, later, the two persons enter in the scene again as a group. As the group is dissolved, the system cannot recover the identification.

Despite these situations, Table 2 shows promising results in multiple-tracking detection. Since there are few papers focused in collision detection, it is difficult to make a comparison against our method. In the case of the Rowe et al. algorithm, there are no tables showing the results obtained. Moreover, their algorithms were tested over less than 200 frames, which is a much more limited test about its performance. Despite this, their system has more difficulties to track isolated objects, obtaining an accuracy lower

than that presented by our method, according to the graphics. Other jobs that involve multiple tracking object do not show results about the collision detection, since they are mainly focused in tracking multiple objects without interaction.

We decided to make a comparison against other recent methodologies that address the same topic. We chose a common used dataset called CAVIAR. This dataset contains 26 different videos using two different cameras. Although every paper uses the corridor camera to test their algorithms, there is no frame without moving objects in the scene, so the background subtraction algorithm cannot be trained. Instead, the front camera is used. Table 3 shows the obtained results. Only moving objects higher than 25 pixels are considered. We also consider an object to be well tracked if it is within a group. Note that only our method, Huang et al. (2008) method and, lesser extent, Zhang et al. (2008) method can operate in real-time systems. Good results are obtained, enabling this algorithm to be used in real-time scenarios. Fig. 18 shows an example of this methodology under the CAVIAR dataset. Both grouping and splitting events are solved, and also long-time occlusion events.

This methodology was tested in a Pentium Quad Core running at 2.40 GHz with 4 RAM GB in a Linux Operative System. The videos used in this test had a 352×288 resolution. The results in Table 1 show that the system can process more than 50 images/s, guaranteeing that the system is able to operate in real-time environments.

6. Conclusions

We describe a hierarchical architecture for multiple-target tracking under uncontrolled scenarios. This representation is advantageous since it is able to cope with many of the problems this issue has, including occlusion recovery and collision event detection, such as splitting and grouping events. Two different trackers are used that are activated depending on the tracking object state. A low-level tracker based on a velocity prediction using Adalines is used to track every isolated object, while the high-level tracker, which stores every object appearance using a fixed pool of L^*a^*b-space color histograms, is used to manage the occlusion and collision events. This method was tested in over 30 video scenes and multiple examples show the performance of this technique. The system can detect every tracking object within the scene and recover its identification after collision event which involve in it. Moreover, the frames per second the system can analyze allow it to work under real-time scenarios.

This architecture is a robust and efficient framework for multiple target tracking, enabling the future possibility of analyzing higher-level behavior and interactions of these objects in a scene. Particularly, detection of abnormal trajectories or forming of groups can be addressed.

Acknowledgments

This paper has been partly funded by the Consellería de Industria. Xunta de Galicia through Grant Contracts 10/CSA918054PR and 10TIC009CT.

References

CANDELA, content analysis and networked delivery architectures. <http://www.multitel.be/~va/candela/>.

CAVIAR, context aware vision using image-based active recognition. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>.

Collins, R., Liu, Y., & Leordeanu, M. (2005). Online selection of discriminative tracking features. *PAMI, 27*(10), 1631–1643.

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(5), 564–577.

Cucchiara, R., Grana, C., Piccardi, M., Prati, A., & Sirotti, S. (2001). Improving shadow suppression in moving object detection with hsv color information. In *2001 IEEE proceedings of the intelligent transportation systems* (pp. 334–339).

Doshi, A., & Trivedi, M. (2006). "Hybrid cone-cylinder" codebook model for foreground detection with shadow and highlight suppression. In *IEEE international conference on video and signal based surveillance, AVSS '06* (p. 19).

Elgammal, A., Duraiswami, R., Harwood, D., & Davis, L. S. (2002). Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE, 90*(7), 1151–1163.

Han, B., Comaniciu, D., Zhu, Y., & Davis, L. S. (2008). Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(7), 1186–1197.

Haritaoglu, I., Harwood, D., & Davis, L. S. (1998). W4: Who? when? where? what? a real time system for detecting and tracking people. In *Proceedings of the third IEEE international conference on automatic face and gesture recognition* (pp. 222–227).

Horprasert, T., Hardwood, D., & Davis, L. S. (2000). A robust background substraction and shadow detection. In *4th ACCV, Taipei, Taiwan* (Vol. 1, pp. 34–41).

Huang, T., Koller, D., Malik, J., Ogasawara, G. H., Rao, B., Russell, S. J., et al. (1994). Automatic symbolic traffic scene analysis using belief networks. In *Proceedings of the 12th National Conference in Artificial Intelligence* (pp. 966–972).

Huang, C., Wu, B., & Nevatia, R. (2008). Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European conference on computer vision: Part II, ECCV '08* (pp. 788–801). Berlin, Heidelberg: Springer-Verlag.

Jepson, A. D., Fleet, D. J., & El-Maraghi, T. F. (2003). Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(10), 1296–1311.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering, 82*(Series D), 35–45.

Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-Time Imaging, 11*(3), 172–185 [Special issue on video object processing].

Li, M., Zhang, Z., Huang, K., & Tan, T. (2009). Rapid and robust human detection and tracking based on omega-shape features. In *16th IEEE international conference on image processing (ICIP)* (pp. 2545–2548).

Li, Y., Huang, C., & Nevatia, R. (2009). Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *IEEE conference on computer vision and pattern recognition, CVPR 2009* (pp. 2953–2960).

MacCormick, J., & Blake, A. (1999). A probabilistic exclusion principle for tracking multiple objects. In *The proceedings of the seventh IEEE international conference on computer vision* (Vol. 1, pp. 572–578).

Maddalena, L., & Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing, 17*(7), 1168–1177.

Nummiaro, K., Koller-Meierb, E., & Van Gool, L. (2003). An adaptive color-based particle filter. *Image and Vision Computing, 21*(1), 99–110.

Rohr, K. (1994). Toward model-based recognition of human movements in image sequences. *CVGIP, Image Understanding, 59*(1), 94–115.

Rowe, D., Reid, I., González, J., & Villanueva, J. J. (2006). Unconstrained multiple-people tracking. *Lecture Notes in Computer Science, 4174*, 505–514.

Song, B., Jeng, T.-Y., Staudt, E., & Roy-Chowdhury, A. (2010). A stochastic graph evolution framework for robust multi-target tracking. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision ECCV 2010. Lecture notes in computer science* (Vol. 6311, pp. 605–619). Berlin/ Heidelberg: Springer.

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Computer Society conference on computer vision and pattern recognition* (Vol. 2, pp. 246–252).

Stauffer, C., & Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 747–757.

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 780–785.

Wu, B., & Nevatia, R. (2006). Tracking of multiple, partially occluded humans based on static body part detection. In *2006 IEEE Computer Society conference on computer vision and pattern recognition* (Vol. 1, pp. 951–958).

Wu, B., & Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision, 75*(November), 247–266.

Xing, J., Ai, H., & Lao, S. (2009). Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *IEEE conference on computer vision and pattern recognition, CVPR 2009* (pp. 1200–1207).

Zhang, L., Li, Y., & Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *IEEE conference on computer vision and pattern recognition, CVPR 2008* (pp. 1–8).

2.2 Journal Paper: Multiple Human Tracking System for Unpredictable Trajectories

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Marcos Ortega Hortas
Affiliation: Universidade da Coruña, Spain

Co-author #3: Manuel Francisco González Penedo
Affiliation: Universidade da Coruña, Spain

Article title: Multiple Human Tracking System for Unpredictable Trajectories
Journal: Machine Vision and Applications
Volume: 25(2)
Pages: 511–527
Editorial: Springer
ISSN: 0932-8092
Year: 2014

Machine Vision and Applications (2014) 25:511–527
DOI 10.1007/s00138-013-0544-7

ORIGINAL PAPER

Multiple human tracking system for unpredictable trajectories

B. Cancela · M. Ortega · M. G. Penedo

Received: 22 November 2012 / Revised: 3 May 2013 / Accepted: 21 August 2013 / Published online: 5 September 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract Tracking multiple objects into a scene is one of the most active research topics in computer vision. The art of identifying each target within the scene along a video sequence has multiple issues to be solved, being collision and occlusion events among the most challenging ones. Because of this, when dealing with human detection, it is often very difficult to obtain a full body image, which introduces complexity in the process. The task becomes even more difficult when dealing with unpredictable trajectories, like in sport environments. Thus, head-shoulder omega shape becomes a powerful tool to perform the human detection. Most of the contributions to this field involve a detection technique followed by a tracking system based on the omega-shape features. Based on these works, we present a novel methodology for providing a full tracking system. Different techniques are combined to both detect, track and recover target identifications under unpredictable trajectories, such as sport events. Experimental results into challenging sport scenes show the performance and accuracy of this technique. Also, the system speed opens the door for obtaining a real-time system using GPU programing in standard desktop machines, being able to be used in higher-level human behavioral systems, with multiple applications.

Keywords Background subtraction · Cascade classifier · Histogram of oriented gradients · Particle filter · Collision detection · Occlusion recovery

B. Cancela ( ) · M. Ortega · M. G. Penedo
VARPA Group, University of A Coruña, Campus de Elviña, s/n, A Coruña, Spain
e-mail: brais.cancela@udc.es

M. Ortega
e-mail: mortega@udc.es

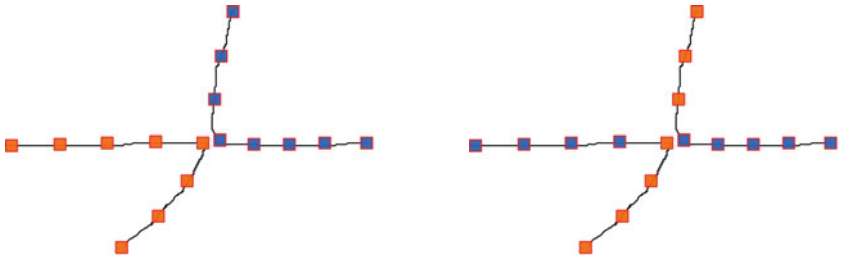
M. G. Penedo
e-mail: mgpenedo@udc.es

1 Introduction

The aim of automatic detection and tracking of human beings is a challenging issue, specially when dealing with surveillance systems. These procedures have to isolate every person in the scene to study his behavior. There are multiple solutions to this topic, from generic algorithms, which can track every moving object in the scene, to architectures specially developed to deal with persons.

All these systems, when dealing with unpredictable trajectories, have the same bottleneck: target tracking. A lot of challenges arise in the aim of tracking every single object around the scene. Two are the most important issues: occlusions and collisions. An occlusion occurs when a person escapes from the camera control. This can occur if a target goes outside the camera range, or when is totally or partially hidden by any element of the scenario. The algorithm must have the ability to detect the occlusion and, if the target reenters again in the scene, to recover the previous identification. On the other hand, a collision occurs when two or more people cross in front of the camera, occluding part of the targets. The issue is similar with the occlusion, but it is more tricky. First, the system must detect which targets are occluded and which are not. So, when this problem is solved, the rest can be processed as occluded targets. In the collision case, if the system does not perform well, then it could derive into a switch identification problem. This issue must be avoided, since it is very difficult to recover the correct solution when it happens.

There are many different approaches in the literature for target tracking. We can find two big groups: low-level and high-level methods. Low-level techniques, like optical-flow [21] or the Kalman filter [14] are simple and fast but they are too soft to work in complicated scenarios, since they have no possibility to recover a target when an occlusion event occurs. We focus on tracking of human beings in unstructured



(a) Real trajectories **(b)** Associated based tracking output

Fig. 1 Associated-based tracking issue. When dealing with sudden orientation changes, associated-based tracking swap the identifications. **a** Real trajectories, **b** associated-based tracking output

scenes, so we need to use high-level approaches to deal with the problems mentioned before.

The main idea into the high-level systems is to add information a priori about the objects of interest. In particular, when dealing with people, information about the human shape is used. In first attempts, Haritaoglu et al. [12] used different cardboard models to represent the human body and located it using dynamic template matching. More recently, Dalai et al. [9] introduced the histograms of oriented gradients (HOGs), which are used to train a SVM for each part of the body. This technique was also used by Felzenszwalb et al. [11] to train any object in the scene. However, these methods try to detect every part in the body, many of which are occluded in crowded scenes.

To solve this, Li et al. [16,17] simplified the method, trying to locate only the omega shape created by the head and the shoulders. A HOG feature-based SVM is used to confirm every target previously located using a Viola–Jones type classifier [25], which improves the speed of the algorithm. A particle filter (PF) is used to track every detecting object in the scene. Based on [10] and using the omega-shape detection, Rodriguez et al. [20] improved the detection in crowded scenes including a density estimation parameter. However, the density information computation forces the algorithm to compute the HOG descriptor in every pixel image, making the system slower. Other different techniques focused in crowd analysis can be found in [28].

More recently, associated-based tracking systems were used. In [23], Starder et al. introduced geometric and long-term temporal constraints to increase the accuracy algorithm. Also they use trajectory filters to increase target identifications. Benfold et al. [4] combined a body and a head HOG detectors with simultaneous KLT tracking and Markov-Chain Monte-Carlo Data Association to estimate the most probable trajectories. Andriyenko et al. [1,2] used a similar scheme. They infer the most usual path for every target, performing the matching by minimizing the energy related to each trajectory. Information about the interaction between subjects is included in [15] to increase the identification accuracy. A condition random field was also used by [26,27] to produce discriminate descriptors, which are used to a better tracking dealing with partial occlusions and collisions.

The problem with associated-based tracking techniques arises when targets operate with no usual behavior, for instance, a sport event, like football or basketball. When an offense player makes a crossover, or a fake movement, and the defender falls into the trap, an associated-based tracking would swap the identifications, as Fig. 1 shows. Associated-based tracking techniques assume targets motions are stable, i.e., linear and constant speed in a short period, causing incapable of dealing with big and unpredictable movements like that in sports. It is very difficult to anticipate that kind of movements. So, a different technique must be used.

To perform the tracking, techniques like particle filters [3] (PFs) or the Lucas–Kanade algorithm [22] are used, which are good methods to track isolated people. However, these algorithms tend to accumulate an error along successive detections, which often results in the loss of the target detection. Moreover, there are problems with occlusions and collisions in multiple-tracking scenarios, which rarely are solved using these methods. For instance, a target lost while walking towards the camera cannot be recovered using these techniques if it reappears walking in other direction. Hence, an ensemble of multiple techniques are required to solve these complicated situations.

In this work, we present a strategy for human tracking under unpredictable trajectories, able to solve many of the problems mentioned before. Based on a previous work [6], this strategy is based on the omega-shaped descriptor. Some improvements to the Viola–Jones detection are made to increase the speed. A particle filter system, in combination with a linear filter to predict the next position, is used to perform the tracking. The detection procedure using The Viola–Jones and the HOG feature-based SVM are also used in the tracking system to reduce the error produced by the particle filter along the successive frames.

A hierarchical architecture is created to deal with the problems associated with multiple-target tracking scenarios. Since the particle filter is used to track every person in the scene, it is disabled when either a collision or an occlusion event is detected. An ellipse representation is used to define the areas in which the lost target could be. Once a new target appears in one of these areas, it is compared against the lost target using a color-histogram representation. The use

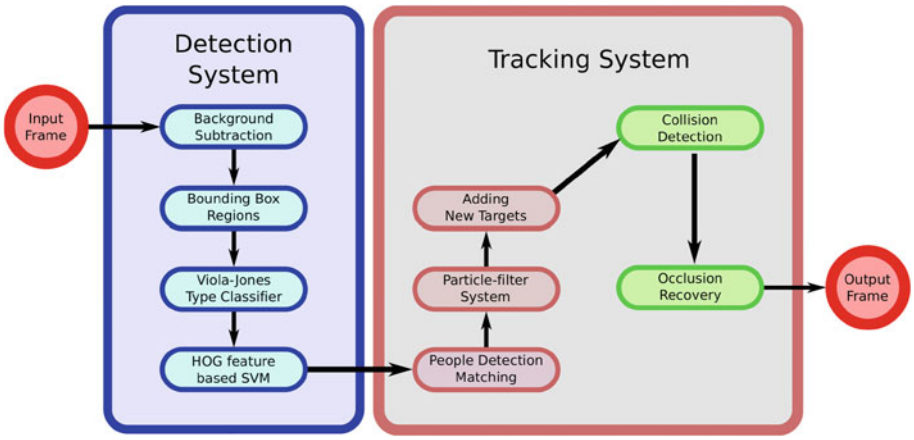


Fig. 2 Multiple-target tracking framework. The system is divided in two different modules: the detection system, which searches for every head included within the scene, and the tracking system, which follows each detection along frames

of this technique is mainly focused in camera locations far from the scene to analyze. The algorithm speed is highly increased when, regardless of the position in which the target is, there are only few differences between the size of his head.

As Fig. 2 shows, two different modules are included in our methodology. First, a detection system tries to seek and obtain every person in the scene. Later, the tracking system matches every detection with the targets previously created, adding new observations if necessary.

This paper is organized as follows: Sect. 2 describes the method used to detect every human being in the scene; Sect. 3 describes the techniques used to perform the tracking, in combination with the collision and occlusion solvers; Sect. 4 introduces some tests and evaluations of the framework, showing the obtained results; finally, Sect. 5 offers conclusions and future work.

2 Human detection

In the first step of the algorithm, the system is going to detect every person in the scene. Our initial approach is similar to the original idea exposed in [17]: a Viola–Jones detector is combined with a HOG featured-based SVM to obtain a good agreement between accuracy and speed. Since the HOG descriptor process is quite expensive in terms of speed, the idea is to relax this step by restricting the positions in which we have to compute it. So, a Haar-like method, the Viola–Jones type classifier, is used to detect head-shoulders omega-shape feature. This method can work very fast, but the classification performance is poor. So, for every detection, the HOG descriptor is computed and evaluated in the SVM.

In this initial approach, human detection system is reduced only to the position in the image for which they know a person could enter into the camera range, to increase the system speed. As mentioned before, we are going to use this system both in detection and tracking procedures, so we create a different approach. In Fig. 2, we show the steps to perform people detection. We consider that a target should be tracked if it moves along the scene in some moment. Static targets are not considered until they move. Once this happens, the algorithm begins to track the target, even if it stops again. First, a background subtraction technique is used to detect the moving objects. Later, a Viola–Jones type classifier is applied in the regions where we detect movement. Finally, the HOG feature-based histogram is responsible for evaluating the Viola–Jones positive detections.

2.1 Background subtraction

A quick method for detecting motion was selected. Although optical flow is the commonly used technique in this case, we are only interested in the positions where a movement occurs, without taking into account neither orientation nor magnitude of the movement. Hence, knowing optical flow has a higher computational cost, we propose to use a background subtraction technique. Our idea is to use an algorithm able to be updated every frame. So, we choose the Mixture of Gaussians (MoG) algorithm described in [29]. We introduce a short window history (over 100 frames), to quickly consider as background-stopped targets. Figure 3 shows an example of the results of this technique.

2.2 Bounding box selection

Once we have detected all the moving pixels in the scene, we select the regions in which the Viola–Jones type classifier is going to be used. So, we split the foreground into different boxes. First, we remove the noise into the foreground regions. To do that, both open and close morphological operators are used, along with a minimum-area filter. Later, we perform a blob detection technique, with a simple four-connectivity algorithm. Finally, a bounding box is obtained for each blob.

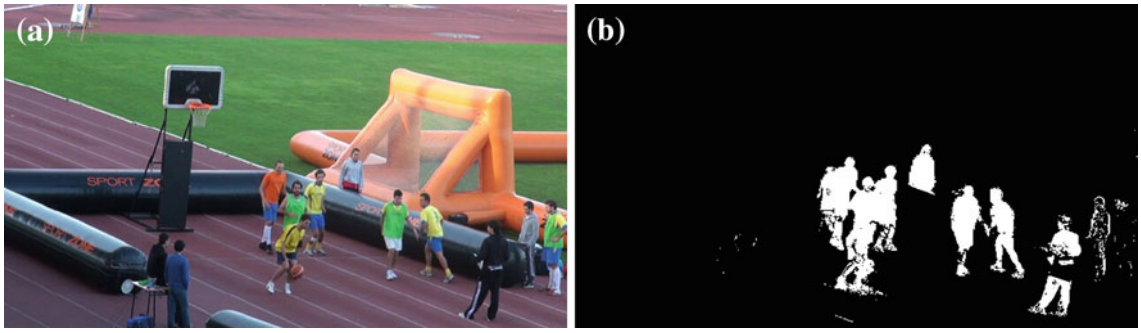


Fig. 3 Background subtraction using MoG algorithm

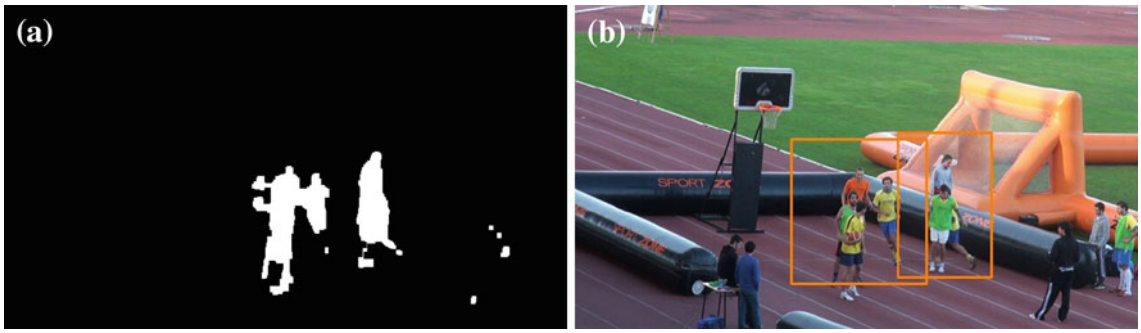


Fig. 4 Bounding box selection using foreground pixels. **a** Foreground pixels after preprocessing techniques. **b** Bounding boxes obtained

Each bounding box marks the regions in which the Viola–Jones type classifier is used.

Due to the nature of background subtraction techniques, it is difficult to locate all moving pixels in the image, specially those which are in the person contour. Having this situation in mind, the size of each bounding box is increased to cope with this issue. The margin is related with the expected head size at the contour of the box. Furthermore, no margin is considered at the bottom, because of the nature of human movement. No positive head-shoulder shapes can be observed at the bottom of a bounding box, unless it matches with the bottom of the scene. Furthermore, if the overlap between two or more bounding boxes is high, we merge them to avoid recalculations. Figure 4 shows an example of this technique.

2.3 Viola–Jones type classifier

After the bounding boxes are located, the Viola–Jones type classifier is executed. In many implementations, the process consists in the classification of Haar features within the image. Multiple window sizes are used along the image. The patches which do not pass all the cascade classifiers are excluded as obvious non-head-shoulder image patches.

We can improve the performance of this technique knowing the estimated head size in every pixel in the scene. According to [13], the object height in an image h_i follows the equation

$$h_i = \frac{y_i}{y_c}(v_i - v_0), \tag{1}$$

being y_i the 3D object size, y_c the camera height, v_i the position in the image we are considering and v_0 the horizon point. Knowing the average person head size (22.6 cm [18]), we decide to set the 3D head-shoulder size to 30 cm. The other parameters, v_0 and y_c can be estimated using ground truth positions and heights of two detections in the image [20]. Using this idea, we can improve the system in two different ways: increasing the speed and reducing the false positive errors, avoiding patches with wrong sizes.

As mentioned before, we also introduce knowledge about the human movement. We assume that, when a head movement occurs, it is also followed by the torso. Having this idea in mind, we choose a small rectangle at the bottom of each patch detected by the Viola–Jones cascade classifier. Its height is related with the height of the patch. If the number of pixels marked as foreground in that rectangle is low (over 10 times the head size), we discard that patch. Figure 5b shows an example. Some results can be discarded without taking into account the HOG feature. Also good patches are deleted because of the absence of movement in the target, causing his torso to be marked as background. An example of this issue can be viewed at the bottom right of the figure. This is a problem not taken into account, because it is assumed we can find it and track it in previous frames, when in movement.

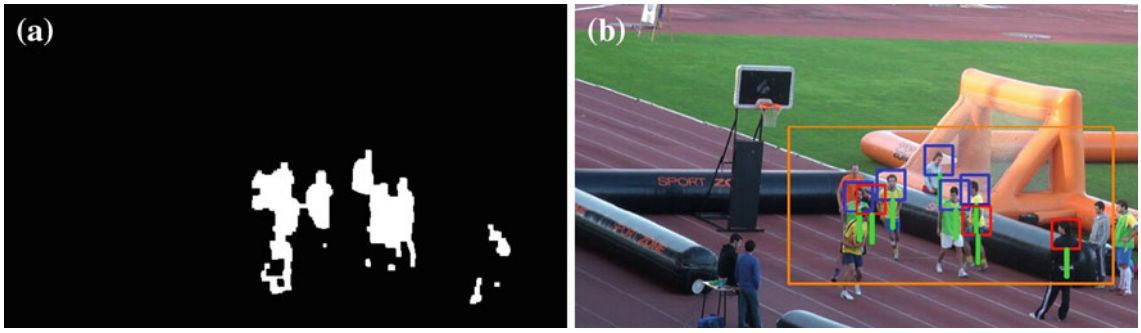


Fig. 5 Viola–Jones cascade classifier example. **a** Foreground pixels after preprocessing techniques. **b** Viola–Jones classification. In *green*, the rectangle used to check the torso. In *blue*, patches returned by the Viola–Jones which passed the torso check. In *red*, the discarded patches (color figure online)

2.4 HOG feature-based histogram

Once we have obtained the Viola–Jones positive patches, a HOG feature-based histogram is used into a SVM to confirm the detection. We perform a classic HOG technique. First, we divide the patch into cells. Adjacent cells form a block, and a normalized histogram within the block for each cell included in it. Each histogram includes eight different orientation bins from 0 to 360 °. HOG feature extraction details can be seen in [9].

As we can see, the HOG feature-based histogram creation has a lot of redundant operations. Hence, Porikli et al. [19] introduced the integral histogram, which highly increases the speed of the algorithm. Combining both Viola–Jones cascade classifier and HOG feature-based histogram, we can obtain a good balance between speed and accuracy. Although we considered the idea of computing the integral histogram only in the bounding boxes explained in Sect. 2.2, we decided to discard it because we also need the histogram to perform the tracking, as it will be explained later. The integral histogram is computed along the whole scene. Examples of accepted and rejected patches can be seen in Fig. 6.

The combination of the HOG descriptor along with the Viola–Jones type classifier often results in many patches related with the same target. Thus, when we add new targets to the tracking system, we have to penalize overlapping detections.

3 Tracking system

Once we have detected persons in the scene, we do not instantiate new targets directly. First, we perform the tracking system to avoid the persons detected which are already tracked. As previously depicted in Fig. 2, three different steps are performed into the tracking system: first, a distance matching is used between the targets previously tracked and the new positions detected by the Viola–Jones in combination with the HOG feature-based histogram; second, a particle filter is launched for every target which has not been tracked using the previous step; finally, the remaining individuals in the new frame are considered to become new targets. Furthermore, two more steps are included to detect both collision and occlusion events, providing a target recovery identification system.

First, we develop an algorithm to predict the position of each target along the following frames. Although the Kalman Filter is a common technique used in this context, we propose the use of a linear filter, since it is a more efficient technique and shows a good performance under noisy images. A bunch of adalines is used to predict the velocity of each target position component. Examples of the performance of this technique can be seen in [5].

Although every Viola–Jones patch could have both different position and size, we will only take into account the parameters related with their position, since, as seen in Sect. 2.3,

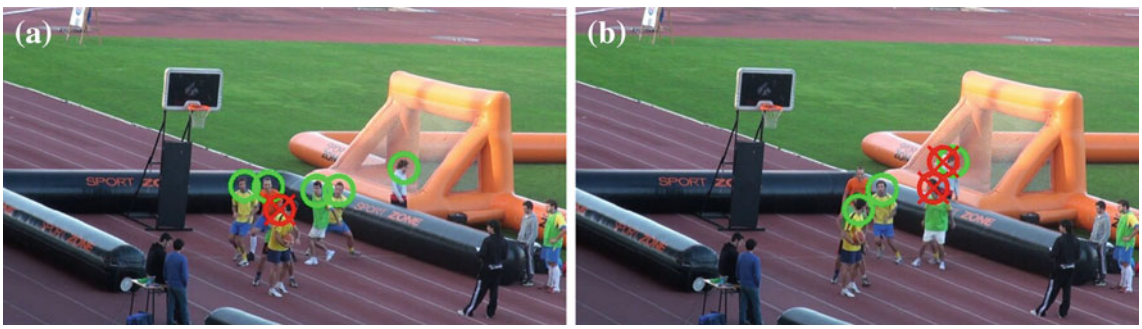


Fig. 6 HOG feature-based histogram example. In *green*, the patches accepted by the SVM. In *red*, the denied patches (color figure online)

Fig. 7 Particle filter error. Along the successive frames, purple target is losing the quality of the detection

Fig. 8 Tracking using person detection system. In *circles*, new patches detected by the Viola–Jones type classifier. In *rectangles*, predicted position for each target previously tracked

once we have the target predicted position, we can determine its size using the Eq. 1. Thus, we can reduce the computational cost.

3.1 Using human detection technique

To track each target into the scene, we propose the use of a particle system. However, the particle system can accumulate an error along the frames, which can often be very difficult to recover from it. In Fig. 7d, we can see how purple target its losing the head-shoulder position, resulting in a bad performance. Hence, it is interesting to add another solution that can correct, as far as possible, that accumulated error. Our solution proposes the inclusion of the people detection technique within the tracking system.

With every patch detected, we create a group with all the possible targets that could fit in the new detection. More formally, if we have a patch in the new frame defined by $\mathbf{p}_i^{t+1} = (x_i^{t+1}, y_i^{t+1})$, where x_i^{t+1} and y_i^{t+1} are the coordinates of the patch center, we compute the target factor as

$$T_{\mathbf{p}_i^{t+1}} = \left\{ \tilde{z}_j^{t+1} \in \Omega^t \,\middle|\, d\left(\mathbf{p}_i^{t+1}, \tilde{z}_j^{t+1}\right) \leq \tau \right\}, \tag{2}$$

where Ω^t is a set containing all the tracking objects detected (and not occluded) at time t, $\tilde{z}_j^{t+1}$ is the predicted position of the centroid of the target z_j and $d(\mathbf{p}_i^{t+1}, \tilde{z}_j^{t+1})$ is the euclidean distance between the center of the two patches. Setting an small value τ we can obtain good results. Using this equation, if we obtain that $|T_{\mathbf{p}_i^{t+1}}| = 1$, we assign the new patch to the target contained into the set. On the other hand, if $|T_{\mathbf{p}_i^{t+1}}| > 1$ a collision occurs. Handling of these events is discussed in Sect. 3.4 in more detail. Finally, if no targets are contained into the set, $\mathbf{p}_i^{t+1}$ is set as candidate to be a new target. Targets that are not associated with any patch detected by the Viola–Jones type classifier are tracked using the next step.

In Fig. 8, we can see an example of how this technique is useful. However, we have to be very careful with the τ parameter. In Fig. 8b, we have a new patch related to the person with the orange shirt. As it has not been tracked yet, using a high value of τ could change the identification of the person behind him, which has been already tracked, leading to a wrong situation. For this reason, a lower τ value must be used ($\tau = 5$ in our experiments).

3.2 Particle filter system

As mentioned before, also using the person detection system for tracking guarantees a good quality in the head-shoulder detection. However, our detection system idea is focused in

locating the persons in the scene avoiding, as much as possible, false positive patches. As a result, the false negative rate is also high, causing the algorithm not to find everyone in the scene at all frames. To avoid the loss of targets, we propose a particle-based system. As object representation, given a patch $\mathbf{p}_i^{t+1}$, we use the extracted local HOG features, using it to model its appearance ($\mathbf{O}_i^{t+1}$). For distance measurement we use the Bhattacharyya coefficient, which is proved as a good method for tracking non-rigid objects [8]. This coefficient is used to determine the similarity between two different observations. So, given a new observation $\mathbf{O}^{t+1}$ and an object representation $\hat{\mathbf{O}}^t$, corresponding to the target z at time t, the similarity between the two vectors is

$$S(\mathbf{O}^{t+1}, \hat{\mathbf{O}}^t) = \sum_{u=1}^{m} \sqrt{O_m^{t+1} \hat{O}_m^t}, \tag{3}$$

being m the vector dimension. Higher values indicate better similarities between the observations.

As in the tracking using the people detection technique, we will use the predicted position of each target to perform the tracking. As mentioned before, we use a linear filter to predict the position of the target in the new frame. However, due to unpredictable target movements, the new target position often differs a few pixels from the prediction. So, having a target z_j previously tracked, we can assume the new target position to be described as

$$z_j^{t+1} = \tilde{z}_j^{t+1} + \omega, \tag{4}$$

being $\tilde{z}_j^{t+1}$ the predicted position of the target z_j at time $t+1$ and $\omega \sim N(0, \Sigma)$ is a Gaussian noise. In our approach, we add the Gaussian noise to the predicted position, generating a bunch of different particles. For each particle, we extract its HOG feature in the patch defined by the new position and the size computed using Eq. 1, and compute the Bhattacharyya coefficient between that particle and the target model. We choose as the new position z_j^{t+1} as the particle which obtain the highest coefficient.

Finally, we have to change the target model. To adapt the model to the changes along the frame, the object must be updated. Also, because of the possibility of a bad chosen particle, the model should maintain information about previous features. Hence, having a target model $\hat{\mathbf{O}}^t$ and a new model $\mathbf{O}^{t+1}$, corresponding with the HOG feature of the winning particle, the target model is updated following

$$\hat{\mathbf{O}}^{t+1} = \alpha \hat{\mathbf{O}}^t + (1-\alpha)\mathbf{O}^{t+1}, \tag{5}$$

where α is the learning parameter. In Fig. 9, we can see an example about the particle dispersion.

Four different states are defined: *training*, *tracked*, *occluded* and *removed*, as Fig. 10 shows. First, when a new target is instantiated, its state is initialized as *training*. A target in this state means that a person is detected in the scene, but there is not enough evidences to confirm the detection yet. While a target remains in this state, each new position has to be confirmed using the SVM, even if the tracking is done by the particle filter system. If the SVM cannot confirm the detection, the target changes its state to *removed* and it is erased.

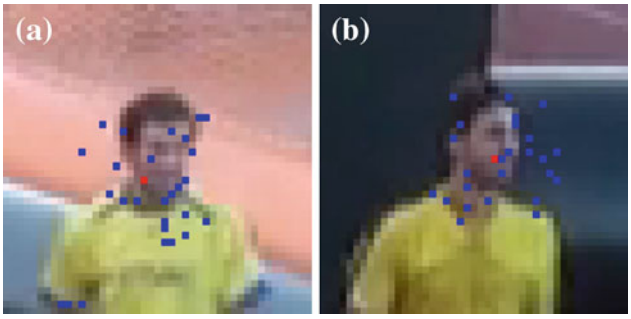


Fig. 9 Particle filter example. In *red*, target predicted position. In *blue*, particles thrown (color figure online)

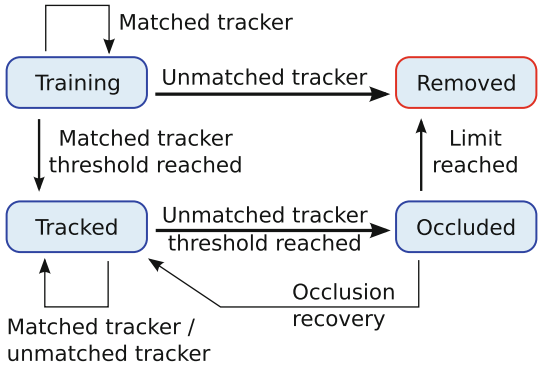


Fig. 10 State diagram of a tracking object

If the number of times the target is confirmed by the SVM reaches a threshold (τ_T), it changes its state to *tracked*. In this state, no SVM evaluation is needed using the particle filter system. So, a different way is needed to detect when the target is lost. Three different possibilities could cause the system to loss a target: particle filter system bad accuracy, collisions and occlusions. Both collisions and occlusions will be explained later.

To deal with the particle filter bad accuracy, we decided to set two different thresholds, γ_S and τ_S, and an occlusion counter ρ_S. The Bhattacharyya coefficient explained in Eq. 3 is used to measure the tracking quality. Every time the coefficient becomes higher than γ_S, or when the target is detected using the person detection tracking, we set $\rho_S = 0$. On the contrary, if $S(\mathbf{O}^{t+1}, \hat{\mathbf{O}}^t) < \gamma_S$, we increase the ρ_S counter. If, after successive results below γ_S, we have that $\rho_S > \tau_S$, we change its state to *occluded*.

This indicates that we cannot be sure about the position of the target. However, despite the fact that the tracker have lost the person, we continue to update the target position using the particle filter technique. Although the target position is not known, the tracker position typically remains close to the person being tracked. So, if the human detection system detects a person near the tracker position, and there are no other targets nearby, the target recovers its state from *occluded* to *tracked*.

Although in early stages we need to confirm the detection using the SVM to avoid possible false positive detection, a technique is needed to detect and remove it when appears. False positive detections are mostly related with background detections. Because of that, a false detection is a steady target which does not move along frames. Thus, the Bhattacharyya coefficient obtains really high values. So, another coefficient is included, τ_H. If the Bhattacharyya coefficient exceeds this coefficient, the target is considered as a false positive, and it is removed. Although this threshold can remove correct detections, like steady targets, in the experiments we found that small movements in the targets causes the particles to never reach τ_H.

3.3 Adding new targets

Once we have successfully tracked all the targets we had detected in the previous frames, we initiate the process of adding new targets to the tracker pool. An energy equation must be defined. Assuming we have a confidence score $s(z)$ for each new location into the scene, our goal is to identify new targets, excluding those which are already tracked. We encode every target in the tracker pool, in combination with all the new detections into a single N-vector $\mathbf{x} \in \{0,1\}^N$, being N the vector size. Our energy minimization function is defined as

$$\min_{\mathbf{x}\in\{0,1\}^N} \underbrace{-\mathbf{s}^T\mathbf{x}}_{E_s} + \underbrace{\mathbf{x}^t\mathbf{W}\mathbf{x}}_{E_p}, \tag{6}$$

where $x_i = 1$ confirms the detection of the tracker z_i into the scene, and $x_i = 0$ if it is a false detection. We choose as confidence score $s(z)$ the distance between the HOG feature, related to the target z, with respect to the SVM hyperplane, and $\mathbf{W}$ is the overlapping matrix.

Minimizing the first term of the Eq. 6, E_s, means that the targets with high confidence detection rate (far from the SVM hyperplane) are considered as detected persons. Moreover, the second term, E_p, ensures valid configurations when non-overlapping detections are selected. This is done by setting $W_{ij} = 0$ when there are no significant overlap between the targets z_i and z_j (in our experiments, $<25\,\%$ overlap between the bounding boxes detection area), and $W_{ij} = \infty$ otherwise. The reason for penalizing overlapping detections is related to the SVM. Confirmed detections with this amount of overlapping are most often associated to the same target. Using this threshold we ensure only one detection per target.

3.3.1 Initialization and optimization

Since the optimization of Eq. 6 is, in general, a NP-hard problem [20], we follow a greedy search procedure, similar to [10]. However, our initialization differs from that. Instead of initializing $\mathbf{x} = 0$, that is, not having any confirmed object in the scene, we put every k target detected with the tracking system explained before with value $x_k = 1$, value which will be blocked. Each possible new target is initialized as $x_k = 0$. Furthermore, since the targets previously tracked are blocked, we can put their confidence score $s(p) = 0$, and the values $W_{ij} = 0$, being z_i and z_j two blocked trackers. This is done to avoid the interference of the blocked targets in the value of the equation. They are only used to avoid overlapping with the new possible targets.

Iteratively, we update $\mathbf{x}$ by turning from 0 to 1 the target z_i which decreases the value of the Eq. 6 by the largest amount. The iterations will stop when the cost of the function cannot be decreased anymore.

3.4 Collision detection

A collision event occurs when two or more targets cross into the scene, causing the total or partial occlusion to many of the people. Dealing with a head-shoulder technique to track people, two different collision possibilities may occur: a head-to-head collision or a head-to-body collision. In the first case, the situation is easy to detect. We only need to see the overlap between the last position detected by the two targets. Having two different targets locations z_i^t and z_j^t, if they have a significant overlap ratio, a collision occurs. In Fig. 11a, we can see an example of a head-to-head collision.

Only one target remains as *tracked* after a head-to-head collision. The decision of which target is in the forefront is also based on the Bhattacharyya coefficient. The target involved in the collision which have the higher $S(\mathbf{O}^{t+1}, \hat{\mathbf{O}}^t)$ is the person considered into the front. The remaining targets will change their states from *tracked* to *occluded*.

On the other hand, a head-to-body collision occurs when a head is occluded by a body of another person (Fig. 11b). To solve this situation, we make an assumption: the average body height is two times the head-shoulder patch height. So, we create a rectangle at the bottom of each target detection simulating the body. If the overlap ratio between the body rectangle with any target position is significant, a collision may occur. It is possible not to be a collision if the position of that head if between the camera and the body of the other target (Fig. 11c). To determine whether a collision event occurs or not, we use the SVM to confirm the detection of the target which is possibly occluded by the body. If the detection is not

Fig. 11 Collision example. Two possibilities may occur: head-to-head collision (**a**) or head-to-body collision (**b**, **c**)

confirmed, we increase the occlusion counter ρ_S explained before.

3.5 Occlusion recovery

Previously, we have mentioned one method to recover a target when it becomes *occluded*, consisting in following the particle filter tracking until the person detection is able to re-detect the target. This method often works if the occlusion is produced by a bad particle filter estimation. However, when dealing with collisions, this system is useless, emphasizing the need for another occlusion recovery system. To solve this problem, a new tracker based on feature selection is used.

RGB channels have been previously proposed for identification [7]. In this work, instead of using raw R, B and G channels, we propose to use another space-color system, isolating illumination from color information, to avoid external effects such as lighting conditions. So, a fixed pool of histograms is defined as follows:

$$h = \omega_1 a + \omega_2 b, \quad w_{1,2} \in \{-1.0, 1\}, \tag{7}$$

where a and b are the color-opponent dimensions in L^*a^*b-color space. Removing all possible linear combinations, we reduce it to four different histograms. In addition, L component is included in another histogram. This component is only used when the other histograms are not enough to distinguish between two different targets. All these histograms are inserted into one unique vector.

Once we have defined the feature selection, we need a technique to decide, having any given target, which pixels belongs to it in the image. As mentioned before, the methodology is required to work under partial or total occlusions. Usually, the camera has no total vision on the whole target. Two different techniques are used to obtain the target features: background subtraction and body estimation. We take advantage about the fact that background subtraction is loaded when the people detection technique is used, as seen in Sect. 2.1, so no more computation is needed.

The body estimation was also mentioned before, when dealing with head-to-body collision detection techniques. A rectangle in the bottom of the head-shoulder detection is used to perform it. Hence, as we can see in Fig. 12, combining these two ideas we obtain the pixels that belongs to each target. However, in some cases, there are foreground pixels shared by two or more targets (Fig. 12c). That pixels are discarded for every target involved.

However, this method also presents some difficulties. The most important is related with the background subtraction technique. It is very difficult to capture thin elements, such as arms and legs. The presence or absence of these body parts could highly vary the target feature based on histograms. Also, this histogram achieves similar values with

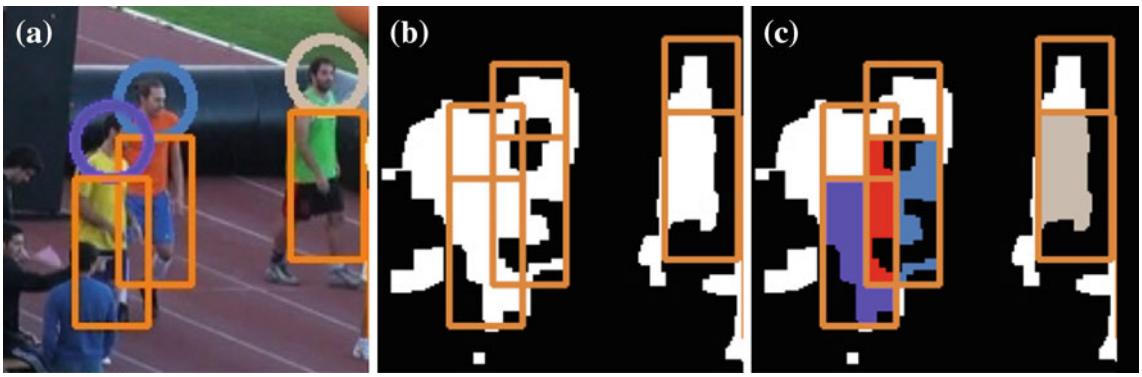


Fig. 12 Feature selection technique example. Using the background subtraction and the body estimation, we detect the pixels that belongs to each target (**c**). Pixels shared by two or more estimations are ignored (red pixels in **c**) (color figure online)

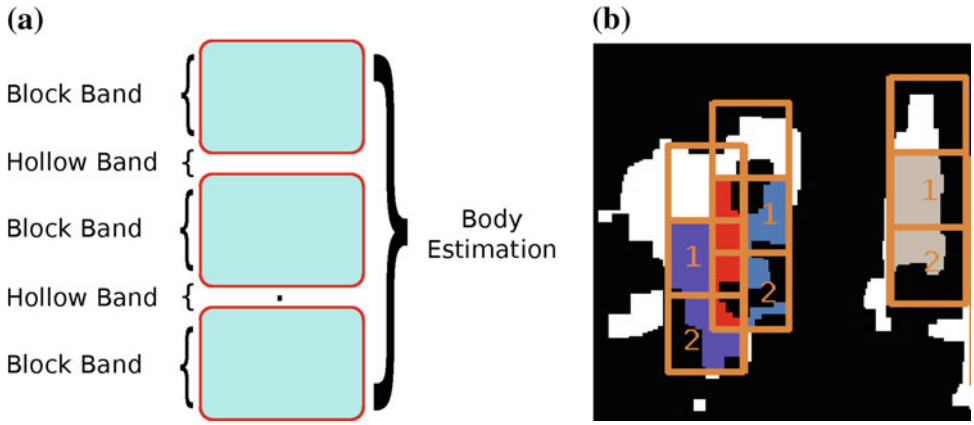


Fig. 13 Feature split diagram. Each body estimation is divided into different block regions (**a**), each one with an associated pool of histograms. For instance, we can divide each target body into two different regions (**b**)

similar clothes combinations. For instance, blue shirt and white trouser target has a similar histogram to a white shirt and blue trouser person. So, as we can see in Fig. 13a, we decide to split the body estimation into different regions separated by hollow bands which are not taken into account when we compute the histograms. Each block has its own pool of histograms. Figure 13b shows an example of a two region split.

Because of the background subtraction issue explained before, it is possible that some block regions has not enough foreground pixels to perform a relevant histogram. Thus, we introduce another threshold, τ_F, which is compared against the ratio of foreground pixels within the block. If the ratio is low, the pool of histograms is not created for that block.

Each histogram is normalized and discretized into N bins. Having the histogram $\mathbf{p}^i = \{p^i_k; k = 1 : N\}$, corresponding to the ith feature of the tracking object, the probability of each feature is calculated as

$$p^i_k = C^i \sum_{a=1}^{M} \delta(b(x_a) - k), \tag{8}$$

where C^i is a normalization factor which ensures that $\sum_{k=1}^{N} p^i_k = 1$, δ is the Kronecker delta, M are the number of pixels belonging to the target, x_a is the pixel position and $b(x_a)$ is the function that associates each pixel value to its corresponding bin.

3.5.1 Occluded target search

Once a target is lost, we do not assume that the target can appear in any position in the scene. We consider two different restrictions to help the recovery system to increase the accuracy. First, we compute the average speed of the target. So, the target position along successive frames is limited by a circumference with the latest known position as center and the maximum distance traveled as the radius. Hence, we can limit the target available positions.

However, this idea is not enough as it does not take into account neither movements nor target groupings. So, a second restriction is used to partially locate the target position. We start from the bounding box idea explained in Sect. 2.2. We perform a blob detection using the background subtraction. Thus, we obtain the position of both isolated and grouping targets. Instead of using a box, we represent each blob as an ellipse. Every j-observed blob at the time t is represented as

$$e^t_j = (x^t_j, y^t_j, h^t_j, w^t_j, \theta^t_j), \tag{9}$$

where (x^t_j, y^t_j) is the position of the ellipse centroid, h^t_j and w^t_j are the size of the maximum and minimum axes, respectively, and θ^t_j is the orientation.

Each time we obtain a target matching in the scene, we store the information about the blob in which it is included. Consequently, when the target is lost, we have the informa-

Fig. 14 Occluded target search example. When the blue target is lost because of bad accuracy (**b**), the algorithm detects the blob in which the target could reappear (**c**) (color figure online)

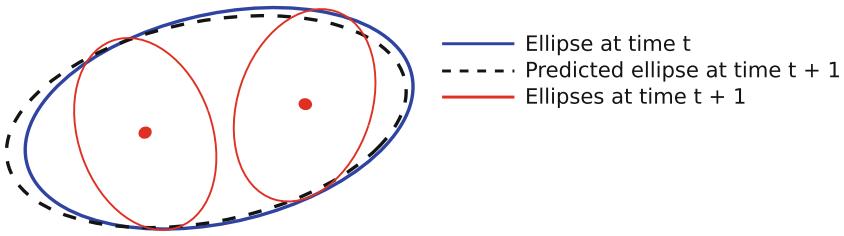


Fig. 15 A split in a new frame occurs when two or more ellipse centroid fit within the previous ellipse of any tracking object

tion about a limiting contour where the target can be located. Figure 14 contains an example of how this method works. However, blobs do not remain invariant along successive frames. Both grouping and splitting events could occur, and have to be considered.

A splitting event occurs when a blob is divided into two or more (Fig. 15). We can use the ellipse properties to detect this issue. More formally, if we have a lost target limited by the ellipse $e_j^t = (x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t)$, we compute the splitting set as

$$\mathrm{Sp}_{e_j^t}\left\{e^{t+1} \in \Omega'^{t+1} \;\middle|\; \frac{x'^2}{(h_j^t)^2} + \frac{y'^2}{(w_j^t)^2} \leq 1\right\}, \tag{10}$$

where Ω'^{t+1} is a set containing all the blobs in the scene at time $t+1$, and x' and y' are the positions of the e^{t+1} ellipse centroid under e_j^t coordinates

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} \cos\theta_j^t & -\sin\theta_j^t & 0 \\ \sin\theta_j^t & \cos\theta_j^t & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & -x_j^t \\ 0 & 1 & -y_j^t \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x^{t+1} \\ y^{t+1} \\ 1 \end{vmatrix}. \tag{11}$$

The $\mathrm{Sp}_{e_j^t}$ set is cropped erasing all the blobs which have no area within the maximum advance circumference. The remaining blobs are used to search new targets that could be the target lost before.

3.5.2 Tracking object matching

The average appearance histogram $\mathbf{m}_j^t$ of each target is updated with every confirmed observation. A recursively process is used to compute it:

$$\mathbf{m}_j^t = \frac{n_j \mathbf{m}_j^{t-1} + \mathbf{p}_j^t}{n_j + 1}, \tag{12}$$

where n_j is the number of the target confirmed observations. As mentioned before, once a new target is instantiated and trained, its state is changed to *tracked*. When this occurs, we evaluate the feature selection. If we have enough values in the histogram pool to perform a comparison, we compare the new target against all the occluded targets that could appear in the blob that has the new target within. A modified Bhattacharyya coefficient is used, taking only into account the histograms which have enough information about the target. If the value exceeds a threshold τ_A, the target is recovered. Figure 16 shows an example of this technique. When the comparison is made, two different targets are lost, and the system is able to distinguish between them and assign the correct identification.

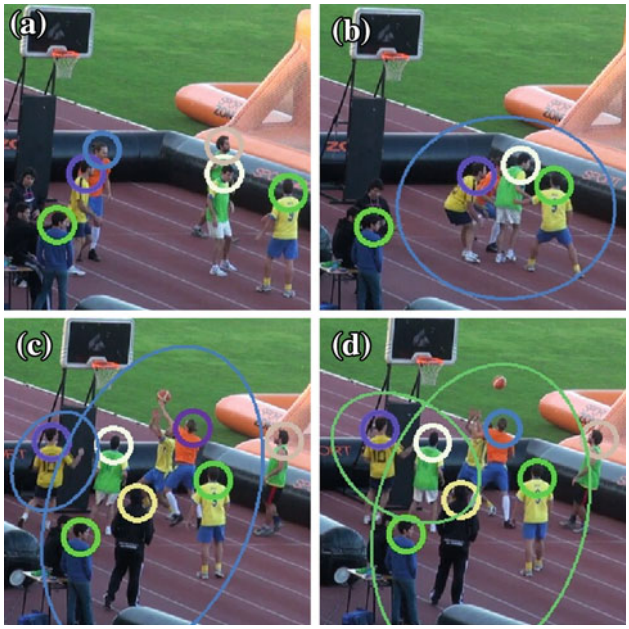


Fig. 16 Target recovery. The *blue target* is instantiated (**a**) and, later, lost (**b**). A new target is instantiated within the region in which the lost target could reappear (**c**) and, after the comparison is made, the target is recovered (**d**) (color figure online)

4 Experimental results

In our experiments we use a full video sequence of a sport event, consisting in a 3×3 basket match. The video used in this test have a 640×368 resolution running at 25 frames per second. Although it is not a heavily crowded scene, it is a very challenging environment, since multiple situations occur that usually do not happen under usual scenarios, including collisions, crossings, sudden orientation changes, jumps or squatting people. The main idea is to test the tracking system under a scenario of such difficulty. More than 15,000 frames were recorded.

The sequences used to test this algorithm involve isolated people, grouping and splitting events, and occlusions behind a crowd. As mentioned in previous sections, this

Table 1 Quantitative results on the basket sequence for our tracking system

Different configurations are shown, depending on the modules enabled

Method	MT	PT	ML	FM	IDS
Our tracking with no collision/recovery system	2	10	1	23	8
Our tracking with no background information	2	11	0	18	10
Our tracking with no HOG assist	3	9	1	31	10
Our tracking	8	4	1	24	8

Table 2 Tracking performance on the basket sequence for our tracking system

Different configurations are shown, depending on the modules enabled

Method	MOTP (%)	MOTA (%)	Prec (%)	Rec (%)
HOG head detections	75.6		27.5	55.3
Our tracking with no collision/recovery system	77.5	43.4	48.8	79.7
Our tracking with no background information	84.5	48.1	50.6	92.0
Our tracking with no HOG assist	80.7	64.0	85.0	73.9
Our tracking	84.5	73.5	88.2	82.0

algorithm performs at a better speed when the distance between the targets and the camera is high, as occurs in this videos. Therefore, we can assume that the head-shoulder size is approximately the same at every position in the scene.

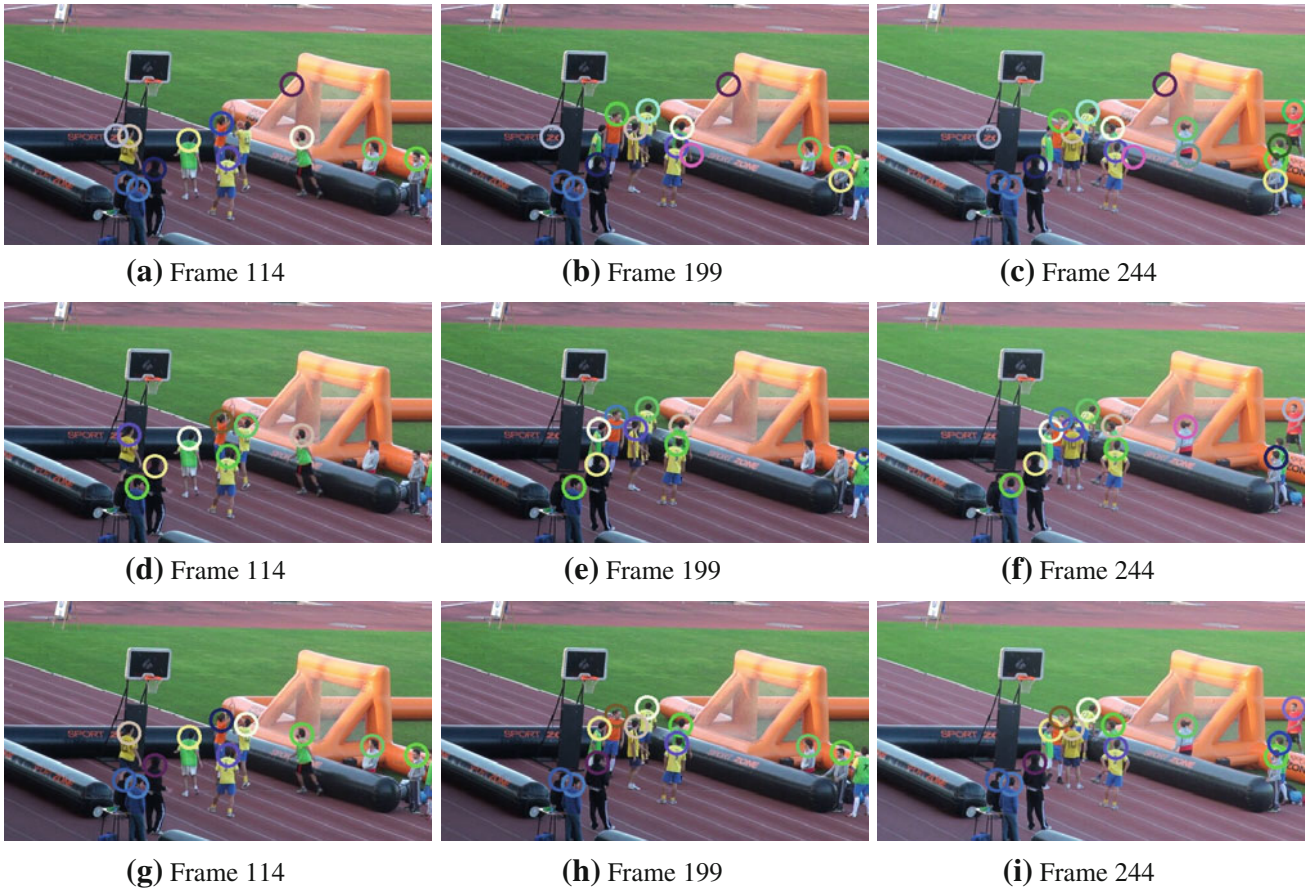
To train both Viola–Jones type classifier and SVM, we use the generic dataset introduced in [16]. We assume the head-shoulder detections in the scene has 32×32 size, which is the same size used in the training dataset. To perform the HOG feature extraction, we divide each sample into 64 cells. Then, four adjacent cells form a block. Using a one cell stride, we obtain 49 different blocks. For each cell, a histogram of eight different orientation bins is computed and stored. Each block is also normalized.

We set the threshold values $\tau = 5.0$ for the tracking using the people detection technique, $\alpha = 0.05$ for the HOG feature learning factor, $\gamma_S = 0.8$ for the particle filter quality threshold, $\tau_S = 2$ for the particle filter limit for successive bad estimations and $\tau_A = 0.85$ as the occlusion recovery value. In addition, 50 different particles are thrown into the particle filter system.

To evaluate our system in a quantitative way, we have performed a test using standard evaluations tools. We calculate the CLEAR-metrics introduced by [24]: MOTA (multiple object tracking accuracy), which take into account false positives, missed targets and identity switches, and MOTP (multiple object tracking precision), which measures the average distance between true and estimated targets. A predicted bounding box is considered correct if it overlaps more than 25 % with a ground-truth bounding box. We also provide system precision and recall metrics.

A ground truth data is created, containing more than 10,000 annotated heads. As we mentioned before, both Viola–Jones type classifier and SVM are trained using the generic dataset introduced in [16]. This means that no specific information about the camera pose are included. In Table 2, we can show this type of training causes the HOG classifier to obtain a poor precision. However, we decided not to modify these trainings to show how our method is able to improve the tracking performance under bad HOG detections. Also we test different system configurations, disabling some of its features. As Tables 1 and 2 clearly show, both the Collision/Recovery and Background systems reduce the number of false positives, obtaining low MOTA values when they are disabled. Furthermore, the inclusion of the HOG detections in the tracking system clearly improves the system behavior (close to 10 %). Although the number of switch identifications is similar in the different configurations we have tested, the recovery system is able to detect the switch and revert the situation, improving both MOTA and accuracy of the system. There is only one mostly lost target, that is difficult to detect because it appears behind the goal, causing the net to interfere in the correct detection.

We also show qualitative results of different aspects related to the proposed algorithm. First, we are going to describe the performance of the detection system. A sequence with multiple targets is used to identify the number of people the system can recover under different configurations. In first place, we use an approach which does not take into account neither background subtraction nor torso information, used to discard some detections, as explained in Sect. 2.3. We include three different measurements: the number of patches passing both Viola–Jones and SVM classifiers, the number of patches corresponding to a person and the ground truth, which includes all the targets within the scene, no matter if they are occluded or not. The system is able to recover almost all the people in the scene. Most of the non-detection problems are related to occlusion events. However, multiple false positives are introduced, adding several noise to the people detection.



(a) Frame 114 **(b)** Frame 199 **(c)** Frame 244

(d) Frame 114 **(e)** Frame 199 **(f)** Frame 244

(g) Frame 114 **(h)** Frame 199 **(i)** Frame 244

Fig. 17 People detection examples. Without background information (**a**–**c**). With background information (**d**–**f**). Hybrid approach (**g**–**i**). **a** Frame 114, **b** frame 199, **c** frame 244, **d** frame 114, **e** frame 199, **f** frame 244, **g** frame 114, **h** frame 199, **i** frame 244

Although, when introducing the background subtraction technique, along with the torso estimation, the number of false positives is avoided. Every detection in the scene matches with a correct person detection. However, the number of persons detected in the scene is reduced. Since we do not have any frame in the sequences without people, the system cannot recover stopped people. So, to obtain a better representation we introduce an hybrid simulation. We avoid the background information along the first frames, enabling it later. Thus, we can detect stopped people. The results show that the number of people detected within the scene even exceeds the results obtained when no information about the background is included. Plus, the number of false positives is very low. Examples of those results can be seen in Fig. 17.

No isolated people tracking tests were conducted, since the particle filter-based visual tracking was tested in other application [17], obtaining good results. However, note that the velocity prediction using linear filters (Adalines) obtain a better estimation, while successfully coping with the noise introduced in the position measure along frames [5]. So, we focus in two different situations: partial occlusions in crowded scenes and identification recovery using the ellipse representation in combination with the fixed pool of histograms.

The challenging situation in crowded scenes is to deal with partial occlusions. Since people heads are really close together, it is very difficult to use body-color information to solve this task. So, the tracking procedure must be carried on by the particle filter system along with the tracking using the people detection system. Figure 18 shows an example of four people moving close together. As we can see in frame 693, there exists an overlap between multiple target detections. In less than 100 frames (4 s), all the four targets are grouped and split several times, having not a single total head-shoulder occlusion. Along the successive situations, the system is able to maintain every target identification, even in high overlapping situations, like in frame 753, where the yellow and the white target are really close together. Also, the detection quality remains good enough to continue the tracking after the splitting event.

We also check the occlusion recovery procedure. As mentioned before, particle filter system has no possibility to recover a previous identification when a total occlusion

Fig. 18 People detection in crowded scenes. Partial occlusions are solved without using color information. Particle filter system is robust under these situations

event occurs. This issue usually involves a collision between two or more targets. Three different examples are shown in Fig. 19. The system is able to detect the collision and assign the correct identification to the people which are nearest to the camera. Once the occluded target is tracked and trained again, the pool of histograms, in combination with the ellipse estimation, are able to re-detect the person.

This methodology was tested in a Pentium Quad Core running at 2.40 GHz with 4 RAM GB in a Linux Operating System. The videos used in this test have above 11 different targets in the image. As Table 3 shows, the system can process over 6 frames per second, without any parallelism, including the tracking system, in combination with the occlusion recovery methodology, that can process every frame in only 20 ms, without taking into account the time needed to process the integral information, which is computed in the detection step. Furthermore, the inclusion of different techniques like the background subtraction and the torso estimation not only improve the accuracy of the system but also the speed.

5 Conclusions

In this work, an architecture for fast multiple people tracking under uncontrolled scenarios is presented. A combination of two different classifiers, a Viola–Jones and a SVM, are used to detect every person in the scene because of the head-shoulder omega-shape feature. A background subtraction technique is used to restrict the image location in which the classifiers are used, and also to perform a torso estimation to confirm the positive solutions.

Two different trackers are used. A feature-based particle filter system, in combination with a velocity prediction system based on linear filters, is used to track the head-shoulder shape along the frames. A high-level tracker, which stores every people's appearance into a fixed pool of *L*a*b*-space

Fig. 19 People recovery before collision events. The system is able to recover the previous identification after the event occurs

Table 3 Time consuming by the detection system on a 640 × 368 image by the Viola–Jones type classifier in combination with the HOG feature-based SVM

	Detector type	Time per frame (ms)
Without head-size estimation	No background subtraction (dense scan)	990
	Background subtraction	305
	Background subtraction + torso estimation	278
With head-size estimation	No background subtraction (dense scan)	223
	Background subtraction	157
	Background subtraction + torso estimation	156

color histograms, is used to recover identifications which are previously lost, usually because of occlusions of bad particle filter estimations. Controlling both trackers, a system management is responsible for identifying each target state, and deciding which tracker must be launched depending on the situation. It also has the ability to remove bad detections and to recover the previous target identification during occlusion events. An ellipse representation is used to estimate the position of the occluded targets, which considerably reduces the regions in which the target could reappear.

This method was tested over a 10-min sport video sequence, which has multiple challenging situations, such as sudden orientation changes, occlusions behind other player, jumps, squatting movements, and more. Multiple examples indicate the performance of this technique, while some stats show the improvement of this methodology with respect to other similar people detection techniques. Furthermore, the histogram-based tracker included in the methodology, along with the ellipse estimation results in a powerful technique to recover occluded target identifications.

The results also show that the system can reduce, compared to other similar techniques, the false positives in the detection procedure, because of the background subtraction information included. Also, collision detection and target

identification recovery under occlusion events are included, which are issue that have not been addressed before in systems like this one. The procedure speed indicates that is possible to obtain a real-system framework using parallelism techniques.

This approach highly improves processing times of previous approaches to this topic. Since the techniques used in this methodology are mainly computed pixel per pixel, the inclusion of GPU programing techniques could derive in a real-time system for tracking people.

As limitations, the use of a background subtraction technique to reduce the space to perform the detection system causes the system not to locate the stopped targets in the scene, so a background training period is needed to obtain a better performance. Furthermore, highly crowded scenes could cause the system to operate similar to a system without background information, increasing the processing time. Sudden illumination changes must also be controlled to obtain a good foreground representation. This is a general methodology for tracking people. Those results could be improved by using local information about the scene to train the classifiers, instead of using a generic head-shoulder feature dataset. Furthermore, it could be also modified to track any other feature in the scene. These properties enable the system to be used as an information source in a higher-level behavior analyzer, such as human interactions or detection of abnormal trajectories.

Acknowledgments This paper has been partly funded by the Ministerio de Ciencia e Innovación through grant contract TIN2011-25476.

References

1. Andriyenko, A., Schindler, K.: Multi-target tracking by continuous energy minimization. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1265–1272. IEEE, New York (2011)
2. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1926–1933. IEEE, New York (2012)
3. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking. IEEE Trans Signal Process **50**(2), 174–188 (2002)
4. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3457–3464. IEEE, New York (2011)
5. Cancela, B., Ortega, M., Fernández, A., Penedo, M.G.: Hierarchical framework for robust and fast multiple-target tracking. Expert Syst. Appl. **40**, 1116–1131 (2013)
6. Cancela, B., Ortega, M., Penedo, M.G.: Human detection and tracking under complex activities. In: 8th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2012) (2013)
7. Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. PAMI **27**(10), 1631–1643 (2005)
8. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000, vol. 2, pp. 142–149 (2000)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005 (CVPR 2005), vol. 1, pp. 886–893 (2005)
10. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: IEEE 12th International Conference on Computer Vision, 2009, pp. 229–236 (2009)
11. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
12. Haritaoglu, I., Harwood, D., Davis, L.: W4: Who? when? where? what? a real time system for detecting and tracking people. In: Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 222–227 (1998)
13. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. Int. J. Comput. Vis. **80**, 3–15 (2008)
14. Kalman, R.: A new approach to linear filtering and prediction problems. Trans. ASME J. Basic Eng. (Ser. D) **82**, 35–45 (1960)
15. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 120–127. IEEE, New York (2011)
16. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 19th International Conference on Pattern Recognition, 2008 (ICPR 2008), pp. 1–4 (2008)
17. Li, M., Zhang, Z., Huang, K., Tan, T.: Rapid and robust human detection and tracking based on omega-shape features. In: 16th IEEE International Conference on Image Processing (ICIP), pp. 2545–2548 (2009)
18. Marieb, E.N., Hoehn, K.: Human Anatomy and Physiology. Pearson Education, San Francisco (2007)
19. Porikli, F.: Integral histogram: a fast way to extract histograms in cartesian spaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005 (CVPR 2005), vol. 1, pp. 829–836 (2005)
20. Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.Y.: Density-aware person detection and tracking in crowds. In: Proceedings of the International Conference on Computer Vision (ICCV) (2011)
21. Rohr, K.: Towars model-based recognition of human movements in image sequences. CVGIP Image Underst. **59**(1), 94–115 (1994)
22. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994 (CVPR '94), pp. 593–600 (1994)
23. Stalder, S., Grabner, H., Van Gool, L.: Cascaded confidence filtering for improved tracking-by-detection. In: Computer Vision-ECCV 2010, pp. 369–382. Springer, Berlin (2010)
24. Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D., Soundararajan, P.: The clear 2006 evaluation. In: Multimodal Technologies for Perception of Humans, pp. 1–44. Springer, Berlin (2007)
25. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001 (CVPR 2001), vol. 1, pp. I-511–I-518 (2001)
26. Yang, B., Nevatia, R.: An online learned crf model for multi-target tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2034–2041. IEEE, New York (2012)

27. Yang, B., Nevatia, R.: Online learned discriminative part-based appearance models for multi-human tracking. In: Computer Vision-ECCV 2012, pp. 484–498. Springer, Berlin (2012)
28. Zhan, B., Monekosso, D., Remagnino, P., Velastin, S., Xu, L.Q.: Crowd analysis: a survey. Mach. Vis. Appl. **19**, 345–357 (2008)
29. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004 (ICPR 2004), vol. 2, pp. 28–31 (2004)

2.3 Conference Paper: Open-world person re-identification by multi-label assignment Inference

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Timothy M. Hospedales
Affiliation: Queen Mary University of London, UK

Co-author #3: Shaogang Gong
Affiliation: Queen Mary University of London, UK

Article title: Open-world person re-identification by multi-label assignment Inference
Conference: British Machine Vision Conference (BMVC)
Location: Nottingham, UK
Date: September, 2014

Open-World Person Re-Identification by Multi-Label Assignment Inference

Brais Cancela[1]
brais.cancela@udc.es
Timothy M. Hospedales[2]
t.hospedales@qmul.ac.uk
Shaogang Gong[2]
s.gong@qmul.ac.uk

[1] VARPA Group,
Universidade da Coruña,
A Coruña 15071, Spain

[2] School of EECS,
Queen Mary University of London,
London E1 4NS, U.K.

Abstract

Person re-identification methods have recently made tremendous progress on maximizing re-identification accuracy between camera pairs. However, this line of work mostly shares an critical limitation - it assumes re-identification in a 'closed world'. That is, between a known set of people who all appear in both views of a single pair of cameras. This is clearly far from a realistic application scenario. In this study, we take a significant step toward a more realistic 'open world' scenario. We consider associating persons observed in more than two cameras where: multiple within-camera detections are possible; different people can transit between different cameras – so that there is only partial and unknown *overlap of identity* between people observed by each camera; and the total number of unique people among all cameras is itself unknown. To address this significantly more challenging open world scenario, we propose a novel framework based on online Conditional Random Field (CRF) inference. Experiments demonstrate the robustness of our approach in contrast to the limitations of conventional approaches in the open world context.

1 Introduction

The task of re-identification (ReID) is often defined as the recognition of the same individual at different times and locations, which may involve different cameras, views, poses and lighting. This challenge is now widely studied by the computer vision community, due to its fundamentally challenging nature, and important practical role underpinning many visual surveillance functionalities including person search and tracking across disjoint cameras.

Re-identification studies generally frame the task as a closed set matching problem. Given a predefined 'gallery' set of known individuals, systems try to label each new 'probe' detection with the identity of the matching gallery individual. Studies have investigated good feature representations [4] and discriminative models [9] to maximise the chance of correct matching. They considered the contexts of single-shot [4, 9] (one image per person per camera) as well as multi-shot [12] (a series of images per person per camera, obtained from tracking) scenarios. However, most studies share two very strong assumptions: *the total number of people in the scene is known* a priori, and there exists a total *overlap of identity*

© 2014. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

between a camera pair, that is, every person appears in both camera views. Although this constrained framing of the ReID problem is a good starting point, it is unrealistic for real-world re-identification scenarios, when there is no prior information about the same people reappearing in the scene at different views. We refer to this unconstrained setting as the 'open world' ReID problem. The open-world problem is more challenging for two reasons: (i) the total number of unique people within each camera and the scene as a whole (cross-cameras) are both unknown, and (ii) each subject may appear in some unknown subset of the cameras.

The closed-world problem is significantly simpler, because it can be divided into a series of independent tasks: "*For each probe person, find the top most similar in the gallery*". In the unconstrained variant, if there are two cameras people may only be seen by one; or if there are more than two, then people may appear in any subset of the cameras. This means there are more possible outcomes (no match), and every unknown identity problem is no longer independent, they become strongly inter-related. For example, consider intuitively the task of trying to match a person with a red-shirt against a gallery in the conventional closed world context. The match is simply the one whose shirt most clearly red. In the open world scenario, these could be completely separate people if two distinct red-shirt people were observed independently in each camera and not in the other. Moreover, if there are two red-shirt probes: in a closed world context, these would be given as distinct. In an open-world context there is additional ambiguity: Are they distinct people, or due to a broken track? The classical approaches clearly make too strong assumptions for this type of scenario.

In this paper we consider for the first time the most general open-world re-identification problem, where there is no prior information about the number of people or their overlap of identity across cameras. To address this, we introduce a new Conditional Random Field (CRF) model, overcoming the entailed challenges of effective graph construction, local optima and efficient inference. Our framework can answer qualitatively more general queries than existing re-identification systems such as: "*How many people are in the scene?*", "*If a person leaves a camera, which other cameras did he appear in, or did he simply disappear?*".

1.1 Related Work

Closed World Re-Identification: There has now been extensive work on closed-world re-identification (ReID). Studies have generally addressed good feature representations [4, 15, 16, 20] and/or learning matching models discriminatively [7, 9, 15, 16]. Most works have considered the 'single-shot' scenario of exactly one image per person using datasets like VIPeR [7]; while others considered 'multi-shot' – how to constructively aggregate information from multiple detections/shots of each person that might be obtained from tracking – using datasets like ETHZ [3]. Further review is beyond scope of this work, so we point the reader to a recent book [6] and survey [18] that summarise the main issues [5].

Towards Open World ReID: Going beyond closed world ReID discussed above, a few recent studies have begun to consider some open-world aspects of ReID. For instance, [12] introduced a CRF model to address multi-shot re-identification when the shots are not assumed to be correctly pre-grouped within each camera: Corresponding to realistic input with track association errors and split detections. Temporal information from each shot is used to restrict the connections between the nodes of the CRF. The system is only tested with the ETHZ dataset, which is recorded using a moving camera. However, pose and illumination variation is not high, and the more constrained assumption of full overlapping person sets is made. Recently, [11] introduces a probabilistic graphical model to associate within-camera trajectories across disjoint cameras. This model reasons generatively about

the appearance of each person, lighting change between cameras and the association between trajectories. Efficient Gibbs sampling is used to find the best solution. However, it still requires prior-knowledge of the number of people in the scene, and unlike [12], it assumes that within camera association is already performed perfectly. No existing work has considered the fully unconstrained open-world problem addressed in this work where within-camera re-identification is not assumed a-priori, person identities only partially overlap across two or more cameras, i.e. no guarantee of all people reappearing in every camera view, and the total number of persons is unknown. In [21], a transfer learning framework is defined to verify a probe person against a set of targets against a large amount of unlabeled data. However, this assumes the target and background people are split a-priori, it reasons about a single probe person at a time instead of jointly about all probes, and it only applies within two cameras.

Set Association: Although the open world scenario has not been addressed before, some existing algorithms are related to this challenge. The Hungarian Method (HM) [14] performs a set-match and can find the best pairwise correspondence between two sets of detections. It is a good solution for the closed world single-shot problem. However it will find an association even if the two sets are partially overlapped or totally disjoint (i.e. none reappearing). Thus even if every person in camera A does not appear in camera B, HM obtains a complete set of matches. Moreover, it cannot deal with multi-shot as it only makes 1:1 connections. We will exploit the HM to define a subset of credible matches for a global CRF to reason about. A classical CRF model [2] could also be used, with pairwise similarity measures to weight links between detections. However, several problems arise: (i) how to define the graph structure and label space, and (ii) CRFs tend to minimise the number of distinct labels used, thus tending to assign every detection to one identity. In this work, we develop a novel CRF model that incrementally constructs an appropriate graph to address these issues.

Our Framework: Contrary to classical ReID, the challenge in an 'open world' scenario also includes within-camera association, i.e. encompassing within-camera ambiguity due to tracking errors. An open world model not only has to distinguish when two detections belong to the same person, as in classical ReID, but also has to recognise when a new person enters in the scene, as in a classical tracking system. We build on CRFs, as they are state-of-the-art solvers for closely-related topics of re-identification [12] and tracking [19]. However, we relax the conventional constraint on requiring a priori set of known labels, and address issues in efficiency and convergence. Specifically, we introduce a novel two-step CRF model, that exploits spatio-temporal information where available. The first step matches within camera detections that belong to the same person. The second step considers both within and across-camera matching, using inter-camera information to revise initial within-camera estimates.

The proposed model makes three important contributions: (1) No label information is needed a priori, allowing the system to detect when a new person enters the camera network; (2) An 'open world' solver, that is, the model does not assume that a person will (re)appear in every camera; and (3) Producing a person count as a byproduct. Our approach enables the flexibility lacking in existing state of the art closed world ReID solutions. Finally we also discuss some different evaluation criteria, as the classic Cumulative Matching Criteria (CMC) that assumes known number of people in a ReID scenario is no longer suitable.

2 A Framework for Open World Re-Identification

In this section we first formalise the task and our model representation. In this model, different candidates of people with unknown id labels are represented as nodes in a CRF. The

objective of the CRF is to infer the most likely correct assignment of multiple id labels simultaneously to all the nodes in the CRF (see Figure 1). We assume as input a set of N observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ across different camera views. Each observation $\mathbf{x}_i = \{c_i, t_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i\}$ consists of: A camera c_i making the detection; the time of detection t_i (we assume cameras are synchronized); the image position $\mathbf{p}_i$ and velocity $\mathbf{v}_i$ where the person was detected; and an appearance feature $\mathbf{a}_i$ from the detection bounding box. The re-identification task is to correctly assign identity labels $\mathcal{L} = \{l_i\}_{i=1}^{N}$, $l \in 1 \ldots L$ to all detections..

To address this task we propose a CRF $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with the following structure. Each node corresponds to a person detection (observation) $\mathcal{V} = \{v_i = x_i\}$. Each edge corresponds to a similarity between nodes/persons $\mathcal{E} = \{e_{ij} = (v_i, v_j)\}$, and the label of each node corresponds to the identity of that person/detection. Our aim is to find the set of labels $\mathcal{L}$ that best fits all the observations $\mathcal{X}$,

$$\mathcal{L}^* = \arg\min_{\mathcal{L}} \left(\sum_i U(l_i|\mathcal{X}) + \sum_{ij} B(l_i, l_j|\mathcal{X}) \right) \tag{1}$$

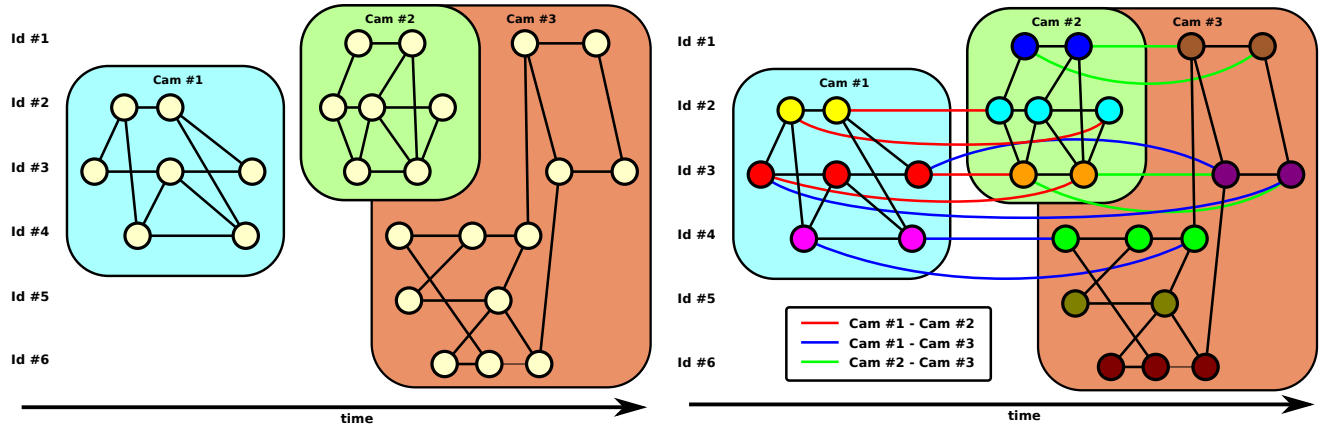
Here $U(l_i|\mathcal{X})$ and $B(l_i, l_j|\mathcal{X})$ denote unary and pairwise energy functions, respectively. U is an $L \times N$ matrix defining the cost of assigning any label l_i to any observation $\mathbf{x}_i$. Importantly, in the open world context, we do not know the total number of people in the network, so $L = N$ to account for the limiting case where every single detection is a unique person. B is also an $N \times N$ matrix, defining the cost of assigning l_i and l_j to a particular pair of observations. We decompose B into two matrices, $B(i, j) = W(i, j)C(\mathcal{L}(i), \mathcal{L}(j))$, where $W(i, j)$ is the weight of the similarity between the two nodes x_i and x_j and $C(\mathcal{L}(i), \mathcal{L}(j))$ is the cost of assigning the labels $\mathcal{L}(i)$ and $\mathcal{L}(j)$ to their respective nodes. We shall define W later, whilst C is a $N \times N$ matrix defined as

$$C(l_i, l_j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

As mentioned before, U is an $N \times N$ matrix. Thus, assuming that we have as many labels as person detections (observations), the cost of assigning any label l_j to any observation $\mathbf{x}_i$ is also a pairwise similarity measure between the observations $\mathbf{x}_i$ and $\mathbf{x}_j$. $B(i, j) = 0$ means there is no direct connection between the two detections. Non-zero values will depend on the appearance features, and spatio-temporal information (if available). The accuracy of pairwise correspondences is higher within the same camera than between cameras, due to less appearance change and stronger continuity. For this reason, our algorithm proceeds in two steps, as illustrated in Fig. 1. First, we solve the CRF allowing connections only between detections within the same camera. Second, we use that solution as an initial condition to build the connections between different cameras, creating the final CRF model. The structure and parameterisation of CRF at each stage is the same, but additional information is included.

2.1 Label Assignment as Within-Camera Tracking

A characteristic of CRF models is that they try to reduce the number of labels in the output. For that reason, while creating a fully-connected CRF for solving the open-world re-identification problem is elegant, it is very hard to tune. Small variations in the pairwise potential causes every connected detection to be grouped with the same label, even if the similarity is low. Thus, we restrict the number of direct links between detections.



(a) Step 1 (b) Step 2

Figure 1: CRF illustration. In the first step, only detections within the same camera are connected. In the second step, a restricted connection between cameras is allowed.

First, all the detections included in the observation set are sorted according to the time they were detected. Then, we establish the similarity between detections by creating the unary potential $\tilde{U}$, defined as

$$\tilde{U}(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1 - \delta^c_{i,j} & \text{if } |t_i - t_j| < \tau_c \text{ and } c_i = c_j \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

where $\delta^c_{i,j} \in [0,1]$ is the probability of assigning label l_i to the detection x_j in camera c. Similarly, the pairwise weight $\tilde{W}$ is defined as

$$\tilde{W}(i,j) = \begin{cases} \left(1 - \dfrac{|t_i - t_j|}{\tau_c}\right) \alpha^c_{i,j} & \text{if } |t_i - t_j| < \tau_c \text{ and } c_i = c_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where t_i and t_j are the times that detections x_i and x_j were recorded, respectively; $\alpha^c_{i,j} \in [0,1]$ is the appearance similarity between detections i and j in camera c; and τ_c a time threshold. Note that the strength between two detections decreases with the time gap similarly to [12].

As explained before, the number of connections between the nodes, using these matrices, is too high. A fully-connected CRF tends to use fewer labels, which is an undesirable property for our model. Thus, we reduce the number of direct connections to two at most for each detection based on higher $\tilde{W}(i,j)$ values. First, we define $\tilde{W}_w$ and $\tilde{U}_w$ as

$$\tilde{W}_w(i,j) = \begin{cases} \tilde{W}(i,j) & \text{if } |P| < 2, \text{ where } P = \{x \in N - \{i\}, \tilde{W}(i,x) > \tilde{W}(i,j)\ \} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\tilde{U}_w(i,j) = \begin{cases} \tilde{U}(i,j) & \text{if } |P| < 2, \text{ where } P = \{x \in N - \{i\}, \tilde{W}(i,x) > \tilde{W}(i,j)\ \} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Two links per node is a good balance. This value can be modified, but we found this is a good connection density (higher values highly increase the false positive rate, whilst a lower value increase the false negative rate). To enforce symmetry, we define W_w and U_w as

$$W_w = \tilde{W}_w + \tilde{W}_w^T \qquad U_w = \tilde{U}_w + \tilde{U}_w^T \tag{7}$$

U_w, W_v and C define our CRF, which can be solved efficiently using the alpha-expansion algorithm [2]. At this point, we have connections between nodes (associated person detections) in the same camera, which are denoted by G. Next, we establish links across cameras.

Input: $U_w, W_w, \{H\}$
Output: U, W
begin
  $U = U_w, W = W_w, T = \emptyset$.
  foreach $c_1, c_2 \in |c|, c_1 \neq c_2$ **do**
    $[p,q] = \text{Hungarian}(H^{c_1,c_2})$.
    for $i = 1..|p|$ **do**
      if $H^{c_1,c_2}(p_i,q_i) > \alpha_t^{c_1,c_2}$ **then**
        $W(p_i,q_i) = W(q_i,p_i) = \frac{f^{c_1,c_2}}{\max(\{f^{c_i,c_j}\})}$.
        $T \cup (p_i, q_i, \frac{f^{c_1,c_2}}{\max(\{f^{c_i,c_j})\}})$.
      end
    end
  end
  for $i = 1..|T|$ **do**
    Without taking into account (p_i,q_i) connection:
    Select $S_i(p) | \forall j \in S_i(p)$, if exists a path between p_i and j using W
    Select $S_i(q) | \forall j \in S_i(q)$, if exists a path between q_i and j using W
    Update the states $U(S_i(q), S_i(p))$ and $U(S_i(p), S_i(q))$.
  end
end

Algorithm 1: Constructing unary and binary CRF potentials.

2.2 Cross-Camera Association

To simplify association across cameras, we only take into account direct connections between the first and the last appearance of a person in each camera. Let $\mathcal{L}_v$ be the labels associated with each node after using the local CRF model. Given the sorted detections, we create two sets B and E enclosing the first and the last label appearances, as follows:

$$\forall p \in [1..N] \quad p \in B \quad \text{if } \forall q \in [1..(p-1)], G(q) \neq G(p) \tag{8}$$

$$\forall p \in [1..N] \quad p \in E \quad \text{if } \forall q \in [(p+1)..N], G(q) \neq G(p) \tag{9}$$

Once we have these sets, we need to select which are the correct matches between the detections. With the same reasoning as before, we want to reduce the number of connections between detections. Assuming the labels obtained in the first step are correct, we can conclude that the final detection of each person in each camera occurs when the subject leaves the camera field of view. The same also happens with the initial detections. Based on this reasoning, we can conclude that every final detection in one camera is related with, at most, one detection in another camera. Thus, for each pair of cameras c_1 and c_2, we create the matrix H^{c_1,c_2}, which stores the affinity between detections i and j, as

$$H_{i,j}^{c_1,c_2} = \begin{cases} \beta_{i,j}^{c_1,c_2} & \text{if } c_i = c_1 \wedge c_j = c_2 \wedge ((i \in B \wedge j \in E) \vee (i \in E \wedge j \in B)) \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

where β^{c_1,c_2} is a cross-camera pairwise person-affinity measure based on appearance and spatio-temporal cues. The lower the β value, the stronger connection. Using the Hungarian Method [14], we search for the most plausible assignment of correct labels. In other words, the Hungarian method is used to find a small subset of plausible links between detections in different cameras. The detected links are included in the CRF as explained in Algorithm 1. In the first loop, we compute the Hungarian Method to obtain the connections between nodes in different cameras. Then, for each pair of connected nodes, we remove that connection and we look for all the connections each node has. So, we obtain all the different states each node can have, without taking into account the new connection. Finally, we enable the connection and we update the unary potential of all the connected nodes, updating the new

Input: Detections $\mathcal{X}$
Output: Associations between detections $\mathcal{L}$
begin
  Compute within camera weights W and U (Eq. 7),
  Solve the CRF Eq (1) with Alpha-expansion [2]
  Solve Initial Hungarian to obtain H (Eq. 10),
  Compute across camera weights W and U (Alg. 1)
  Solve the CRF Eq (1) with Alpha-expansion [2])
end

Algorithm 2: Overview of CRF algorithm for open-world ReID.

states the nodes can reach with this new connection. The weight of this connection is further adapted by the expected quality/reliability of affinities computed between these two cameras according to the estimated F_1 score f^{c_1,c_2} across cameras:

$$W(i,j) = \begin{cases} \tilde{W}(i,j) & \text{if } c_i = c_j \\ \frac{f^{c_1,c_2}}{\max(\{f^{c_i,c_j}\})} & \text{if } c_i \neq c_j \text{ and } x_i \text{ and } x_j \text{ are linked} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

This is done because we want to rely more on connections between camera pairs that can match reliably, and less on unreliable pairs. Finally, we create the CRF using the matrices U, W and C. To solve this CRF, we use the alpha-expansion algorithm again. An overview of our two-step CRF algorithm is described in Algorithm 2.

2.3 Pairwise Affinity Measures

The model depends on pairwise within and across-camera similarity measures δ^c (Eq. 3), α^c (Eq. 4) and β^{c_1,c_2} (Eq. 10). These are all learned in the training step.

Within-camera: Various techniques can be used to compute similarities: $\delta^c, \alpha^c \in [0,1]$. For simplicity, we assume $\delta^c = \alpha^c$. To obtain these, we train a pairwise appearance-based person-similarity model $d^c(\cdot,\cdot)$ per camera. Let λ^+ be the set containing all the matching pairs, whereas λ^- the opposite; and $d^c(\mathbf{a}_i,\mathbf{a}_j)$ some pairwise distance metric (KISS or distance to the hyperplane in the RankSVM model). To normalise the distances for comparability across cameras, δ^c and α^c are then defined as

$$\alpha^c_{i,j} = \delta^c_{i,j} = \frac{|(\mathbf{a}_l,\mathbf{a}_m) \in \lambda^+, d^c(\mathbf{a}_l,\mathbf{a}_m) \leq d^c(\mathbf{a}_i,\mathbf{a}_j)|}{|(\mathbf{a}_l,\mathbf{a}_m) \in \lambda^+, d^c(\mathbf{a}_l,\mathbf{a}_m) \leq d^c(\mathbf{a}_i,\mathbf{a}_j)| + |(\mathbf{a}_n,\mathbf{a}_p) \in \lambda^-, d^c(\mathbf{a}_l,\mathbf{a}_m) \leq d^c(\mathbf{a}_i,\mathbf{a}_j)|} \tag{12}$$

Across-camera: To obtain the across-camera measure for cameras c_1 and c_2 with respective detections x_i and x_j, we compute KISS or RankSVM similarity measures: one for appearance ($d^{c_1,c_2}(\mathbf{a}_i,\mathbf{a}_j)$), and another one for the combination of both position and velocity ($\rho^{c_1,c_2}(\mathbf{p}_i;\mathbf{v}_i,\mathbf{p}_j;\mathbf{v}_j)$). We combine the two distances to obtain the similarity measure

$$\beta^{c_1,c_2}_{i,j} = \gamma^{c_1,c_2} d^{c_1,c_2}(\mathbf{a}_i,\mathbf{a}_j) + (1-\gamma^{c_1,c_2})\rho^{c_1,c_2}(\mathbf{p}_i;\mathbf{v}_i,\mathbf{p}_j;\mathbf{v}_j) \quad \gamma^{c_1,c_2} \in [0,1] \tag{13}$$

3 Experiments

Dataset: To evaluate our contribution, we need a dataset that reflects the open-world challenge. Many classic ReID datasets, such as VIPER or ETHZ assume total overlap of persons across cameras. PrID dataset [8] has a multiple-shot version with partial overlap, but it contains only two different cameras. Thus, we decide to focus on the challenging SAIVT-Softbio

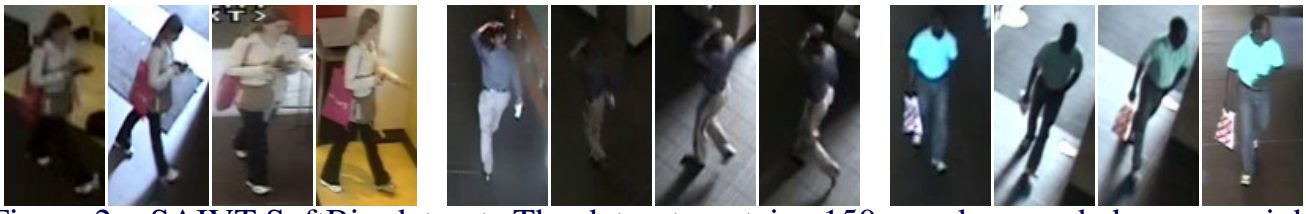


Figure 2: SAIVT-SoftBio dataset. The dataset contains 150 people recorded over an eight camera network. It includes several angle orientations and sudden illumination changes.

database [1] (see Fig. 2), that includes 150 people recorded using 8 different cameras. To our knowledge, it is the only dataset that simultaneously meets all the requirements for a full open-world task: Multi-shot data and multiple cameras with camera-transition uncertainty.

Experimental Settings: Our contribution is agnostic to the appearance feature, and the base pairwise matching model used. To evaluate the system, we divide the dataset into 3 disjoint subsets. The first third (train set), is used to train all pairwise within and across-camera matching models, giving d and ρ in Eqs. (13)-(16). We consider the ELF [17] feature with RankSVM [10, 17] and KISS[1] [13] pairwise models. The second portion (calibration set) is used to calibrate all thresholds, α^c, δ^c, and β^{c_1,c_2} measures; and γ^{c_1,c_2} values. The best combination of these parameters is obtained by looking for the best F-score, denoted by f^c or f^{c_1,c_2}, depending if its within or between cameras. The final third is used to evaluate the performance. We average performance over 10 random splits.

Baselines: As we address the open world problem with no prior information about the number of people or their camera overlap, no existing models directly apply. For baselines, we therefore define a more conventional 'engineering' generalisation to open world of RankSVM [10, 17] and KISS [13]. We train both on the training set, and then use the calibration set to optimise the threshold for the pairwise affinity. Pairs with affinity over threshold are declared as sharing the same label. We denote these NaiveRankSVM and NaiveKISS.

Evaluation Metrics: To evaluate the performance of open-world problems the conventional CMC metric is insufficient, due to partial overlap of a variable number of labels and > 2 cameras. We therefore apply statistical analysis: Given the final and ground truth labels, $\mathcal{L}^*$ and $\mathcal{L}_{gt}$, we analyse all pairs. If two nodes have the same label in $\mathcal{L}_{gt}$ and in $\mathcal{L}^*$, it is a true positive. The same label in $\mathcal{L}^*$ and different in $\mathcal{L}_{gt}$, a false positive, and so on. As the number of negative pairs is very large, accuracy and specificity have high values (≈ 1). Precision (percentage of pair matches that are correct), recall (percentage of correct pair matches that are detected) and their combination, the F-score, are better measures to use.

3.1 Results

Open World Re-identification: We evaluate on SAIVT, using five images per person per camera. We consider three cameras (Cam 3, 5 and 8, that are challenging according to [1]), where a person that appears in one camera may or may not appear in the others. Table 1 shows results obtained from analyzing every possible pair (within and between cameras). We present variants of our framework using both RankSVM and KISS as the base pairwise models. The CRF results are based on global inference across all three cameras, however columns break down association performance as evaluated within individual cameras (first three), across each pair of cameras (middle three), and across all three cameras ("whole model"). The baseline methods obtain somewhat better recall, due to their non-conservative nature. However on the other hand, the low number of false negatives causes a huge incre-

[1]For KISS, we reduce the dimension of ELF to 100 with PCA, as it is not robust to high dimensional data

Table 1: Re-identification among three cameras from SAIVT. The last column shows the global performance. Other columns show local performance. E.g., C3-C8 shows the quality of the connections between camera 3 and camera 8 when the whole CRF model is computed.

F_1-Score	C3	C5	C8	C3 - C5	C3 - C8	C5 - C8	Whole model
Naive RankSVM	31.7%	34.1%	27.1%	15.9%	20.1%	24.6%	26.2%
Naive KISS	32.6%	29.4%	34.7%	23.4%	31.0%	29.6%	29.5%
RankSVM+CRF	50.1%	41.1%	**73.2%**	18.2%	43.4%	32.4%	42.0%
KISS+CRF	**57.3%**	**52.0%**	70.0%	**30.3%**	**47.6%**	**43.7%**	**48.3%**
Precision	**C3**	**C5**	**C8**	**C3 - C5**	**C3 - C8**	**C5 - C8**	**Whole model**
Naive RankSVM	30,2%	22,2%	36,7%	14,9%	27,7%	25,7%	22,0%
Naive KISS	22.0%	20.0%	22.0%	15.9%	20.7%	19.9%	19.7%
RankSVM+CRF	**63.8%**	**61.4%**	**62.3%**	37.2%	**55.4%**	45.2%	**53.7%**
KISS+CRF	56.4%	59.2%	58.5%	**38.0%**	48.4%	**47.1%**	50.3%
Recall	**C3**	**C5**	**C8**	**C3 - C5**	**C3 - C8**	**C5 - C8**	**Whole model**
Naive RankSVM	50,6%	**87,6%**	44,2%	24,7%	29,4%	43,4%	42,1%
Naive KISS	**70.1%**	63.3%	91.7%	**50.3%**	**70.1%**	**65.4%**	**66.1%**
RankSVM+CRF	47.4%	38.8%	**94.0%**	15.5%	43.1%	30.8%	39.4%
KISS+CRF	62.8%	50.1%	91.1%	28.5%	51.1%	44.7%	49.8%

Table 2: Inferring the number of distinct people in the dataset.

Ground truth	Naive RankSVM	Naive KISS	RankSVM+CRF	KISS+CRF
48	61 ± 17.6	57.8 ± 11.2	65 ± 13.2	$\mathbf{54.1 \pm 7.9}$

ment in the number of false positives, resulting in significantly worse precision. Our CRF model is more robust, as evidenced by its maintenance of high precision values. Moreover, it improves both of the base methods it is paired with. Because of the dichotomy between obtaining high recall and precision, we conclude that the F-Score is the best overall metric to validate an open-world ReID algorithm.

Estimating the number of people: An important general question of interest to camera network operators is how many unique people are observed by the camera network in a given time period? This is implicit in the open world ReID task. Inference in our CRF model computes this as a byproduct[2], so we can answer this question directly. Table 2 shows the estimated number of unique people among the approximately 600 detections across all three cameras. The estimated number of people along with the standard deviation of the estimate over multiple runs are given. In each case our framework improves on the baseline result, with KISS+CRF obtaining the best and most stable estimate.

4 Conclusion

We have proposed the first method to address the most practical 'open world' variant of the re-identification problem. That is, when no information is provided a priori about the number or distribution of people. We develop a two-step CRF model using both appearance, temporal and spatial information that can be solved by fast energy minimization techniques using graph cuts. Evaluation on a challenging public dataset with three cameras demonstrates that the model improves on engineered baselines built on either of two classic pairwise ReID techniques. Moreover, important metadata such as person counts can be generated as a by-product of inference in our model. In our future work, we would like to test our algorithm with more cameras and build explicit person and camera lighting models.

[2]One assumption is made in this point: since we assume we are going to have more than 1 detection per person and per camera, labels with only one associated detection are treated as noise, and removed.

References

[1] Alina Bialkowski, Simon Denman, Patrick Lucey, Sridha Sridharan, and Clinton B Fookes. A database for person re-identification in multi-camera surveillance networks. In *DICTA*, 2012.

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[3] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *ICCV*, 2007.

[4] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010.

[5] Shaogang Gong, Marco Cristani, Chen Change Loy, and Timothy M. Hospedales. The re-identification challenge. In Shaogang Gong, Marco Cristani, Shuicheng Yan, and Chen Change Loy, editors, *Person Re-identification*, pages 1–20. Springer, 2014.

[6] Shaogang Gong, Marco Cristani, Shuicheng Yan, and Chen Change Loy, editors. *Person Re-Identification*. Springer, 2014.

[7] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*. 2008.

[8] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Image Analysis*, pages 91–102. Springer, 2011.

[9] Martin Hirzer, Peter M Roth, Martin Köstinger, and Horst Bischof. Relaxed pairwise learned metric for person re-identification. In *ECCV*. 2012.

[10] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD*, pages 217–226, 2006.

[11] Vijay John, Gwenn Englebienne, and Ben Krose. Solving person re-identification in non-overlapping camera using efficient gibbs sampling. In *BMVC*, 2013.

[12] Svebor Karaman and Andrew D Bagdanov. Identity inference: generalizing person re-identification scenarios. In *ECCV 2012. Workshops and Demonstrations*, 2012.

[13] Martin Kostinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[15] Ryan Layne, Timothy M. Hospedales, and Shaogang Gong. Person re-identification by attributes. In *BMVC*, 2012.

[16] Ryan Layne, Timothy M. Hospedales, and Shaogang Gong. Re-id: Hunting attributes in the wild. In *BMVC*, 2014.

[17] Bryan Prosser, Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by support vector ranking. In *BMVC*, 2010.

[18] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. People re-identification in surveillance and forensics: a survey. *ACM Computing Surveys*, December 2013.

[19] Bo Yang and Ram Nevatia. An online learned crf model for multi-target tracking. In *CVPR*, 2012.

[20] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013.

[21] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Transfer re-identification: From person to set-based verification. In *CVPR*, 2012.

Chapter 3

Behavior Analysis Published Papers

3.1 Journal Paper: On the Use of a Minimal Path Approach for Target Trajectory Analysis

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Marcos Ortega Hortas
Affiliation: Universidade da Coruña, Spain

Co-author #3: Manuel Francisco González Penedo
Affiliation: Universidade da Coruña, Spain

Co-author #4: Jorge Novo Buján
Affiliation: Universidade da Coruña, Spain

Co-author #4: Noelia Barreira Rodríguez
Affiliation: Universidade da Coruña, Spain

Article title: On the Use of a Minimal Path Approach for Target Trajectory Analysis
Journal: Pattern Recognition
Volume: 46(7)
Pages: 2015–2027
Editorial: Elsevier
ISSN: 0031-3203
Year: 2013

Pattern Recognition 46 (2013) 2015–2027

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

On the use of a minimal path approach for target trajectory analysis

B. Cancela *, M. Ortega, M.G. Penedo, J. Novo, N. Barreira

Varpa Group, Department of Computer Science Campus de Elviña s/n, University of A Coruña, Spain

ARTICLE INFO

Article history:
Received 16 April 2012
Received in revised form
2 October 2012
Accepted 9 January 2013
Available online 17 January 2013

Keywords:
Online trajectory analysis
Dynamic potential
Abnormal behavior detection
Minimal path
Geodesic active contours

ABSTRACT

Vision-based action recognition has multiple applications, mainly focused in video surveillance systems. The art of labeling each target behavior in crowded scenarios is a complicated field since usually we do not have visual confirmation of the parts of a target to infer its behavior. Thus, trajectory analysis becomes a good choice to try to infer knowledge about target movements. Most of the contributions to this field involve a training period in which we obtain information a priori about the environment, storing a dataset with all the possible usual routes. Based in the minimal path theory using geodesic active contours, we present a novel architecture where no initial information about the scene is needed, while it is possible to include it if necessary to specify constraints. Experimental results in two different application domains show the performance and flexibility of this method, being able to be used in multiple trajectory analysis problems.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays, target behavior analysis becomes an important field of study when dealing with security environments. The idea of an automatic framework that can deal with abnormal movements is one of the most active research fields. There are multiple solutions to this topic, like detecting abnormal movement over crowded scenes [1,2], monitoring traffic in highways [3,4] or evaluating animal behavior [5].

Despite the fact of the multiple techniques that exist in target behavior analysis, all of them have a problem in common: target tracking. The quality of these techniques largely depends on the tracking algorithm used. Over crowded scenes, these algorithms have to be able to track every object of interest without mistaking them. There are multiple challenges that have to be solved, like total and partial occlusions due to static objects between the target and the camera, or target collisions, which includes grouping and splitting events. The latter occur when some tracking objects are so close that the system can only recognize one target. The system must recover the correct identification of each target after the collision event occurs.

Among the multiple target behavior techniques used, trajectory analysis is a powerful tool to find abnormal behavior into the scene or detect most common trajectories, which is a good information to be used in multiple fields, like obtaining the best place to sell a product or in road analysis to detect possible traffic jams.

Johnson and Hogg [6] proposed a method for learning behavior models using a vector quantization method to learn typical routes taken by pedestrians from representative trajectories. However, no high-level semantic information is derived and their method requires the knowledge of entry/exit areas of the scene, which are defined manually.

There are many different approaches in the literature for trajectory analysis. One of the first approaches uses a vector quantization method to learn typical routes [6]. However, this method requires to manually set the entry areas in the scene. Clustering techniques based on a flow vector are also used to detect abnormal movements [7–9]. Using a classification technique algorithm (*K*-means or likelihood functions), they classify future sequences as familiar or novel. In [10], a self-organizing map also classifies each route as usual or novel converting routes in a fixed trajectory vector. It can also work with partial routes, but is highly dependent of the quality of the training set, which requires a previous evaluation of the scene. The same problem appears in Rao et al. [11], where a probabilistic density model to define each target route is used, while a log-likelihood function is used to classify each event, and in Mecocci et al. [12], where the Altruistic Vector Quantization algorithm is modified to be able to compare different length routes. In both cases no multiple-target tracking is used.

A statistical approach is also used by Porikli et al. [13], using Hidden Markov Models (HMMs) to obtain representations of object features, like speed, size or orientation. Thus, combined with affinity matrices, they apply an eigenvector decomposition to detect usual and unusual behavior. No a priori information is needed, since they detect as usual the high occurrence of events

* Corresponding author. Tel.: +34 981 16 70 00x1330; fax: +34 981 16 71 60.
E-mail addresses: brais.cancela@udc.es (B. Cancela), mortega@udc.es (M. Ortega), mgpenedo@udc.es (M.G. Penedo), jnovo@udc.es (J. Novo), nbarreira@udc.es (N. Barreira).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.01.013

using an unsupervised training set. Calderara et al. [14] use the same type of training set, creating a mixture of Von Misses distributions. The abnormal movement is detected calculating the probability of the trajectory according to the distributions. This probability is easily updated online. Vaswani et al. [15] also use HMMs to encode target routes. Abnormal activity detection is formulated as a change-detection problem. Again, the full trajectory is needed to detect the behavior. A similar idea is used in [16], also introducing the orientation into the system. In this case multiple targets are tracked, but target collisions, like the explained before, are ignored.

The main problem in the algorithms based on trajectory analysis is that they usually need a complete route to compute the results. This issue is critical when dealing with online applications such as surveillance systems. Online analysis is needed in order to operate in those environments. Hu et al. [17] try to solve this problem introducing motion segments. Other techniques need to build a training set to obtain an initial information which is used to compare each new input [6,10,8,9]. However, there could be scenes in which we do not have any information a priori about the environment. Furthermore, a little change in the scene after the training period could result in important changes in target behavior, making the training useless.

In this work we present a new strategy for trajectory analysis under multiple-target scenarios. This strategy is based on the minimal path algorithm using geodesic active contours [18]. One of the advantages of this method is that we do not need to compute and store the usual routes, since storing the routes of each target into a potential image is enough to obtain the best track between the initial and the final target position. This final position could be any frame in which we want to evaluate that target, enabling this methodology for online processing. Also, due to the nature of the minimal path algorithm, orientation is implicitly introduced. This system is able to deal with path changes along time.

A multiple-target tracking algorithm is used in order to obtain each target track. Using a hierarchical structure, including two different trackers, this method can deal with many of the problems explained before, like total occlusions and target collisions. A further explanation of the tracking methodology can be found in [19].

The use of this minimal path technique in order to develop a trajectory analysis becomes a powerful tool that can be used in many applications depending on the potential image we want to use. This potential image enables us to introduce information a priori about the environment or, on the contrary, eliciting the information about every target behavior in the scene using an online equation.

This paper is organized as follows: Section 2 introduces the minimal path method using geodesic active contours, explaining the modifications introduced in our methodology; Section 3 makes a brief introduction about different techniques to obtain the correct potential image associated to the routes; Section 4 discusses two different applications that can be implemented using the methodology described, showing the obtained results; finally, Section 5 offers a discussion about the improvement of this system against classical clustering techniques, and also includes conclusions and future work.

2. Minimal paths

There are many approaches in the literature to solve the minimal path problem. The most used techniques are based in graphs, like Dijkstra, F*, A* and other variants. Some of these algorithms are used in images, like road detection [20] or distance maps computation [21]. The basic idea is to use each image pixel as vertex in a graph. The advantages of these techniques are the speed and the complexity. However, when dealing with images, these algorithms suffer from 'metrication errors' [18]. When considering an image like a graph, where all pixels are nodes with 4 (or 8) connectivity, it is clear to see that the length of a shortest path between two pixel could highly differ from the continuous solution. In fact, if we increase the resolution of the image, the 'metrication error' will not decrease, which is the main problem of using these methods in images. Fig. 1(a) shows an example of this problem when dealing with images. A simple potential is used, allowing the solution to be a simple straight line between starting and ending point. However, graph search algorithms can obtain this desirable solution.

Thus, we need a technique to deal with the continuous problem as far as possible. The 'metrication error' has to decrease while we increase the image resolution. So, our approach is based on the minimal path approach presented by Cohen and Kimmel [18]. This method is based on the idea of finding the minimal path between two points using a potential image. This potential image P takes lower values near the features of interest.

The basic idea is a reformulation of the classical snake equation [22]. This is a 'parametric' equation containing two different forces, one that controls the strength and flexibility of the snake and another force, called potential, which attracts the snake to the features of interest. Later, geometric active contours [23] intrinsically include the snake parametrization into the external potential. However, geometric active contours do not arise from the minimization of an energy, like the classical parametric snakes. To solve this situation, Caselles et al. [24] introduced the geodesic active contours. These contours are based in the Fermat's principle of light propagation, which says that the path taken between two points by a ray of monochrome light is the path that can be crossed in least time. This is the reason why

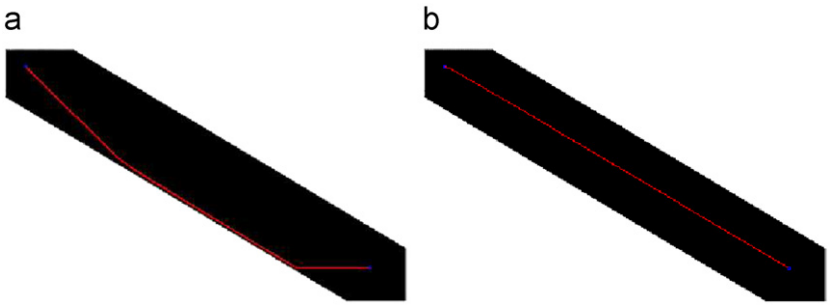


Fig. 1. Metrication error. Using classical graph-search algorithms (a) over images results in a solution that is not consistent with the continuous solution, since the error remains in spite of increasing image resolution. However, this can be solved using the minimal path approach (b) presented by Cohen et al., using a second-order numerical approach for the back-propagation (in this case, Heun's method is used).

this technique is useful to obtain the minimal path between two points.

The aim of the minimal path approach between two points p_0 and p_1 using geodesic active contours is to obtain a curve $C(s)$ that minimizes the functional

$$E(C) = \int_\Omega E_{int}(C(s)) + E_{ext}(C(s))\, ds = \int_\Omega \omega \left\| \frac{\partial C}{\partial s}(s) \right\|^2 + P(C(s))\, ds, \quad (1)$$

where $\Omega = [0,L]$, L is the length of $C(s)$, ω is the regularization term, s is the arc-length parameter and P is the image potential. The internal energy is the partial derivative of the curve with respect to s and controls the regularity in the contour. The external energy term is the potential and represents the desired image features. Since we have defined the end points, the curve is restricted by two boundary conditions: $C(0) = p_0$ and $C(L) = p_1$.

As mentioned before, s is the arc-length parameter, meaning that $\|\partial C/\partial s(s)\| = 1$. Thus, the energy of the model can be written as

$$E(C) = \int_\Omega \omega + P(C(s))\, ds = \int_\Omega \omega \tilde{P}(C(s))\, ds. \quad (2)$$

The regularization term satisfies that $\omega > 0$. This parameter allows us to control the smoothness of the curve. So, assuming that the image potential P has positive values, we conclude that $\tilde{P} > 0$. With this minimization problem, we first search for the surface of minimal action U_0 which starts in $p_0 = C(0)$. This surface is defined as the minimal energy integrated along the starting point p_0 and any given point p

$$U_0(p) = \inf_{A_{p_0,p}} \int_\Omega \omega + P(C(s))\, ds = \inf_{C(L)=p} \left\{ \int_C \tilde{P} ds \right\}, \quad (3)$$

where $A_{p_0,p}$ is the space of all curves that connect p_0 and p over the image potential P. In order to compute U_0, a front propagation is defined, in which we obtain a set of equal energy contours $\mathcal{L}$ in 'time', where t is, in fact the value of the energy. So, in the evolution equation, t represents the height of the level set of U_0

$$\frac{\partial \mathcal{L}(v,t)}{\partial t} = \frac{1}{\tilde{P}} \vec{n}(v,t), \quad (4)$$

where $\vec{n}(v,t)$ is the normal to the closed curve $\mathcal{L}(\cdot,t)$. This equation starts at a small circle centered in p_0 and evolves until reaches all the points in the image. The value of $U_0(p)$ is the time t when the front passes over p.

There are many numerical approaches to compute this minimal action surface. Like Cohen et al., we apply the Sethian Fast Marching Method [25], which is the best solution for real-time systems. This solution is consistent with the continuous propagation rule, which means that the more you refine the grid, the better the solution converges to the true one. Given the potential value $\tilde{P}_{i,j} = \tilde{P}(i\Delta x, j\Delta y)$ in a grid, with $\Delta x = \Delta y = 1$, the method approximates $U_{i,j}$ solving the equation

$$(\max\{u - U_{i-1,j}, u + U_{i+1,j}, 0\})^2 + (\max\{u - U_{i,j-1}, u + U_{i,j+1}, 0\})^2 = \tilde{P}_{i,j}^2, \quad (5)$$

selecting for $U_{i,j}$ the largest u that satisfies the equation. Using this algorithm, when a pixel p is reached by a front propagation started in p_0 and computed, it is possible to obtain the minimal path between these two points without computing the rest of the image, which is a very interesting advantage. This is particularly useful when dealing with only one end point, improving the speed of the algorithm.

The minimal action surface U has a convex like share, which means that U_0 has only one local minimum that is the starting point p_0. So, for any point p in the image, we only need to follow the gradient descendant direction, which always converge at p_0. Thus, using a simple steepest gradient descent algorithm we can find the minimal path. Starting at the final point p, we select the connected pixel with the lowest value U as the next point in the path. This method guarantees the convergence to a solution, but also causes the 'metrication error' mentioned before in the graph search algorithms. So, more complex methods, such as second-order Runge–Kutta, like Heun method, are used to solve this problem, making this approach consistent with the continuous solution. Fig. 1(b) shows an example of the solution of this method. Contrary to the graph-search methods, this is consistent with the continuous solution.

Algorithm 1. Modified Fast Marching method.

Definitions:

- *Alive* set: points of the grid for which U has been computed and it will not be modified.
- *Trial* set: next points in the grid to be examined (4-connectivity) for which a estimation of U is computed using the points in *alive* set.
- *Far* set: the remaining points of the grid for which there is not an estimate for U.

Initialization:

- For each point in the grid, let $U_{i,j} = \infty$ (large positive value). Put all points in the *far* set.
- Set the start point $(i,j) = p_0$ to be zero: $U_{p_0} = 0$, and put it in the *trial* set.

Marching loop:

- Select $p = (i_{min}, j_{min})$ from *trial* with the lowest value of U.
- If p is equal to p_1 being p_1 the final point then we finish.
- Else put p in *alive* and remove it from the *trial* set.
- If $\tilde{P}(i_{min}, j_{min}) < \tau$, for each of the 4 neighboring grid points (k, l) of $(i_{min} j_{min})$:
  - If (k, l) belongs to *far* set, then put (k,l) in *trial* set.
  - If (k, l) is not in *alive* set, then set $U_{k,l}$ with Eq. (5).

In our case, we have to do some modifications to Sethian Fast Marching method. There will be regions in the scene for which no targets passed along time. We have to restrict the minimal path possibilities not to pass for those regions, despite of the regularization parameter ω chosen. To solve this situation, we assign to the pixels in those regions a high value. So, in the Fast Marching algorithm, if we reach a point with a value higher than a chosen threshold τ, we do not allow the propagation over this point. Algorithm 1 shows the necessary steps to make this Modified Fast Marching method. Since reached points are chosen in order, we only need one pass in the image to compute the minimal action surface U. Fig. 2 shows an example of applying this method to obtain the minimal path for a person that walks from the bottom to the top of the image. In this case, having only one end point, according to Cohen et al., we can optimize the Fast Marching Method. Basically, we make a front propagation starting both in the initial and the final point. So, when both fronts reach the same point, we make a back-propagation from that point to both starting points, obtaining the minimal path. This solution explores less points in the image than the usual. We can see the result is close to the real movement of the person.

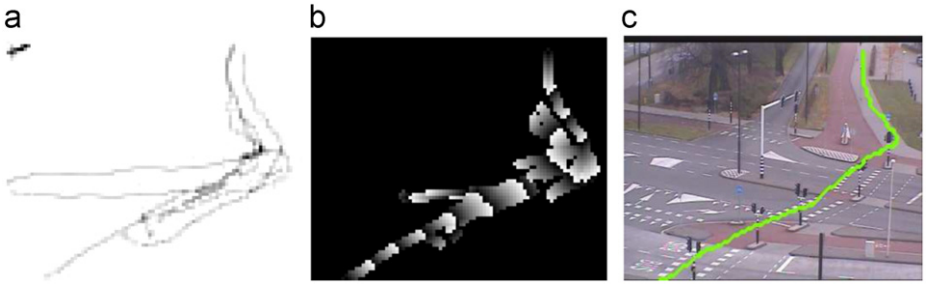


Fig. 2. Restricted Fast Marching method example. (a) Potential image. In white, forbidden pixels. (b) Minimal Action surface U obtained using the Restricted Fast Marching method. (c) Minimal path, as a result of applying the Heun's method in the back propagation over the U surface.

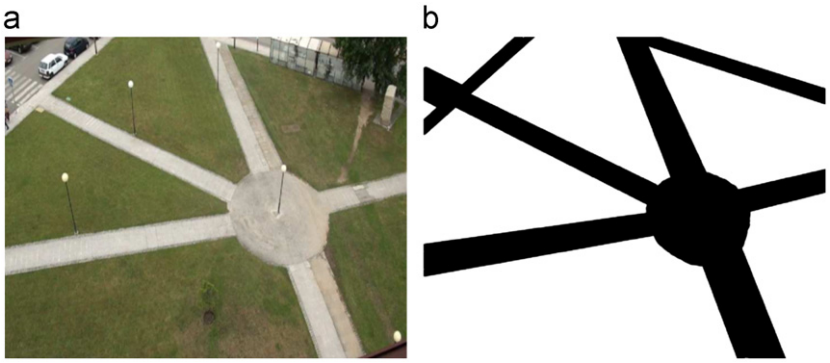


Fig. 3. Fixed potential example. (a) Input image. (b) Potential image. This example allows human movements over the passages, while is not able to locate a minimal path when the person moves through the grass.

3. Discussion on the potential

Dealing with the minimal path approach defined by Cohen et al. one of the crucial points is how to design an ideal potential that can be used to obtain a correct minimal path. In the case of a tracked object, there are a lot of different potentials that can be used depending on the type of knowledge we want to insert into the methodology. Since we are able to track every moving object in the scene using the multiple-target tracking explained in the previous section, it is easy to create a potential image marking the positions for which they have been traveled. So, you do not need any a priori information about the scene, since it is possible to build the potential image online. The potential image provides a huge flexibility to this methodology. We introduce a discussion on different potentials that can be used to obtain the minimal path. Each potential allows us to model different possible scenarios.

3.1. Fixed potential

Although the main purpose to this contribution is to produce a system that can obtain a minimal between two points in a scene without any information from the environment, it is also possible to include information a priori about the scene, like the regions that are not allowed to pass any target, i.e., people walking down a runway or accessing a forbidden area. This type of potential is particularly useful under surveillance scenarios, since it is easy to perform an initial mask with all the regions in the scene that can be reachable for any target. For instance, in Fig. 3 a binary image is used to show which parts of the environment are accessed. Hence, minimal path behavior is restricted to these conditions.

The problem of using this type of potential is the abnormal movement, since it is impossible to obtain a minimal path between any given points not marked as accessible. One solution is to remove the threshold τ condition in Algorithm 1, but the result produce bad estimations trying to stay as much as it possible under pixels marked as allowed. Other solution is to include the target trajectory in the potential with a high value in the points marked as forbidden. With this solution it is possible to obtain a more accurate minimal path, but does not take into account the path of other possible targets with abnormal movement.

3.2. Online potential

If we have no information a priori about the scene, we can use the moving objects to create the potential image. Storing the object positions along the time, we can draw its path around the scene. So, starting in a potential $P=0$, we increase the value over the path positions. Fig. 4 shows an example of this technique. Different persons walking over the sidewalk and cars driving along the road are defining the potential image, in which we clearly see the paths that are most used (darker paths).

As mentioned before, the minimal path algorithm needs a potential image $\tilde{P}>0$, which takes lower values near the features of interest. So, a first idea is to use the potential image

$$\tilde{P}=255-P+\omega,$$

with $P_{i,j}$ the number of objects that travel over the pixel image $I(i,j)$, limited to 255. This equation satisfies the condition explained before. Intuitively, this method will obtain good results, but it can produce an undesirable effect. If the number of objects in the scene is low, P values will also be low. According to [18], given a potential $\tilde{P}>0$, the curvature magnitude $|\kappa|=\|\partial^2 C/\partial s^2\|$, where s is the arclength parameter, is bounded by

$$|\kappa| \leq \sup_{\Omega}\left\{\frac{\|\nabla\tilde{P}\|}{\tilde{P}}\right\}. \tag{6}$$

According to this, and assuming that $\tilde{P}$ values are close to 255 and $\|\nabla\tilde{P}\|\approx 0$, we conclude that $|\kappa|\approx 0$, which means the minimal

Fig. 4. Online potential example. (a, b) For each target detected, its path is added to the potential image (c). In black, most visited pixels. In white, forbidden positions.

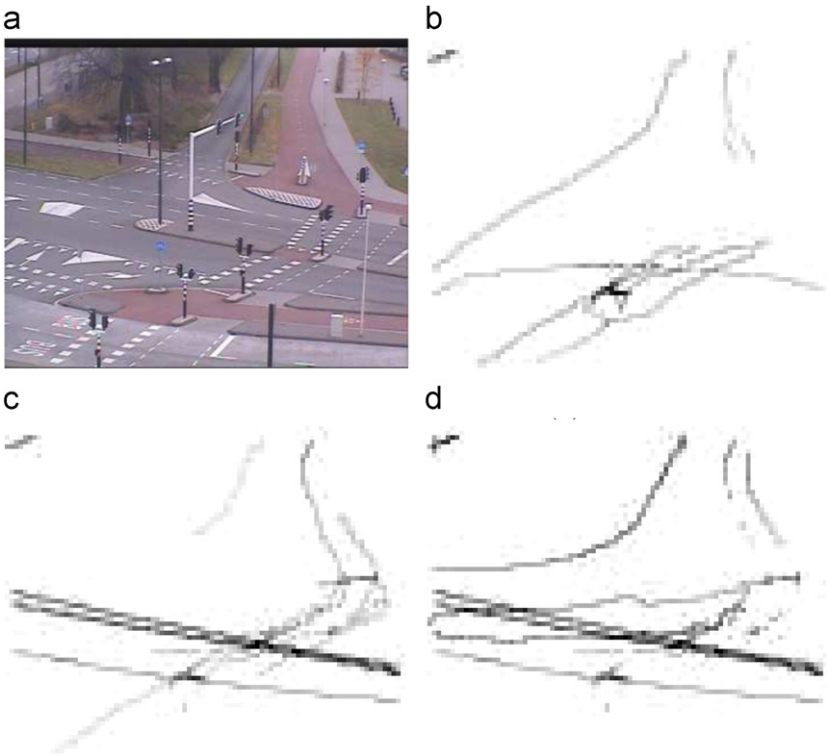


Fig. 5. Dynamic potential example using $\alpha = 0.995$ and $\gamma = 0.5$. Between frames 500 and 1500 we can see some routes that disappears from the potential, like the one that runs from the bottom-left to the top of the potential. (a) Background image. (b) Frame 500. (c) Frame 1000. (d) Frame 1500.

paths are, essentially, straight lines. This is a bad situation because we cannot control the curvature magnitude using the regularization parameter ω, as its value is irrelevant in that scenario. Our solution to this problem is to use the potential

$$\tilde{P} = \sup_D\{P\} - P + \omega, \tag{7}$$

with D an image domain in which we define the potential. Using this equation we can guarantee that

$$\inf_D\left\{\sup_D\{P\} - P\right\} = 0. \tag{8}$$

This condition allows us to establish a new curvature threshold, which is bounded by

$$|\kappa| \leq \sup_\Omega\left\{\frac{\|\nabla\tilde{P}\|}{\omega}\right\}. \tag{9}$$

An explanation of this result can be found in [18]. Thus, the curvature magnitude $|\kappa|$ depends on the regularization parameter ω, which can be set manually depending on the expected behavior. Also, in this case we consider the threshold for Algorithm 1 as $\tau = \sup_D\{\tilde{P}\}$, that are the pixels in the image with value $P = 0$.

3.2.1. *Dynamic potential*

Online potential enhances the most used paths against the others. However, it is possible that targets show an abnormal behavior, in the sense of erratic movements around the scene. Since we do not have any information a priori about the scene, we cannot distinguish if it is an usual movement or not. Thus, without information about the environment, we need a way to isolate the unusual movements and remove them from the potential. Also it is possible that the target routes from a fixed starting and ending point change along time. There are many possibilities that can explain that behavior (an obstacle in the previous trajectory, another route that is not considered before by the other targets, etc.). So it is important to adapt the potential in order to avoid possible bad minimal path estimations.

It is necessary to create a dynamic potential that can deal with these situations. Our approach is the inclusion of an extra parameter α, which decreases P values. Hence, the new potential

at any given time t is

$$\tilde{P}_t = \sup_D\{P_t\} - P_t + \omega, \tag{10}$$

with

$$P_t = \alpha P_{t-1} + T_t, \tag{11}$$

where T_t are the new pixels traveled by all targets between time $\tau = [t-1, t]$ and $0 < \alpha \leq 1$. If $\alpha = 1$ we obtain the potential described in Eq. (7). Note that the potential upper limit can change between iterations, making it able to deal with every training scheme. So, changing the threshold explained in Algorithm 1 to $\tau = \sup_D\{\tilde{P}\} - \gamma$, we can mark as forbidden regions those pixels which values are $P \leq \gamma$. Fig. 5 shows an example of the evolution of this kind of potential. Some trajectories, like the one that crosses from the bottom-left to the top of the image, are used only one time, so, after a few iterations, they are removed from the potential. The value α must be close to one, since it is possible to cause failures in the detection of the minimal path due to the propagation restriction inserted in Algorithm 1.

3.3. Pool of potential images

The problem of detecting minimal paths in multiple-target tracking is that you have multiple different starting and ending points, so the use of the potential images explained before can lead to bad estimation. For example, in Fig. 6(a) and (b), the minimal path between the bottom and the top of the image, which is a route which has been used by a person, is highly influenced by the cars that cross the scene from left to right. This behavior is not desirable since the result obtained is clearly wrong when talking about a human behavior. Our idea is to include a pool of different potentials that are activated depending of some high-level knowledge. Two proposals are discussed next. Again, there are lots of different approaches that can deal with these issues as a result of system flexibility.

3.3.1. Starting position potential pool

The basic idea is to divide the scene into a $N \times N$ grid, creating a pool of N^2 different histograms. Hence, each target uses the potential associated with the node that contains the point in which that target has been tracked for the first time. In practice, targets that activate nodes are in the boundaries, so we consider to use only one potential image associated to all the inner nodes, reducing the number of histograms to $4(N-1)+1$.

Thus, a potential is updated, using Eq. (11), every time a target associated with the corresponding node is moving within the scene. Also the potentials linked with the neighbor nodes are updated. Fig. 7 shows an example of this technique. Each region has its own potential image, which is not influenced by objects which starts in positions far from that part of the scene. Note that this is not a 'trajectory cluster' approach, since we are only restricting the potential effect over different regions. Fig. 6(c) shows how this solution improves the one obtained using only one potential image.

3.3.2. Target classifier pool

Another way to introduce a pool of potentials is to classify each target into different groups in order to improve the minimal path approach. This is so because the expected behavior of a target also depends on their inner condition. For example, the expected behavior for a car is different from a person, since it is not possible to do sudden direction changes.

For example, using a video stored in the CANDELA data set [26], and with a simple ellipse area threshold, we can separate cars from persons or bikes. Each target only updates its potential, having no influence about the minimal paths calculated for the remaining types of targets. Fig. 8 shows an example about the potential obtained using this technique. It is also possible to use more high-level reasoning in order to classify the targets, and, for example, merge the two different pools explained. Fig. 6(d) shows that this method obtains a better accuracy than using only one potential image.

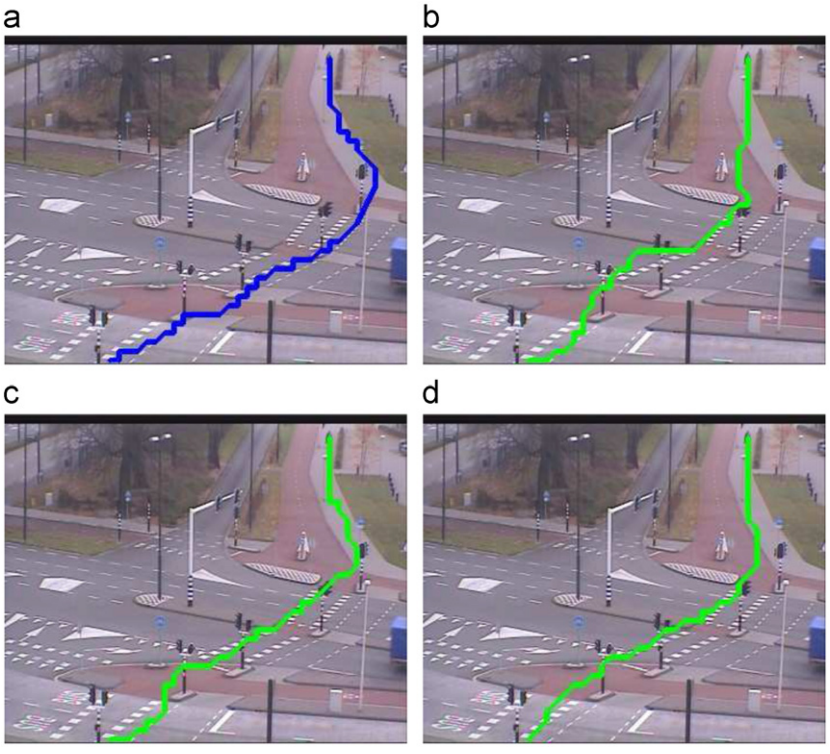


Fig. 6. Minimal path obtained with different potential images. (a) Original path. (b) Single potential image. (c) Starting position potential pool. (d) Target classifier pool. The single potential image obtains a poor result, since it is influenced by all the cars that cross the scene. The potential pool techniques can solve this situation.

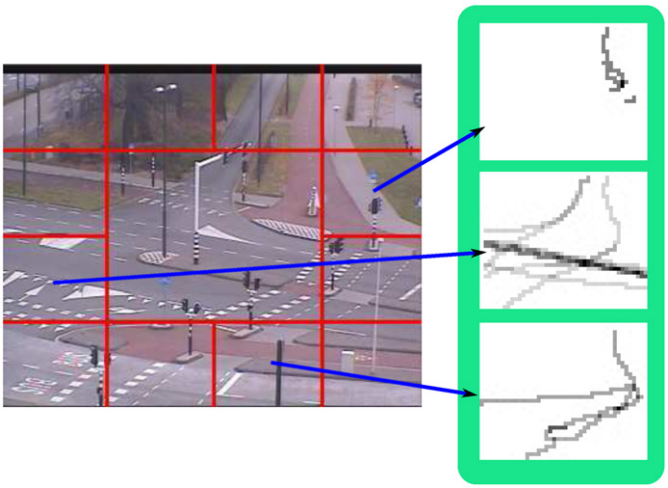


Fig. 7. Starting position potential pool example. The image border is divided into $4(N-1)$ regions, each of one has its own potential image. As we can see in the examples, since we update every potential only if the object tracking starts into its region or in the neighbors, there are potentials which have different values.

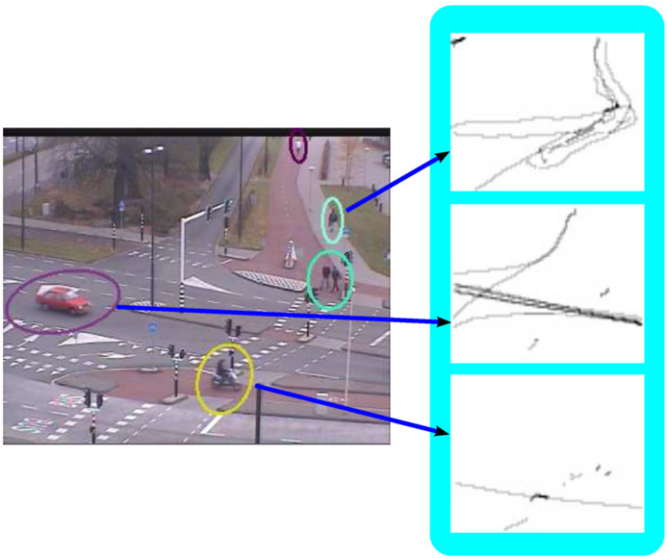


Fig. 8. Target classifier pool example. Using a simple ellipse size threshold, and avoiding ellipse groups, it is possible to separate each target into three different groups: persons, bikes and cars, each with its own potential.

It is difficult to establish which of the pools mentioned before obtains a better result, since its performance could be highly dependent on the problem which has to be solved. Also, other different techniques could be added instead of these potential images that we have described. Contrary to the initial thought, the inclusion of multiple potentials do not have influence in a possible computational cost. The reason is that, even we increase the memory inserting new images, the cost of the algorithm is determined by the Fast Marching Method. Thus, having multiple potential pools each one of them smaller than using only one causes the Marching Method to have less possible accessed pixels, increasing the algorithm speed.

4. Examples and results

When dealing with multiple-target tracking systems we cannot predict all the routes that are going to be used by any target in the scene. Thus, the main purpose of the minimal path algorithms, i.e. search the minimal path between the initial to the final point in order to be used by the target, is not feasible. Instead, we demonstrate this algorithm proves useful in common scenarios of the tracking problem. We have used several real video environments. Depending on the case of use, a different potential will be used among the previously exposed ones. As we mentioned before, the multiple-target tracking system used to test this methodology is explained in [19].

4.1. Online abnormal target movement detection

In the first example, we are interested in detection of abnormal movements. An abnormal movement is a change in the system model whose parameters are unknown, according to [15]. This change could be slow or drastic, and, in case of abnormal trajectories, could be an abnormal behavior with erratic movements, sudden changes in speed or direction or even when an object stops a long time [7]. Our abnormal detection algorithm is based in the idea that, knowing all the possible trajectories within any given scene, a target always tries to use the route that involves less energy demand to do, which is equivalent of finding the minimal path using Algorithm 1 explained before. One of the advantages of our system is that it is able to evaluate the movement at every time we want, having the initial point fixed as the position in which the target is found in the first time, and the final point as the present target position in the moment of computing the minimal path. Several checking can be done during a target's path.

We are focused in two different points in this topic given the information we have about the environment: when we have information a priori and when we have not. Depending on the situation there are different possibilities of classifying each route. When dealing with restricted scenarios, like airports or police stations, it is very helpful to introduce information about the environment. Two different abnormal movements are detected in this case: access to forbidden regions and erratic movements over the granted space. Due to these premises, a combination of two of the potentials mentioned before was used. First, a fixed potential is implemented in order to introduce all the forbidden regions, activating an alert of abnormal movement whenever any tracking object entered in them.

However, having only one fixed potential can introduce problems, since the minimal path into a constant potential is an straight line, which is not necessary the best choice in cases where multiple objects are moving in the scene, such as multiple queues. Having this issue in mind, we have to introduce an online potential, as described in Eq. (7), only applied in the granted points described in the fixed potential.

On the contrary, there are situations where we cannot obtain any information about the environment, having only the behavior of the tracking objects to model all the granted areas. In this case the use of a fixed potential is useless. An online potential, like Eq. (7), has also problems since an abnormal movement that introduces a new route remains in the potential image. Hence, a dynamic potential has to be introduced in order to deal with all the problems mentioned before. Eq. (11) is chosen in this scenario.

Unlike the environments which we have information about them, in this case it is impossible to make any assumption about the inlets, the outlets or the target behavior. Thus, the use of a pool of potentials is needed in order to obtain a better accuracy. Despite the fact we do not know any information about the environment, it is possible to predict the type of all different targets that can appeared in the scene. If that is possible, the target classifier pool explained in Section 3.3.2 is the best choice. Otherwise, the starting position pool explained in Section 3.3.1 is a good solution.

4.1.1. Validation metrics

Multiple choices exist in the literature to compare different tracks. Euclidean distance was used in the first attempts. Having a list of positions along time, each point is compared against another. This method is sensible to shift error [27]. Two similar tracks but shifted are not similar. To solve this situation, other techniques are developed, including Dynamic Time Warping (DTW) [28], the Longest Common Subsequence (LCSS) [29] or the metric performed by Piciarelli and Foresti (PF) [27], which are able to compare vectors with different lengths.

However, these methods fails to detect local abnormal movements. In Fig. 9 we can see some examples. Local abnormal movements cannot be detected by these algorithms, since the results they provide are computed as an average. When a target choose a path, and then it decides to go back, reaching again the same position, the algorithms mentioned before ignore the path used between these two times, causing the system to probably recognize the path as usual. This problem arises because of the idea of comparing two vector with different lengths, which is introduced in these methods. Finally, erratic movements, like a car constantly changing its line, cannot be detected as abnormal too. Thus, we decide to explore other methodologies that could be capable of dealing with these problems. We take advantage of the FMM method to perform new metrics.

Once we obtain the minimal path associated to each target, we need a metric to compare that trajectory against the target route, so we can establish it if it can be considered as the same way. In this case, we also consider two different approaches, one using register techniques and another one using a variant of the Fast Marching Method proposed by Sethian.

Registration techniques are methods commonly used where an image is compared to a template in order to determine their similarity. The main idea is to compare two templates, one is used as reference and another that can be modified using geometric transformations in order to increase the similarity by aligning the templates. In order to find the similarity between two aligned images, we use the normalized cross-correlation

$$\mathcal{R}(x,y)=\frac{\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}, \tag{12}$$

where T is the reference template and I the one which can be modified. Values of $\mathcal{R}(x,y)$ close to 1 indicate the trajectories are similar, while values near to 0 show they are totally different. This metric has been successfully used in similar scenarios where the images to compare contained lines on a scene, in particular, blood vessels [30,31]. In our case, since we compare two templates with the same size, the possible transformations are limited. So, these are restricted into a set of small translations. Also, as we are in a fixed camera scenario, the rotations are limited to small angles. Finally, in order to improve the positive results and also speed up the algorithm, we reduce the size of the image by a correction factor. The acceptance criteria is set using a threshold. Fig. 10(c) and (d) shows an example of the type of images this method use, this is, the target route template and the minimal path template, respectively.

On the other hand, it is possible to modify the Fast Marching Method defined by Sethian to obtain a distance map (DM). The main idea is, having the minimal path between the initial and the last point reached by the target, to create a distance map that calculate de minimum distance from each point to the minimal path. One easy way to do that is changing the initialization of Algorithm 1, setting all minimal path p points with value $U_p=0$ and all their p_n neighbors which are not part of the path with value $U_{p_n}=\tilde{P}(p_n)$, being $\tilde{P}=1$. Again, we apply a downsize of the image resolution. For this case, we downsize if by a factor of 8. Fig. 10(e) and (f) shows the distance map image and how the target route fits in into it, respectively.

Another modification is performed in the DM method in order to obtain another metric. The idea is based in the fact that the minimum distance between a point and a given minimal path is also influenced by the environment. For instance, consider we have a highway, with the two opposite lines separated by a barrier. The minimum distance between two points which are in opposite lines should be greater than the typical euclidean distance. Thus, we introduce the weighted distance map (WDM). The idea is similar to the DM, changing the potential $\tilde{P}=1$ for the potential used to obtain the minimal path. This idea can be used in this system, because we are performing a propagation front to obtain the distance. Unfortunately, in point-based metrics this idea cannot be considered.

4.1.2. Metrics results

In our experiments, we empirically choose the values $\alpha=0.999$ and $\omega=\min(\sup_D\{\nabla P_t\}/10,1)$. The inclusion of a potential

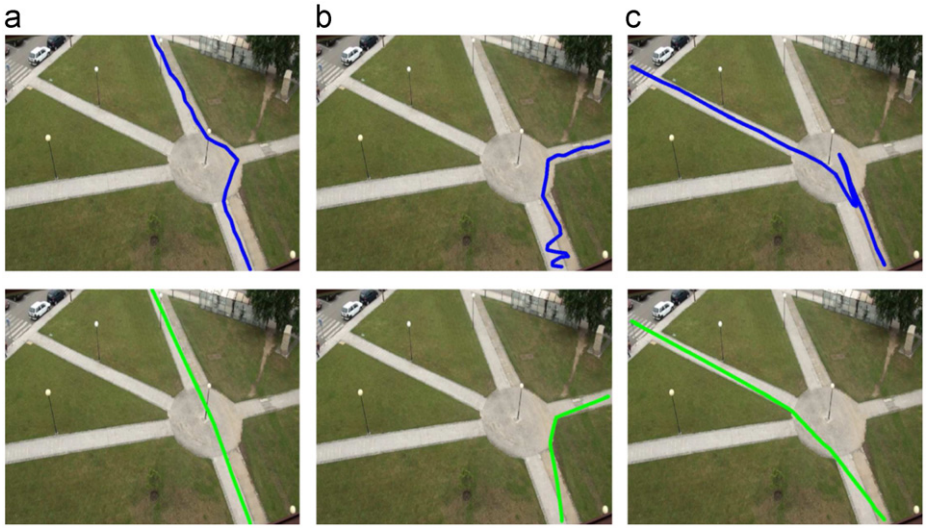


Fig. 9. Abnormal movement examples. The metrics that exist in the literature, like DTW, LCSS or PF, have problems to detect some kinds of abnormal behavior, like (a) abnormal local movements, (b) reaching the same position in two different times and (c), erratic movements along the minimal path. In blue, the real path. In green, the minimal path. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

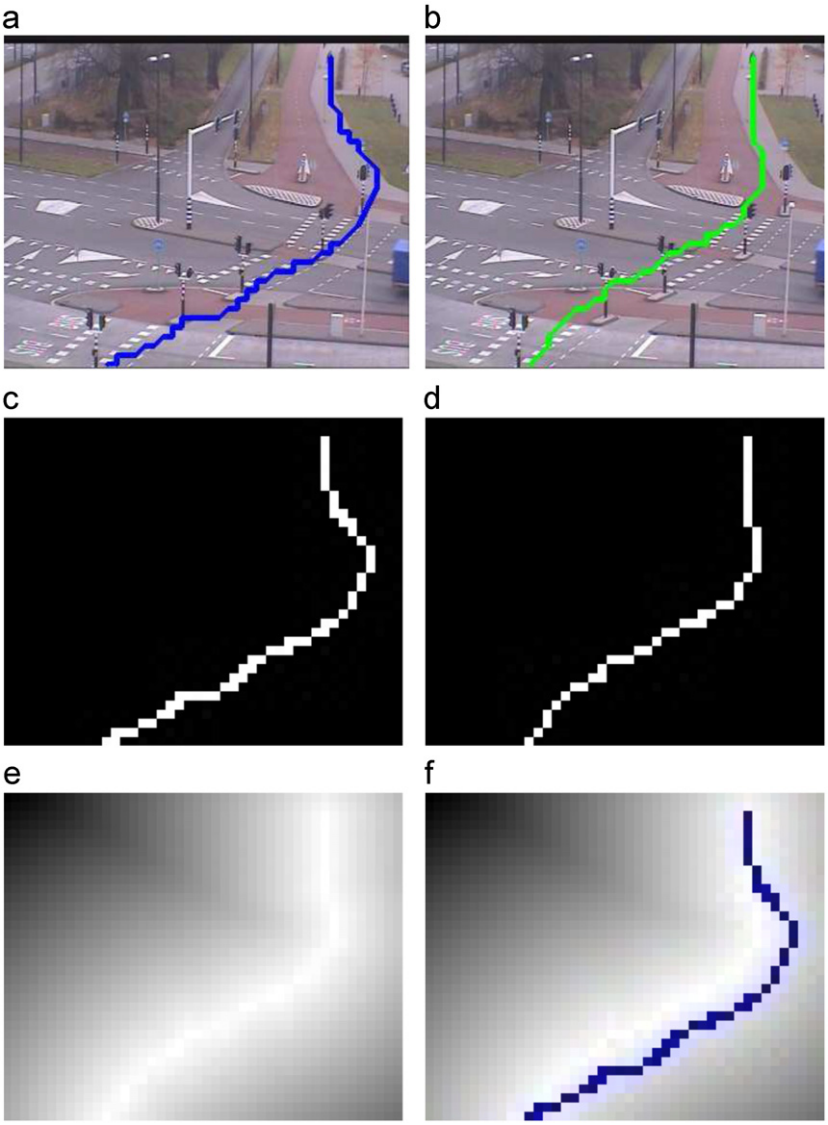


Fig. 10. Matching trajectory methods. (a) Target route. (b) Minimal path route. (c) Target route template for registration technique. (The image size is reduced by 8, in order to speed up the algorithm while the number of positive matches increases.) (d) Minimal path route template for registration technique (the image size is also reduced). (e) Distance map image (in white, values close to the minimal path). (f) Target route fix into distance map image. This route is really close to the minimal path solution.

pool depends on the initial analysis of the environment, i.e., the number of different tracking object classes or the number of inlets and outlets. Is not always necessary to use it in every case. Two different scenarios were tested: a cross-roads included in the CANDELA dataset and another experiment using an own dataset, BARD [32]. This dataset contains human movements over a crossroad. Contrary to CANDELA dataset, usual entry/exit regions are clearly bounded. Usual movements cross the scene along the pavement, while abnormal movements cross the grass. Different videos were used with a duration between 1 and 2 min, resulting in more than 5000 samples.

In the first case, using the BARD dataset, we perform a fixed potential with no updates, since we clearly have five different entry/exit regions. As we know that only persons will appear in this environment, we only use an unique potential image. Fig. 11(c) shows the potential image used, with $\omega = 1$. In the second case. that is, using the CANDELA dataset, a target classifier pool is used, since it is easy to distinguish between persons, cars and bikes, using only a bunch of thresholds that check the target size. An online algorithm is used to update each potential.

In Fig. 12 we can see an example of how the system works in the BARD dataset, using a fixed potential. One person walks over the grass, so the minimal path cannot be computed. The methodology marks this target as abnormal behavior (Fig. 12(b, c)). On the contrary, when a person walks over the pavement, the system is able to obtain the minimal path (Fig. 11(a) and (b).

In Fig. 13 we can see an example of the second scenario, that is, using a target classifier pool of dynamic potentials. In this example, two persons cross the scene by the sidewalk, but, after a few seconds, they start to run over the road to the other side of the image, which is clearly a wrong behavior. Fig. 13(a) and (b) shows the complete trajectory that these persons are following, while Fig. 13 (c, d) show their minimal paths, respectively. As we can see, the paths highly differ. Note that we are using an online potential image, which means that we do not have any information a priori about the scene. Thus, these paths could have been considered as usual movements if no targets updated the potential image before.

At the same time, it is possible the abnormal path could be caused because of a physical obstruction. However, since an

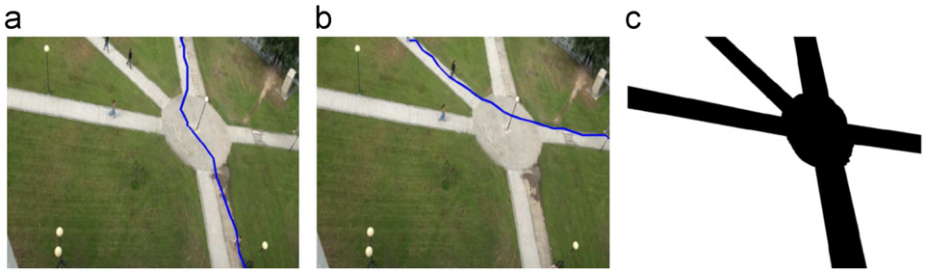


Fig. 11. Fixed Potential examples. Each target performs a route along the predefined ways (a, b). (c) Fixed potential image.

Fig. 12. Abnormal behavior example. Using a fixed potential image, the system detects as abnormal behavior every target that appeared into forbidden regions.

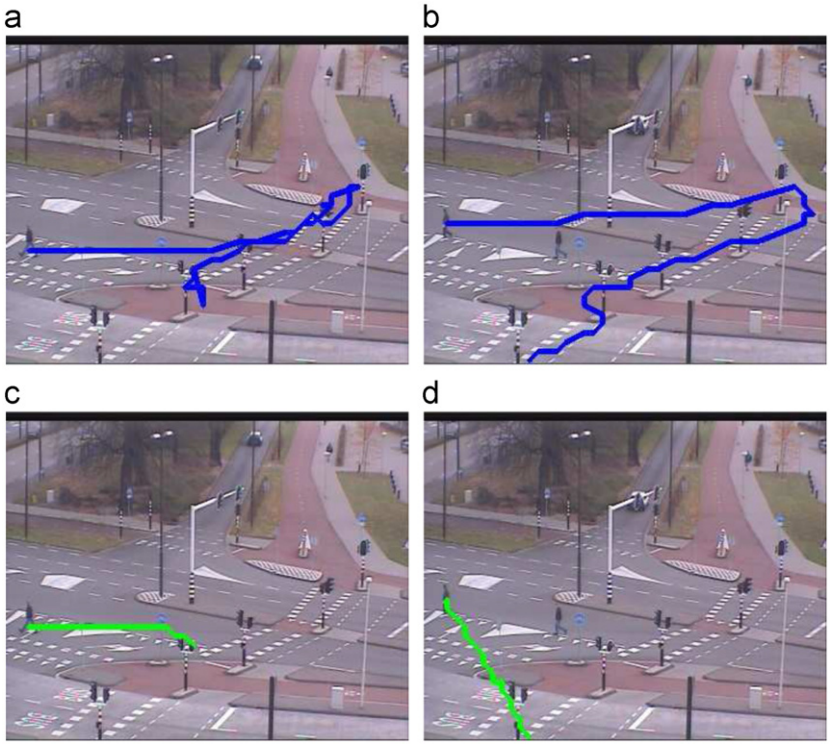


Fig. 13. Abnormal behavior examples. Two different persons make an unusual movement, causing the minimal path between the initial and final position highly differ.

abnormal movement is detected in the first steps, the new path will be marked as usual after a few steps, using a dynamic potential approach. Inaccessible regions due to a possible obstruction will reduce their potential quickly, while the new regions used will increase their potential values.

So, the different trajectory comparison techniques mentioned before were tested using these dataset. To compute the minimal path we reduce the image size by 8. Furthermore, in the case of the cross-correlation technique, we only allow rotations from -10° to 10° and 1-pixel translations. Table 1 shows the false positive (FP) and false negative (FN) ratio depending on the chosen threshold. As we can see, the cross-correlation technique obtains poor results. On the contrary, using the distance map, no matter if we use the mean or the variance, it is possible to clearly establish a threshold. Also, this method needs less computing demand than the cross-correlation technique, which needs to compute multiple translations and rotations. Although the mean and the variance obtain better results, the variance can handle all the abnormal movements showed in Fig. 9, so we decide to choose it as the better metric.

Table 1
Abnormal path detection statistics. Using the distance map we can obtain a very good performance, according to the tests we made. Cross-correlation technique obtains a poor result.

Measures	Threshold	Specificity (%)	Sensibility (%)
Cross-correlation	0.6	86.36	56.69
Mean (distance map)	3.5	100	100
Variance (distance map)	6	100	100

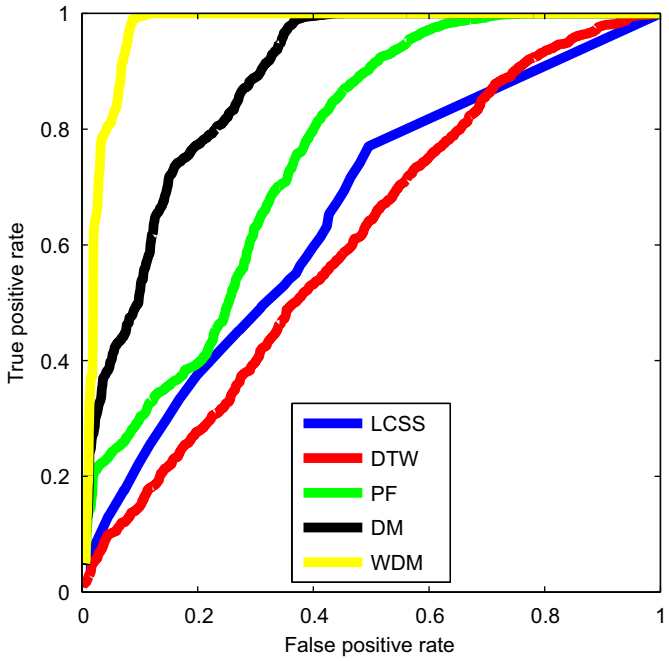


Fig. 14. ROC curve for the pedestrian trajectory analysis. Both DM and WDM outperforms classical vector distance methods.

Finally, we decided to compare the DM and the WDM, using the deviation metric, against three different vector distance metrics: DTW, LCSS and PF. We decide to use these three methods because of the results obtained by Morris and Trivedi [33], which show these three methods obtain the better performance. We use an own dataset containing 610 different routes, where 50 of them are abnormal movements. Each route has stored 27 positions, resulting in 15 860 path comparisons.

Both DTW, LCSS and PF were calibrated, tuning their parameters in order to obtain the best performance in this dataset. The idea is to see how the methods are capable of detecting abnormal movements. We use a fixed potential to obtain the minimal paths, which is showed in Fig. 3(b). This potential is also used in the WDM measure. Fig. 14 shows the obtained results. Clearly, DM and WDM outperform the classical vector distance methods, since they are capable of dealing with local abnormal movements. The reason WDM obtains the better results is because the value obtained highly increases when a target appeared into a forbidden region. In Fig. 15 we show some abnormal movements that can only be detected by the WDM method.

4.2. Main trajectories detection

As mentioned before, sometimes we cannot infer any information about the environment a priori. So, it is also interesting to obtain an approach about the main trajectories used in the scene. This is an useful information in multiple fields of action, i.e., improving a traffic light timing, since we know the usual behavior of all the objects involved, or to improve the product placements into a market checking the main routes used by the customers, which enables us to put any product we want to sell in a strategical situation.

So, we propose a second experiment with the idea to show the versatility of the presented methodology in the situation of determining dynamic trajectories without any information a priori. To do that, we use the CANDELA dataset, which contains a camera position far enough to obtain a good number of different trajectories. For this experiment we focus in cars. Although it is not the scope of this paper, automatic discrimination between cars and humans is achieved by a simple area threshold over the target ellipse representations.

The idea consists on determining all the trajectories associated to the cars within the scene, showing the importance of each one based in the frequency of mobile objects on it. This method updates the potential image iteratively every frame. We need all the information about the routes involved in the scene. Thus, we use the online potential explained in Eq. (7) instead of dynamic potential.

When a target is detected for the first time, its position is stored. We do not need to store all the target route, since we are only focused in their initial and final point. We consider a trajectory when the target tracking stops. Other trajectories can be considered, such as when the target stops or when a sudden orientation change occurs, for example. This is only one example of the multiple solutions this methodology can provide.

Once we have the trajectory defined, we check the trajectory pool to see if there exists any route created before that have similar boundary points. Is we find it, we increase the weight of that route. We are only focused into the beginning and into the end, using the euclidean distance to perform the matching. If we cannot find a similar route, we create another one.

After the online potential is updated, the main trajectories are obtained by performing the minimal path approach over all the trajectories included into the trajectory pool mentioned before. Each trajectory has its own weight, according to the number of targets transiting it.

In Fig. 16 we can see an example of the application of this technique. Initially, we suppose not to have any information a priori about the environment. So, we use a target classifier pool that isolates cars into one potential. In this case, we only divide the image size by 4, in order to obtain a best path accuracy. Finally, we obtain the minimal paths showed in the image. Also it is easy to indicate the percentage of use of each trajectory, as we can see in the figure. Note that the path that crosses the scene from the left to the right is the most used. Additionally, when two different tracks are merged at some segment, their weights are added, increasing its percentage of use, as we can see into the right part of the image.

5. Discussion and conclusions

In this paper, we describe a novel methodology for trajectory analysis based on the minimal path theory using geodesic active contours, which is more accurate than classical graph search techniques when dealing with images. This technique allows to determine the ‘action’ or ‘potential’ image that is going to be used to find the minimal path, which is often hard to obtain. Since we obtain the route of each target that appears in the image, it is easy to create that potential image.

Different potential images were introduced in order to find the best accuracy in the system depending on the approach we want to solve. No information a priori is needed to run this system, although it is also possible to include it in order to enrich the environment by adding some constraints. Applying the proposed methodology to different problems using one public dataset,

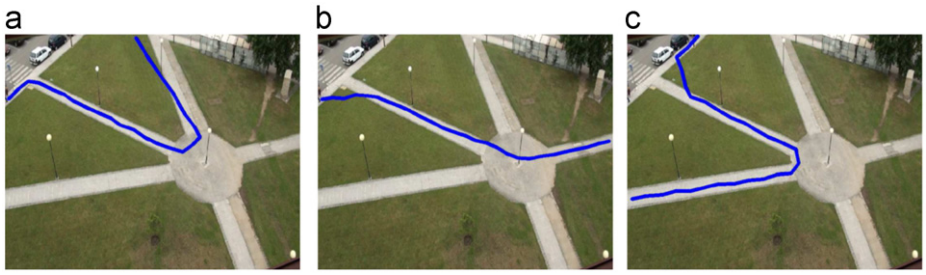


Fig. 15. Using a fixed potential image, we define as abnormal behavior every target that appeared into forbidden regions, even if it is for a short space of time. Only WDM method can detect these abnormal movements.

Fig. 16. Car main routes. Using a simple online potential which is only updated with the cars in the scene we can obtain the trajectories used, which clearly fit with the roads in the scene. The percentage of use of each route is also marked.

which is referenced in the literature, and our own dataset, which is included in order to increase the number of different situations that can happen. Two different experiments are proposed to show the flexibility of this methodology. A system for detecting abnormal behavior is presented, including some quantitative results which do no exists in the literature. A system for obtain the main trajectories into a scene with no information a priori is also presented.

Previous approaches to trajectory analysis need to store a set of different clusters, each one having information of an unique path. Multiple clustering techniques can be used, being the most recent used direct [34], divisive [35], graph based [36] and spectral [37]. In Table 2 we illustrate the problems these methods have. First, you always need a set of clusters, so a previous training is needed. This is a problem when the target behavior changes along time, making the training useless. On the contrary, our method can work both with and without any information a priori about the environment.

Since a training procedure is needed, you need to merge different similar paths to perform a cluster. Although, as we see before, there exists metrics that can work with vectors with different sizes, a fixed length is needed in order to store the cluster. This is a huge problem, since a small vector length could cause the system to lose information. For instance, if we have a car which is constantly changing between different lines, making a cosine-like trajectory, we can lose a huge part of the information if the points stored always coincides with the same road, turning an abnormal movement into an usual one. Furthermore, a fixed length causes the system to have problems with partial trajectories. If we lose a target in the middle of its trajectory, and it is never recovered, causes the path to be useless to incorporate in its

Table 2
Comparison between clustering techniques against minimal path technique.

Properties	Clustering methods	Minimal path method
Information a priori	Needed	No needed
Path length	Fixed	Variable
Memory requirement	Incremental	Constant
Online updating	Restricted	Yes
Environment information	Explicit	Implicit
Occlusion impact	High	Low
Vector distance metrics	Yes	Yes
Distance map techniques	No	Yes

respective cluster, making the system fair to deal with online updating. We have to discard the path or to create a new cluster, causing the system memory to highly increases. Storing all this information in a simple potential image, we can solve all these problems, while keeping stable the memory requirements.

Finally, the clustering techniques need a vector distance measure to work. On the contrary, our method can use both that techniques and distance maps also. This helps the system to be more accurate with a continuous solution, that means, the rotation of the scene does not interferes in the solution. This makes the system more robust than clustering techniques. Furthermore, as we can see in the WDM, information about the environment can be implicitly included within the distance function, which allows the abnormal movement detection procedure to be more accurate.

We obtain promising results about its performance, showing the flexibility of this methodology for being used in many different applications. For instance, a fixed potential could be used in surveillance scenarios, since we have information a priori about the forbidden regions. On the contrary, an online potential is a good choice when we want to find the usual trajectories along the scene.

In a future research a minimal path system which uses an oriented potential image would be interesting to add, to avoid undesired scenarios, mainly focused in vehicle situations. For example, using this system over a car that make a U-turn in a roundabout will obtain a bad estimation, since the minimal path without orientation does not need to circle the roundabout, despite the fact the trajectory obtained is an illegal path according to circulation laws. Another problem concerns to different level crosses, that is, a bridge over a road. We obtain the same bad estimation explained before.

Moreover, despite the fact the speed is computed and stored using a multiple-target tracking framework, it was not taken into account for this study. A study that relate the speed with the regularization term ω would be interesting. Also an analysis about the stopped target would be useful in behavior detection.

Conflict of interest statement

There is no conflict of interest.

Acknowledgments

This paper has been partly funded by the Consellería de Industria, Xunta de Galicia through Grant contracts 10/CSA918054PR and 10TIC009CT.

References

[1] X. Wu, Y. Ou, H. Qian, Y. Xu, A detection system for human abnormal behavior, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005), 2005, pp. 1204–1208.
[2] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, 2009, pp. 935–942.
[3] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russell, Towards robust automatic traffic scene analysis in real-time, in: Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994. Conference A: Computer Vision Image Processing, vol. 1, 1994, pp. 126–131.
[4] J.M. Gryn, R.P. Wildes, J.K. Tsotsos, Detecting motion patterns via direction maps with application to surveillance, Computer Vision and Image Understanding 113 (2) (2009) 291–307.
[5] J. Fan, N. Jiang, Y. Wu, Automatic video-based analysis of animal behaviors, in: Seventeenth IEEE International Conference on Image Processing (ICIP), 2010, pp. 1513–1516.
[6] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Image and Vision Computing 14 (8) (1996) 609–615.
[7] W. Grimson, L. Lee, R. Romano, C. Stauffer, Using adaptive tracking to classify and monitor activities in a site, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998, pp. 22–31.
[8] I. Junejo, O. Javed, M. Shah, Multi feature path modeling for video surveillance, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 2, 2004, pp. 716–719.
[9] D. Makris, T. Ellis, Learning semantic scene models from observing activity in visual surveillance, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35 (3) (2005) 397–408.
[10] J. Owens, A. Hunter, Application of the self-organising map to trajectory classification, in: Proceedings of the Third IEEE International Workshop on Visual Surveillance, 2000, pp. 77–83.
[11] S. Rao, P. Sastry, Abnormal activity detection in video sequences using learnt probability densities, in: TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region, vol. 1, 2003, pp. 369–372.
[12] A. Mecocci, M. Pannozzo, A completely autonomous system that learns anomalous movements in advanced video surveillance applications, in: IEEE International Conference on Image Processing, 2005. ICIP 2005, vol. 2, 2005, pp. II-586–589.
[13] F. Porikli, T. Haga, Event detection by eigenvector decomposition using object and frame features, in: Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04, 2004, p. 114.
[14] S. Calderara, R. Cucchiara, A. Prati, Detection of abnormal behaviors using a mixture of von mises distributions, in: IEEE Conference on Advanced Video and Signal Based Surveillance. AVSS 2007, 2007, pp. 141–146.
[15] N. Vaswani, A.R. Chowdhury, R. Chellappa, "Shape activity": a continuous state hmm for moving/deforming shapes with application to abnormal activity detection, IEEE Transactions on Image Processing 14 (10) (2005) 1603–1616.
[16] F. Jiang, Y. Wu, A. Katsaggelos, Abnormal event detection from surveillance video by dynamic hierarchical clustering, in: IEEE International Conference on Image Processing, 2007. ICIP 2007, vol. 5, 2007, pp. V-145–V-148.
[17] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, S. Maybank, A system for learning statistical motion patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1450–1464.
[18] L.D. Cohen, R. Kimmel, Global minimum for active contour models: a minimal path approach, International Journal of Computer Vision 24 (1997) 57–78.
[19] B. Cancela, M. Ortega, A. Fernández , Manuel G. Penedo, Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios, Expert Systems with Applications 40 (4) (2013) 1116–1131.
[20] M. Fischler, J. Tenenbaum, H. Wolf, Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique, Computer Graphics and Image Processing 15 (1981) 201–223.
[21] G. Borgefors, Distance transformations in arbitrary dimensions, Computer Vision, Graphics, and Image Processing 27 (1984) 321–345.
[22] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, International Journal Of Computer Vision 1 (4) (1988) 321–331.
[23] V. Caselles, F. Catt'e, T. Coll, A geometric model for active contours in image processing, Numerische Mathematik 66 (1993) 1–31.
[24] V. Caselles, R. Kimmel, G. Sapiro, Geodesic active contours, International Journal of Computer Vision 22 (1997) 61–79.
[25] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proceedings of the National Academy of Sciences of the United States of America 93 (4) (1996) 1591–1595.
[26] CANDELA, content analysis and networked delivery architectures, http://www.multitel.be/∼ va/candela/.
[27] C. Piciarelli, G.L. Foresti, On-line trajectory clustering for anomalous events detection, Pattern Recognition Letters (2006) 1835–1842.
[28] S. Salvadore, P. Chan, Fastdtw: toward accurate dynamic time warping in linear time and space, in: KDD Workshop on Mining Temporal and Sequential Data, 2004, pp. 70–80.
[29] D. Buzan, S. Sclaroff, G. Kollios, Extraction and clustering of motion trajectories in video, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 2, 2004, pp. 521–524.
[30] C. Mariño, M.G. Penedo, M. Penas, M.J. Carreira, F. González, Personal authentication using digital retinal images, Pattern Analysis and Applications 9 (2006) 21–33.
[31] C. Mariño, M. Ortega, N. Barreira, M.G. Penedo, M.J. Carreira, F. González, Algorithm for registration of full scanning laser ophtalmoscope video sequences, Computer Methods and Programs in Biomedicine 102 (1) (2011) 1–16.
[32] BARD, behavioral analysis and recognition dataset, 〈http://www.varpa.org/bard/〉.
[33] B. Morris, M.M. Trivedi, Learning trajectory patterns by clustering: experimental studies and comparative evaluation, in: CVPR, IEEE, 2009, pp. 312–319.
[34] B. Morris, M.M. Trivedi, An adaptive scene description for activity analysis in surveillance video, in: ICPR, IEEE, 2008, pp. 1–4.
[35] D. Biliotti, G. Antonini, J.P. Thiran, Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences, in: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05), vol. 02, WACV-MOTION '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 50–57.
[36] X. Li, W. Hu, W. Hu, A coarse-to-fine strategy for vehicle motion trajectory clustering, in: Eighteenth International Conference on Pattern Recognition, 2006. ICPR 2006, vol. 1, 2006, pp. 591–594.
[37] W. Hu, D. Xie, Z. Fu, W. Zeng, S. Maybank, Semantic-based surveillance video retrieval, IEEE Transactions on Imgage Processing 16 (4) (2007) 1168–1181.

Brais Cancela Barizo received his M.Sc. degree in Computer Science in 2009. He also worked on vessel tracking in retinal images due to disease analysis. He currently serves as a Ph.D. student in the University of A Coruña. His research areas of interest are medical image analysis, computer vision, biometrics and human behaviour analysis.

Marcos Ortega received his M.Sc. degree in Computer Science in 2004 and his Ph.D. degree cum Laude in 2009. He also worked on face biometrics studying the face evolution due to ageing effects as a visiting researcher in the University of Sassari and methods for age estimation under different facial expression conditions as a visiting postdoctoral fellow in the University of Amsterdam. He is currently an Associate Professor in the Department of Computer Science at the University of A Coruña. His research areas of interest are medical image analysis, computer vision, biometrics and human behaviour analysis.

Manuel G. Penedo received the B.Sc. degree and the Ph.D. degree in Physics from the University of Santiago de Compostela, Spain, in 1990 and 1997, respectively. From 1991 to 1999 he was an Associate Professor of the Computer Science Department in the University of A Coruña, Spain. From 1999 he was a Lecturer at the same department. His current research interests include computer vision, pattern recognition, biomedical image processing and perceptual grouping.

Jorge Novo received the M.Sc. degree and the Ph.D. cum Laude degree in Computer Science from the University of A Coruña in 2007 and 2012, respectively. He also worked, as a visiting researcher, with CMR images in the detection of landmark points at the Imperial College London. His main research interests lie in the fields of computer vision, pattern recognition and biomedical image processing.

Noelia Barreira received Master's and Ph.D. degrees in Computer Science in 2003 and 2009 from the University of A Coruña, Spain. She is currently an Associate Professor in the Department of Computer Science at the University of A Coruña. Her research interests lie in the fields of computer vision, pattern recognition, and biomedical image processing.

3.2 Conference Paper: Path Analysis Using Directional Forces. A Practical Case: Traffic Scenes

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Marcos Ortega Hortas
Affiliation: Universidade da Coruña, Spain

Co-author #3: Manuel Francisco González Penedo
Affiliation: Universidade da Coruña, Spain

Article title: Path Analysis Using Directional Forces. A Practical Case: Traffic Scenes
Conference: 6th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)
Location: Madeira, Portugal
Date: June, 2013

Path Analysis Using Directional Forces. A Practical Case: Traffic Scenes

Brais Cancela, Marcos Ortega, and Manuel G. Penedo

VARPA Group, University of A Coruña
Campus de Elviña s/n, A Coruña, Spain
brais.cancela@udc.es
http://www.varpa.org

Abstract. This paper presents a new solution for path analysis using minimal path techniques with external directional forces. Previously techniques presented in the literature need to store every different path that exists in the scene. This is a problem in terms of memory. They also need the complete route to perform the computation, being unable to be used detecting uncommon events, like accidents, in real time. We introduce a path planning technique that, using only a velocity field, is able to cope with these problems. The technique can be used with no information a priori about the environment, while it is possible to include or even modified it. A case of study based on traffic analysis is presented to show the performance of the methodology. A complex turnaround scene along with highway real data tested our methodology, showing promising results.

Keywords: traffic analysis, minimal path, oriented upwind scheme.

1 Introduction

Path analysis is an important field of study, specially dealing with security environments. A framework that can help human users to detect and prevent from dangerous situations could highly increase human efficiency. Depending on the scene in which we want to use it, i.e. crowded scenes or traffic monitoring, there exists different solutions in the literature to solve this topic.

The use of path analysis is a powerful tool to be used in behavioral systems, such as detecting abnormal behavior in targets with erratic movements, road analysis detecting traffic jams and possible escape routes, or simply detecting zones of interest, where people tends to stay longer than usual.

Several approaches exist in the literature to deal with this topic. In early attempts, Johnson and Hogg [6] modeled a system that learns typical routes taken by pedestrians using a vector quantization method. Unfortunately, this method needs to define a priori the people entrances and exits in the scene. Abnormal movements were also detected using flow vector based clustering techniques [7][9]. Each route is classified as familiar or novel using a classification algorithm (K-means or likelihood functions).

Porikli et al. [12] use Hidden Markov Models (HMM) to obtain representations of object features, like size, orientation or speed. Combining the features with affine matrices, an eigenvector decomposition is applied in order to detect usual and unusual

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 366–373, 2013.
© Springer-Verlag Berlin Heidelberg 2013

behavior. The system uses an unsupervised training set to define the usual routes. Combining this idea with a mixture of Von Misses distribution, Calderara et al. [2] detect abnormal movements calculating the probability of the trajectory according to the distributions.

These first approaches use a training set that can infer initial information about the environment [6][7][9]. The problem in this systems arises when a change in the environment occurs, i.e. a car accident, forcing the targets to change their usual behavior, making the target useless. Thus, an algorithm that is able to update the usual routes online is a good feature to incorporate.

A general issue with all these previous path analysis algorithms is that a complete path to perform the comparison is needed. In online security systems this issue is critical, since the system would not foresee a possible dangerous situation until it has happened. So, an algorithm that can detect abnormal situations at the moment they are happening is also interesting to include. The main idea consist in being able to evaluate each path at every moment.

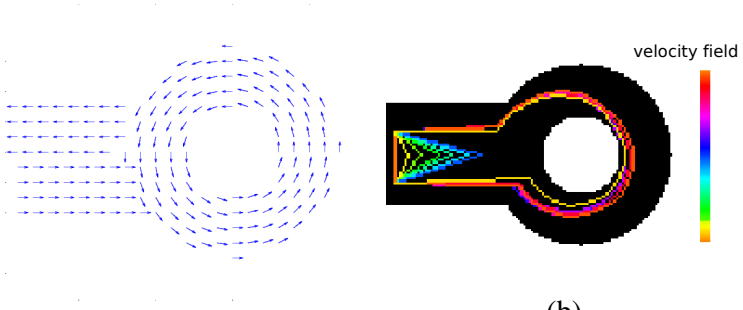
In this work we present a new strategy for path analysis that can solve many of the problems explained before. This approach is based in the idea of that a target tends to choose the path that takes less time to reach its goal, avoiding unnecessary huge deviations. Thus, this system can be modeled as a minimal path approach [4]. In our case, directional forces are also included, in the sense of adding speed information in the model. In the methodology, no a priori information about the environment is needed. Only a potential image and a velocity field is needed to compute the usual routes. The algorithm exposed here can compute a minimal path between an initial and a final point. The initial position is the first time the target is tracked, while the final position could be its position at any frame, enabling it to detect unusual movement at the moment they occur.

This paper is organized as follows: section 2 introduces the minimal path algorithm we are going to use; section 3 introduces the domain in which we are going to test the methodology, describing how to obtain both the potential image and the velocity field, and showing some results; section 4 shows a discussion about the suitability of the method; finally, section 5 offers conclusions and future work.

2 Path Estimation Techniques

As mentioned before, the targets tend to reach their desired goal using the minimum amount of time required. Of course, this trajectory has to take into account the scene constraints. For instance, a car must keep their path within the asphalt, or a boat, which always has to stay over the water. This constraint does not limit our system, since it can be used without it, but the solutions provided with this feature are more accurate.

So, having this in mind, we can conclude that we are dealing with a minimum path problem. There are many approaches in path planning that deals with this issue. In discrete spaces, methods like A*, F* or Dijkstra-like algorithms [15] are used to obtain the better route. However, these methods suffer from 'metric error', that is, their solution is not consistent with the continuous case. To address this problem, Cohen et al. [4] uses geodesic active contours, finding a path of minimal length in a Riemannian metric, being consistent with the continuous case.



(a) (b)

Fig. 1. Minimal paths over a roundabout. (a) Velocity orientation. (b) Oriented Upwind Method [5] with different velocity intensity.

2.1 Ordered Upwind Method

The Ordered Upwind Method (OUM) [14] was created in order to include that directional forces into the path planning problem. However, the computational complexity is increased from $O(NlogN)$ in the isotropic case (FMM), to $O(\Upsilon NlogN)$ in the OUM approach, being Υ the relation between the upper and the lower values on the directional forces.

We have to build a surface of minimal action U. Four sets of point are considered, *far*, *trial*, *alive* and *Accepted Front*. The latter is defined as the set of *alive* points that are adjacent to at least one member of the set of *trial* points. Let AF the set of line segments x_jx_k where x_j and x_k are points included in the *Accepted Front* set, and exists a *trial* point adjacent to both points. Additionally, for each *trial* point x_i, its near front NF is defined as

$$NF(x_i) = \left\{ (x_j, x_k) \in AF \middle| \exists x \; on \; (x_j, x_k) \; s.t. \; \|x - x_i\| \leq h\frac{F_2}{F_1} \right\}, \tag{1}$$

where h is the grid size, and F_1 and F_2 are the lower and upper bounds of the external forces. As in the FMM case, we initialize all the grid points labeled as *far* and $U = \infty$. In the starting point p, it is labeled as *trial* and $V_p = 0$, where V_p represents a tentative value of $U(x_i)$, and let $V_{x_j,x_k}(x_i)$ be a consistent upwinding approximation for U from a virtual simplex $x_jx_ix_k$. So, the tentative value V_{x_i} is defined as $V(x_i) = \min_{(x_j,x_k)\in NF(x_i)} V_{x_j,x_k}(x_i)$. To compute de V_{x_i} value we have to define how to calculate the $V_{x_j,x_k}(x_i)$ value.

3 Case of Study: Traffic Analysis

The technique mentioned in the previous section is general, since it can be used in several fields of study. In our case, we are going to adapt the algorithm into a traffic analysis domain. As mentioned before, we take advantage of the fact that a vehicle tries to use the path that cost less time to reach the desired goal, avoiding unnecessary deviations. To compute de temptative value $V_{x_j,x_k}(x_i)$ in the OUM algorithm, we decided to modify its computation of in order to be more accurate. Our solution is:

$$V_{x_j,x_k}(x_i) = \min_{\zeta\in[0,1]} (\tau(\tilde{x}, x_i) + \zeta U(x_j) + (1-\zeta)U(x_k)), \tag{2}$$

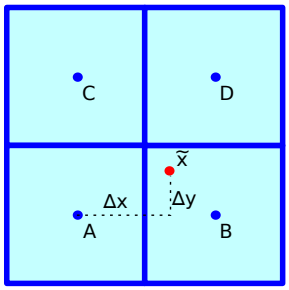


Fig. 2. Bilinear interpolation example in a discretized quadratic grid

where $\tilde{x} = \zeta x_j + (1-\zeta)x_k$ and $\tau(\tilde{x}, x_i)$ is the value of the penalty function when moving from $\tilde{x}$ to x_i:

$$\tau_{ij} = \frac{\sqrt{\Delta} - (W_x \cdot d_x + W_y \cdot d_y)}{V_a^2 - V_w^2}, \tag{3}$$

being $\Delta = V_a^2 \cdot (d_x^2 + d_y^2) - (W_x \cdot d_y - W_y \cdot d_x)^2$, (d_x, d_y) are the space difference between the position x_i and x_j, V_a the target velocity, and V_w, W_x and W_y the field speed and its components. To compute the velocity field in the position $\tilde{x}$ we perform a bilinear interpolation:

$$\begin{aligned} V_w(\tilde{x}) =& (1-\Delta_y)(1-\Delta_x)V_w(A) + \Delta_y(1-\Delta_x)V_w(D) \\ &+ \Delta_y\Delta_x V_w(C) + (1-\Delta_y)\Delta_x V_w(B), \end{aligned} \tag{4}$$

as it can be seen in Fig. 2. The solution of the Eq. 2 is done by using the "golden section search". In Fig. 1-(b) we can see this algorithm provides a smoother result.

3.1 Velocity Field

First, we have to define the velocity field V_w that is going to be included in the algorithm. We are going to perform a microscopic way of the targets, that is, every target is going to be detected and processed separately. This technique is more precise than macroscopic approaches that evaluate the density flow [10], but requires more computational time.

A multiple-target tracking approach is needed. The performing of this technique is not the main goal of this paper. In our experiments, a simple optic-flow method is used [1], but any other more complex technique could be needed in more complex scenarios. The algorithms explained before only need the position of the target. Shape is not considered. In our case, the blob centroid that encapsulates every target is used to store its location.

Two different structures are used to create the velocity field. First, a counter image T is used. The path positions reached by a target are increased in this counter image. Also, the counter image is decreased every frame following the equation $T^{t+1} = \beta T^t, \beta \in (0, 1]$. This guarantees the recent tracks have more weight, making the system suitable for dealing with environmental changes that may occur in the scene. As a rule, β should

take values close to 1. Finally, having a target t that we know both its path and its velocity at every point in the scene (V_p), the velocity field is defined as:

$$V_w(C) = \frac{T(C)V_w(C) + V_p}{T(C) + 1}, \tag{5}$$

being C the target path.

3.2 Scene Properties

As we mentioned earlier, we have to take into account the scene properties. In this case, we forbid the algorithm to scape from the asphalt. Thus, in the neighbour updating step, we avoid to update the point in which there are no asphalt, Since we do not have any information a priori about the environment, we use the information provided by the tracking system. So, we use a threshold, τ, in the counter image T to determine whether a pixel is part of the asphalt or not. Fig. 3 shows an example of the construction of this image.

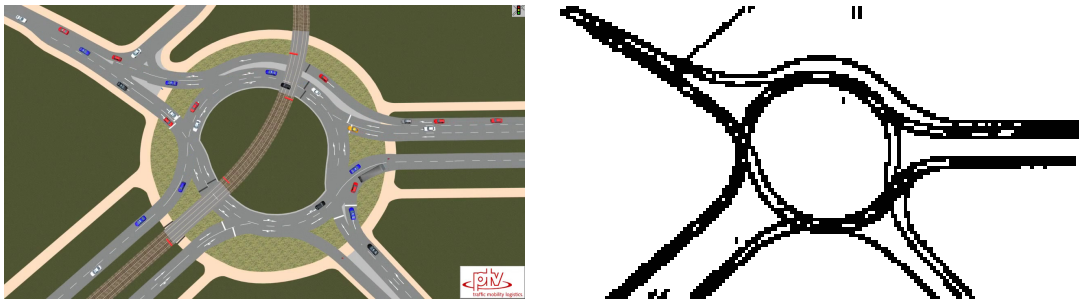


Fig. 3. Potential created after 2000 frames in a turnaround traffic scene

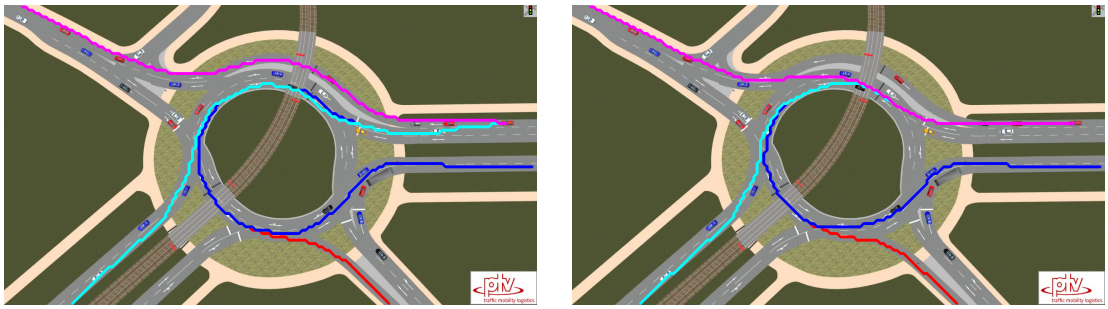
3.3 Experimental Results

To test this methodology we use the traffic simulator PTV Vissim provided by the PTV group. As mentioned before, one of the most challenging situations dealing with traffic analysis is the turnaround. Thus, we decide to use a turnaround scene[1], introducing a tram priority in order to increase the complexity. Using the velocity field defined in section 3.1, we clearly obtain low velocities near the rail because of the traffic lights. We reduce the velocity field image size, in order to increase the system efficiency. Testing different possibilities, we achieve that reducing the original video resolution by an 8 factor obtains the best results.

A target identification is also needed in order to avoid the tram in the potential image, which could cause the system to obtain poor results, since the cars cannot drive over the rails. In this case, a simple threshold size is enough to identify the train.

In the Fig. 4 we can see how the OUM method is updating the velocity field at every frame, causing the minimal paths to be modified. In the frame 500, the path that

[1] `http://www.youtube.com/watch?v=RtxEZINCpCw`



Frame 500 Frame 1200

Fig. 4. Minimal paths obtained in different times. The paths start from the upper-right roundabout entrance, reaching all the departures. All the routes are updated at every moment without needing to store all the different paths.

takes less time to reach the top-left departure, starting from the top-right of the image, involves the lane running off the turnaround. However, in frame 1200 the path changes, running within the turnaround lanes. Despite the fact that the examples showed involve the end of their respective paths in the scene, this system is able to compute the minimal path at every moment the target is detected, having only to take its current position as final position. We also test the methodology using real data provided by the NGSIM

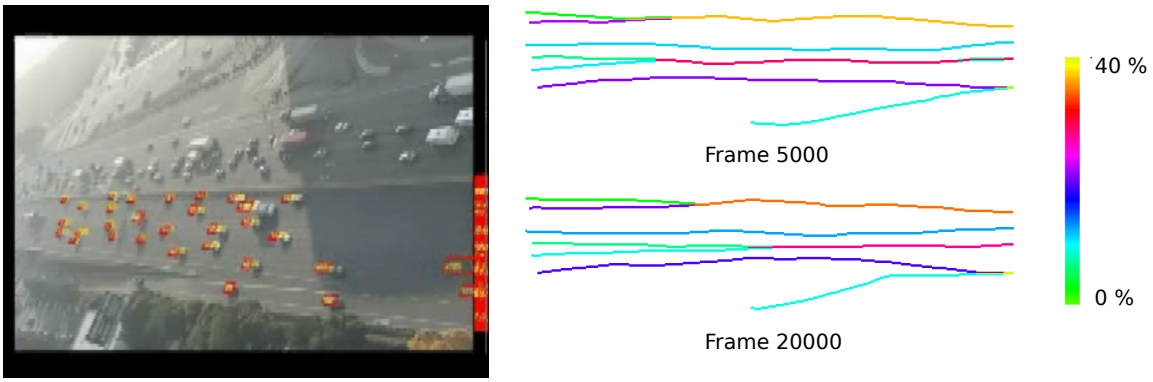


Fig. 5. Main paths information over a highway. At right, information of the most used trajectories, depending on the starting lane chosen. Vehicles tend to use the upper lanes to make room for the vehicles that enter using the bottom approaching lane.

dataset[2]. We use information about traffic movement over a highway, with more than 30 minutes recording information. In Fig. 5 we can see some of the obtained results. We show the minimal paths of the main trajectories the vehicles are following, depending on the starting lane chosen. We can see how the vehicles tend to use the upper lanes, which are faster than the others. Also, it use them to make room for the vehicles that use the approaching line, before entering the highway. These results confirm the idea that the vehicles try to use the path that involves the minimum time to reach the goal.

[2] `http://ngsim-community.org`

It is important to note that in this particular work no quantitative comparisons can be provided as the discussion is focused in assessing the use of a new paradigm for modeling of usual behavior which possesses a qualitative nature by itself. Future uses for this paradigm would include particular domains such as, for example, abnormal behavior detection which is not the point in this introduction.

4 Discussion

As we show in the examples explained before, the OUM algorithm is able to, using the information provided by the environment, determines the usual routes a target follows in order to reach its target. One of the most interesting fields of study in which our idea can be useful is the detection of abnormal behavior. Since we can detect the usual route of any target (we only need its initial position and its last known position as final point), it is possible to perform an abnormal detection system, in order to prevent accidents before they occur.

As mentioned before, techniques found in the literature needs the complete route to perform a comparison [16], which is useless in this topic. However, there exists techniques that can deal with partial routes, like Dynamic Time Warping [8]. However, this technique requires high computational cost and is more sensitive to noise in the tracking system. More recent techniques can predict the usual paths given any position in the scene [13], but have no information about past events.

The techniques found in the literature are mainly focused in clustering or probabilistic models. This means that, for performing a path computation, we have to compare every path stored against every new path we have. Using our methodology, we can reduce this complexity, since we only have to compare the new path against one usual route, that is, the result obtained with the method proposed in previous sections. Once we have the usual route, we can compare it against the new path using any trajectory distance measure [11], or even clustering techniques [3].

5 Conclusion

In this paper, a new vision for path analysis using directional forces is provided. The idea is to avoid the need of storing every possible path in the image. A technique based in the minimal path idea is provided and tested. A potential image and a velocity field are the information the method requires to compute the usual paths.

Although it is possible to include information a priori about the environment in this system, this one is able to work without it. To do that, a tracking system is needed to define the velocity field, which is updated at every frame, making this system robust against changes over the time, showing the flexibility of this methodology. Experimental results over a traffic analysis case of study show promising results.

In a future research a high-level reasoning about the results provided by these methods would be interesting to add, enabling the system to provide better information for the user. For example, a method that can show the frequency of the main routes used by the targets. Also, an online technique for detecting abnormal behavior, used to prevent possible accidents. An approximation of the OUM technique, which can reduce its computational time, is necessary to reach an real-time system.

References

1. Bouguet, J.: Pyramidal implementation of the lucas kanade feature tracker. Intel Corporation, Microprocessor Research Labs (2000)
2. Calderara, S., Cucchiara, R., Prati, A.: Detection of abnormal behaviors using a mixture of von mises distributions. In: IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007, pp. 141–146 (September 2007)
3. Cancela, B., Ortega, M., Fernández, A., Penedo, M.G.: Path Analysis in Multiple-Target Video Sequences. In: Maino, G., Foresti, G.L. (eds.) ICIAP 2011, Part II. LNCS, vol. 6979, pp. 50–59. Springer, Heidelberg (2011)
4. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: A minimal path approach. International Journal of Computer Vision 24, 57–78 (1997)
5. Elston, J., Stachura, M., Frew, E., Herzfeld, U.: Toward model free atmospheric sensing by aerial robot networks in strong wind fields. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 369–374 (May 2009)
6. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. Image and Vision Computing 14(8), 609–615 (1996)
7. Junejo, I., Javed, O., Shah, M.: Multi feature path modeling for video surveillance. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 2, pp. 716–719 (August 2004)
8. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: Proc. 6th Int. Conf. on Knowledge Discovery and Data Mining, pp. 285–289 (2000)
9. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(3), 397–408 (2005)
10. Moore, B.E., Ali, S., Mehran, R., Shah, M.: Visual crowd surveillance through a hydrodynamics lens. Commun. ACM 54(12), 64–73 (2011)
11. Morris, B., Trivedi, M.: Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 312–319 (2009)
12. Porikli, F., Haga, T.: Event detection by eigenvector decomposition using object and frame features. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2004., p. 114 (June 2004)
13. Saleemi, I., Shafique, K., Shah, M.: Probabilistic modeling of scene dynamics for applications in visual surveillance. IEEE Trans. Pattern Anal. Mach. Intell. 31(8), 1472–1485 (2009)
14. Sethian, J.A., Vladimirsky, A.: Ordered Upwind Methods for Static Hamilton–Jacobi Equations: Theory and Algorithms. SIAM Journal on Numerical Analysis 41(1), 325–363 (2003)
15. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control 40, 1528–1538 (1995)
16. Wang, X., Ma, K., Ng, G.-W., Grimson, W.: Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. International Journal of Computer Vision 95, 287–312 (2011)

3.3 Conference Paper: Trajectory Similarity Measures Using Minimal Paths

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Marcos Ortega Hortas
Affiliation: Universidade da Coruña, Spain

Co-author #3: Manuel Francisco González Penedo
Affiliation: Universidade da Coruña, Spain

Article title: Trajectory Similarity Measures Using Minimal Paths
Conference: 17th International Conference on Image Analysis and Processing (ICIAP)
Location: Naples, Italy
Date: September, 2013

Trajectory Similarity Measures Using Minimal Paths

Brais Cancela, Marcos Ortega, Alba Fernández, and Manuel G. Penedo

University of A Coruña,
Varpa Group, Department of Computer Science,
Campus de Elviña, s/n, Spain
{brais.cancela,mortega,alba.fernandez,mgpenedo}@udc.es
http://www.varpa.org

Abstract. Dealing with surveillance systems, large amount of distance measures are presented in order to classify both normal and abnormal behavior. Typically, techniques based in point-to-point distances are used. However, these techniques do not take into account information about the environment, like pits or restricted areas, for instance. Using a minimal path algorithm to model the usual paths, we develop new trajectory distance measures that are able to introduce information about the scene. The results obtained show promising results.

1 Introduction

Detecting human activities and behavior is a huge field of study in computer vision. One of the most active topics is related with the study of human behavior and their group relationships. This field has a special interest in surveillance systems. The idea of being able to detect abnormal behavior has being widely study. For instance, a strange movement could result in an abnormal behavior which has to be detected in order to throw an alarm.

The classical path classification methodologies are based in clustering techniques. Different configurations were used: direct [15], using techniques like k-means or fuzzy *c* means; agglomerative [5], where we merge clusters until we obtain the desired number; divisive [4], the top-down dual to agglomerative clustering; Hybrid [11], Graph-based [14] or Spectral [10]. All the techniques mentioned above are limited, since they require routes with the same number of samples to compute the clusters, and they are not easy to update along time. Suppose, for example, that an usual target is interrupted because of an object placed in the track. A new cluster is created with the new routes, but the previous cluster still remains in the system. A target which decides to jump that object, which clearly is an abnormal movement, will be declared as normal behavior because of the existing cluster. More recent techniques include the use of nonparametric Bayesian models [20], [19] or use models to predict the motion behavior [8].

To overcome the clustering issue, we developed a new system that can be easily updated [6]. We use a minimal path algorithm to model the abnormal behavior. In normal situations, a target tends to choose the route that costs the least time to reach its desired destination. Thus, trajectories that highly differ from this "ideal" route are marked as abnormal. A new metric were presented, based in a distance map algorithm, which requires a high computational time.

A. Petrosino (Ed.): ICIAP 2013, Part I, LNCS 8156, pp. 400–409, 2013.
© Springer-Verlag Berlin Heidelberg 2013

In this work we present a new trajectory distance measure that can be used in surveillance systems to detect abnormal behavior. This technique uses the properties of the minimal path algorithm to obtain a metric without increasing the computational time, solving the distance map technique disadvantage. This paper is organized as follows: section 2 shows different trajectory measure algorithms; section 3 describes our method; section 4 shows some experimental results and section 5 offers conclusions and future work.

2 State of the Art

We define a trajectory as a collection of the positions a target reach along its way. So, we have a collection of N positions defining a target route. To compare it, there exist in the literature different approaches for trajectory distance metrics. Methods based in classical measure techniques were used, like euclidean distance [10] or principal component analysis (PCA) [3]. However, these methods obtain poor results, requiring trajectories with the same size to be compared. Other attempts, like the modified Hausdorff distance [2], does not take into account the order into the trajectory points.

Thus, distance measure techniques have to be able to compare unequal length trajectories, while taking into account the route orientation. In [12], Keogh et al. presented the Dynamic Time Warping (DTW) technique. Basically, this method tries to find a time warping that minimizes the distance between two different trajectories. It can be used with trajectories with different sizes. Buzan et al. [5] introduced a similar idea, the Longest Common Subsequence (LCSS). It can also be used with unequal length data, becoming more robust to noise. The reason is that not all the trajectory points need to be matched. Similar to these methods, Piciarelli and Foresti (PF) [17] uses a dynamic time warping window, which is increased along time, that is, the maximum error allowed is low at the starting trajectory point, becoming larger while we are reaching the end. The performance of these metrics were tested in [16].

Although these methods can deal with the problems mentioned before, they all present a major issue for real domains: they do not take into account information about the environment. For instance, in Fig. 1, we can see two examples of situations these techniques cannot correctly address. In the left, two parallel trajectories are defined. Using the similarity measure it is easy to conclude that both trajectories are similar. However, the red route is produced by a counterclockwise car, which clearly is an abnormal behavior. On the right image, both the red and the green route are similar to the blue one, but the red one crosses the central reservation.

To solve situations like the previously illustrated one, we develop a methodology that is able to introduce information about the environment [6], based in the geodesic active contours [7] and the level set theory [13]. A modification of the minimal path approach using geodesic active contours performed by Cohen and Kimmel [9] is provided. In this work, starting at any given point p_0, a minimal path map over an image is obtained, with a $\mathcal{O}(N \log N)$ complexity, being N the number of pixels in the image. To do that, a potential image P is created, which includes information about the environment. Later, the potential used in the algorithm is defined as

$$\tilde{P}(p) = \omega + P(p), \tag{1}$$

(a) (b)

Fig. 1. These trajectories are defined as similar using classical distance measure techniques, while including the scene information the red routes are clearly abnormal

being ω the regularization term. This potential has to be defined so $\tilde{P} > 0$ and $P \geq 0$, meaning that $\omega > 0$. Values close to 0 implies pixels that are easy to reach, while higher values means the opposite.

Having this potential, the surface of minimal action U can be created. This surface assigns a value to each pixel in the image, which corresponds with the least cost that takes to reach that pixel, starting at the initial point p_0. To create that surface, the Fast Marching Method (FMM) proposed by Sethian is used [18]. An ordered upwind scheme is used, updating the cost of a given pixel $U_{i,j} = u$ following the equation

$$(\max\{u - U_{i-1,j}, u + U_{i+1,j}, 0\})^2 + (\max\{u - U_{i,j-1}, u + U_{i,j+1}, 0\})^2 = \tilde{P}_{i,j}^2. \quad (2)$$

Once the minimal path between the initial and the final point is obtained, we can evaluate the real path against this minimal cost approach. Note that the final point mentioned before is not necessarily the point where we stop to track a target. The minimal path can be computed each frame a target is detected, using the last position detected as final path. This is an advantage with respect to other methodologies, since the abnormal behavior can be detected as soon as it occurs.

So, in order to evaluate the real path against the minimal path approach, a distance map image is created. Having this minimal path algorithm in mind, it is easy to see that, if using as initial points all the points in a given trajectory $C(s)$ instead of the point p_0 in the original algorithm, the surface of minimal action U we obtain is a distance map, where the value $U_{i,j}$ of each pixel is the minimum cost that takes to reach that point, starting at any point that belongs to the trajectory $C(s)$. Note that, the distance between two consecutive pixels is defined by the potential $\tilde{P}$. In [6], a discussion about the potential image is provided.

The problem of this method is the computational cost needed to create the distance image. Since the algorithm is, in essence, equivalent to the minimal path approach, the cost to obtain the map is $\mathcal{O}(N \log N)$. Thus, our goal is to create a new trajectory distance measure that is able to include information about the environment, taking into account the properties of the minimal path approach, but avoiding unnecessary extra computations.

3 Minimal Path Metrics

Our goal is to use the properties of the Fast Marching Method used to create the minimal action surface U to detect if a trajectory is abnormal. That surface of minimal action has a convex like behavior, in the sense that, starting at any given point p, and following the gradient descent direction in U, we always converge to the initial point p_0. This means that the minimal action surface U has one local minimum, which is $U(p_0) = 0$. Furthermore, this geodesic active contour-based technique is consistent with the continuous problem, in the sense that the solution provided by the FMM becomes closer to the exact solution while reducing the grid. This property allows this algorithm to avoid the 'metrication error', which appears in the classical graph search algorithms, like A* or F*.

This property is crucial to present our methodology. If, for instance, we introduce the potential image $\tilde{P} = \tau > 0$, and we compute the minimal action surface U, starting at any given point within the grid p_0, we obtain a distance map, as the value at any point p, U_0, is the distance between this point with respect to the initial one p_0. And, contrary to the graph search algorithms, since the FMM is consistent with the continuous case, the solution we obtain is close enough to the euclidean distance. Note that this system is isotropic, having no information about directional forces. The cost to reach a point is always the same. no matter which direction the front-propagation come by.

However, using a potential like the mentioned before causes the system to lose the information about the environment, since this potential is constant all over the space. Fortunately, it is possible to compute the distance map regardless the type of potential used. We have to define another minimal action surface D, which is going to be updated using the equation

$$D(p) = \begin{cases} \dfrac{D(p_a) + D(p_b) + \sqrt{2\tau^2 - (D(p_b) - D(p_a))^2}}{2} & \text{if } \tilde{P}(p) > (U(p_b) - U(p_a)) \\ D(p_a) + \tau & \text{otherwise} \end{cases}, \tag{3}$$

where p_a and p_b are the neighbors used in the Eq. 2 to update the surface of minimal action $U(p)$. satisfying that $U(p_a) \leq U(p_b)$, and τ is the distance between two neighbor pixels. Typically, $\tau = 1$.

Thus, while computing the minimal action surface U we can compute the distance map D without any substantial cost increment. In algorithm 1 a pseudocode of our FMM implementation is presented.

Once we have the distance map related to a trajectory, we are able to perform a similarity measure that can detect abnormal behavior. For notation, we have a trajectory

$$Tr = \{p_0, p_1, \ldots, p_M\}, \tag{4}$$

where each point represents positions that are reached for the target. Note that this is an ordered sequence of events, where the position p_i is reached before the position p_{i+1}. These trajectories can be obtained using tracking techniques, being p_0 the first time a target is tracked. Since, as we demonstrate in [6], a target usually tends to follow the

Algorithm 1. Distance Surface Method

Definitions:

- *Alive* set: points of the grid for which U has been computed and it will not be modified.
- *Trial* set: next points in the grid to be examined (4-connectivity) for which an estimation of U is computed using the points in *alive* set.
- *Far* set: the remaining points of the grid for which there is not an estimate for U.

Initialization:

- For each point in the grid, let $U_{i,j} = \infty$ (large positive value).
  Put all points in the *far* set.
- Set the initial point p_0 to be zero:
  $U_{p_0} = 0$, $D_{p_0} = 0$, and put it in the *trial* set.

Marching loop:

- Select $p = (i_{min}, j_{min})$ from *trial* with the lowest value of U.
- If p is equal to p_1 being p_1 the final point then we finish.
- Else put p in *alive* and remove it from the *trial* set.
- If $\tilde{P}(i_{min}, j_{min}) < \tau$, for each of the 4 neighboring grid points (k, l) of (i_{min}, j_{min}):
  - If (k, l) belongs to *far* set, then put (k, l) in *trial* set.
  - If (k, l) is not in *alive* set, then set $U_{k,l}$ with Equation 2
  - and set $D_{k,l}$ with Equation 3.

path that cost less effort to reach the goal, we can conclude the relation between the minimal path distance with respect to a normal trajectory behavior is close to 1, that is,

$$\frac{\sum_{i=2}^{M} d(p_{i-1}, p_i)}{D(p_M)} \approx 1, \tag{5}$$

where $d(p_{i-1}, p_i)$ is the distance between two consecutive points in the trajectory. To compute this distance it is also possible to use the minimal path approach, starting in p_{i-1} instead of p_1. However, the points contained in the route are usually close together, meaning the euclidean distance often results in a good approach.

Having this in mind, different metrics are presented for detecting abnormal behavior. All these metrics are based in two different equations. The first one tries to obtain the relation between the target route and its associated minimal path. We called it Minpath Relation (MR), and is defined by

$$MR(p_N) = \left(\frac{\sum_{i=2}^{N} d(p_{i-1}, p_i)}{D(p_N)} - 1 \right)^2, \tag{6}$$

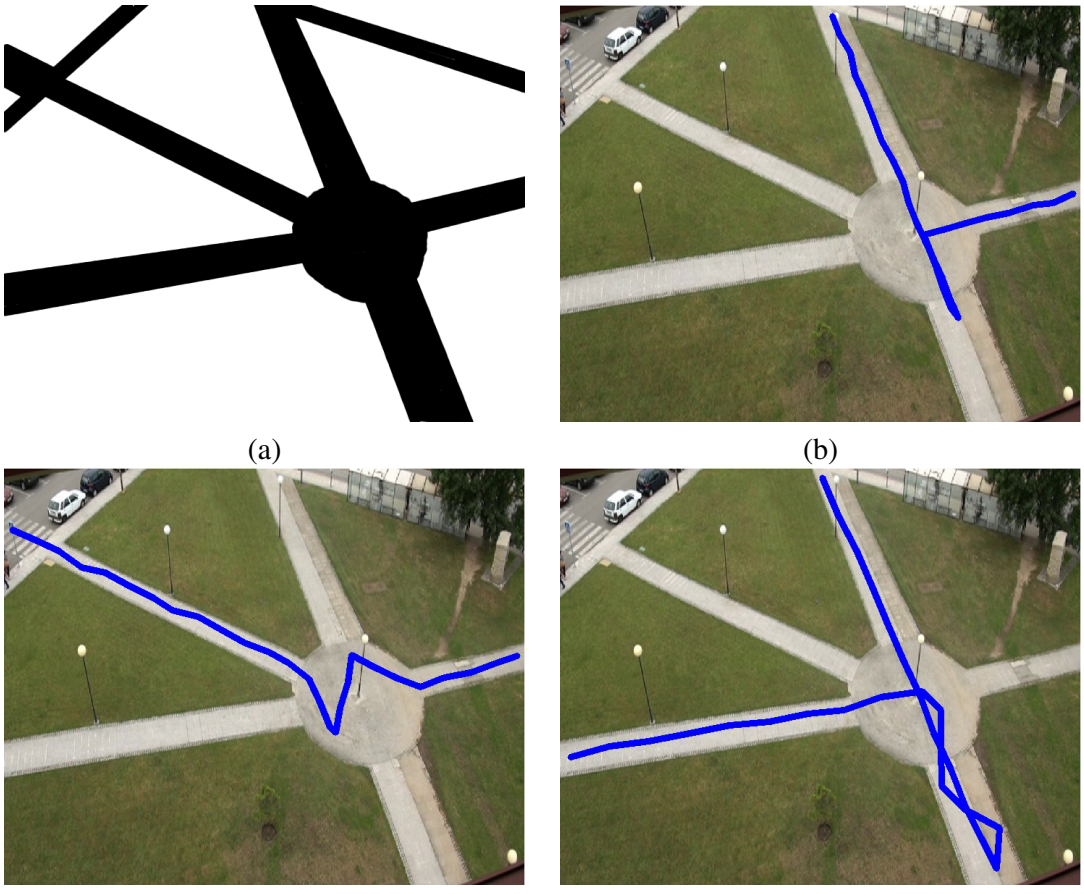
where $1 < N \leq M$. The second one tries to detect local variations in the MR metric. We called it Local Minpath Relation (LMR), and is defined by

$$LMR(p_N) = \left(\frac{d(p_{N-1}, p_N)}{D(p_N) - D(p_{N-1})} - 1 \right)^2. \tag{7}$$

In both metrics, values close to 0 mean the path is correct, while higher values could indicate an abnormal behavior.

4 Experimental Results

In our experiments we have used a dataset publicly available, BARD [1], in order to test the methodology. Several trajectories are developed over an intersection scene, resulting in more than 15000 trajectory points. In the experiments we decided to use a fixed potential image, that can be shown in Fig. 2-(a). We consider the grass areas as forbidden areas, using high values in the potential to model them.



(a) (b) (c) (d)

Fig. 2. Path trajectory examples. (a) Potential image used to computed the minimal path. (b, c, d) Abnormal behavior examples.

In first place, we decided to test different approaches of the metrics explained before. We decided to test, according to the MR equation, both MR, the Mean MR

$$MMR(p_N) = \frac{\sum_{i=2}^{N} MR(p_i)}{N}, \tag{8}$$

its variance

$$VMR(p_N) = \frac{\sum_{i=2}^{N} MR(p_i)^2}{N} - MMR(p_N)^2 \tag{9}$$

and its top value

$$TMR(p_N) = \max_{i \in [2..N]} p_N. \tag{10}$$

When dealing with the LMR equation, the mean (MLMR), the variance (VLMR) and the maximum (TLMR) are used. As mentioned earlier, values close to 0 mean there is no abnormal behavior in the route. Thus, we can detect abnormal situations by simply introducing a threshold. In Table 1 we show the results obtained using these metric over the BARD dataset. Looking at the ROC Area Under the Curve values obtained, we can conclude the metrics exposed obtain good results, especially MMR and TMR measures. Most of the errors achieved by these metrics are related with the difficulty of annotate the correct moment where a normal route starts being erratic. As a result, it is possible that some errors may occur because of bad manually annotations.

Table 1. Minpath Metric Results. Using the ROC Area Under the Curve metric, we found that all the techniques achieve good results.

Metric	**ROC AUC**
MR	0.9417
MMR	0.9691
VMR	0.9578
TMR	0.9673
MLMR	0.9535
VLMR	0.9448
TLMR	0.9512

Having these results, we conclude the MMR technique achieve the better results. However, this comparison is made by using a potential image with the same size of the video frame, in this case, (576×720). Although the computation of these metrics is equivalent to the computation of the minimal path method, $\mathcal{O}(N \log N)$, the time needed is too high to be used in real-time systems. This is not a problem when dealing with a fixed potential, where the potential cannot be updated along frames, because we only need to compute the minimal path one time per each target. Storing the minimal action surface U, we can obtain the MMR metric with a $\mathcal{O}(1)$ complexity.

However, in more complex scenarios, we need to use a potential that is going to be modified along time. In this case, we need to compute the minimal action surface U every time we want to obtain the MMR metric. Thus, we need to reduce the potential image size in order to reduce the computational time. In Fig. 2 we can see the results obtained by reducing the image. As we can see, similar results are obtained if we reduce the potential image to (72×90), allowing our method to speed up the response without decreasing significantly the performance of the metric.

In order to compare the results of the metrics mentioned before against the baseline techniques, we use the ROC curve. In Fig. 3 we compare our metrics with the baseline

Table 2. Minpath Metric Results. Using the ROC Area Under the Curve metric, we found that all the techniques achieve good results.

Decreasing factor	ROC AUC						
	MR	MMR	VMR	TMR	LMR	VLMR	TLMR
f = 1 (576×720)	0.9417	**0.9691**	0.9578	0.9673	0.9535	0.9448	0.9512
f = 2 (288×360)	0.9488	**0.9646**	0.9519	0.9619	0.9493	0.9411	0.9471
f = 4 (144×180)	0.9574	**0.9581**	0.9424	0.9545	0.9453	0.9379	0.9438
f = 8 (72×90)	**0.9528**	0.9480	0.9224	0.9395	0.9308	0.9267	0.9326
f = 16 (36×45)	0.8965	**0.9124**	0.8678	0.8909	0.8646	0.8695	0.8712
f = 32 (18×22)	0.7912	**0.8598**	0.8179	0.8371	0.7214	0.7290	0.7327

methods. Since our method clearly outperforms methods that need samples with the same size to perform the computation, we decide to compare our method against more powerful techniques, like the previously mentioned PF, LCSS and DTW. Moreover, we introduce our previous distance map based techniques. We can conclude that the MMR metric clearly outperforms the baseline methods, except the Weighted Distance Map. However, the MMR can obtain similar results while avoiding the computation of the distance map image, which has a $\mathcal{O}(N \log N)$ complexity, meaning our new method is more suitable for being used in real-time applications.

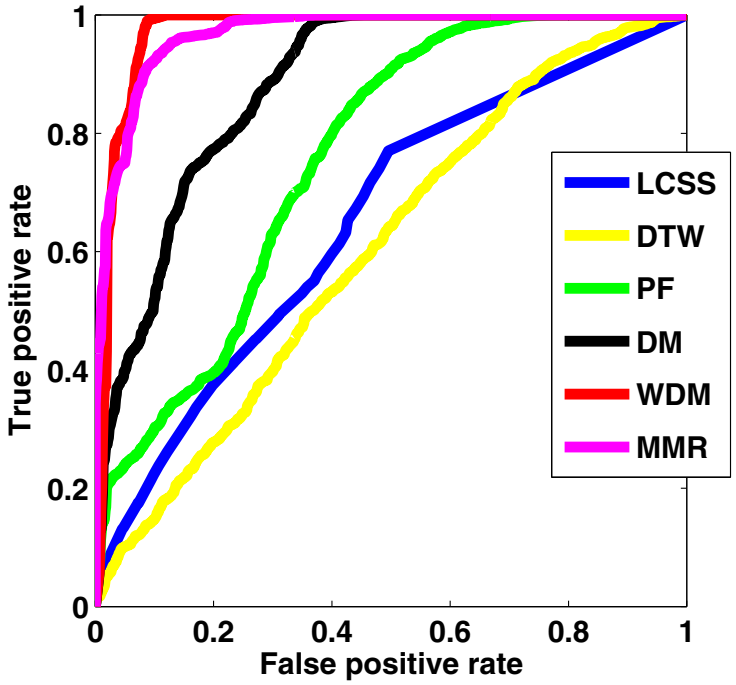


Fig. 3. ROC curve. Our new metric outperforms the baseline methods, with the exception of the Weighted Distance Map. However, the computational cost allows our new method to be more suitable to be used in real-time environments.

5 Conclusions

In this paper we present new metrics to detect abnormal behavior using target trajectories. Using minimal path techniques, which are proven to be useful detecting abnormal behavior, we develop new metrics that make use of the surface of minimal action properties, without increasing significantly the computation of such minimal paths. A comparison between different new metrics is performed, where the MMR obtains the better results. The results also show the potential image can be substantially reduced, having no significantly yield loss. Another comparison against baseline methods demonstrates our methods outperform classical abnormal behavior metrics, while obtains similar results to the recent Weighted Distance Map metric, which is proven to be computationally expensive.

In future works, we plan to introduce both direction and speed in our minimal path algorithm, allowing our system to have more information, in order to detect abnormal behavior that is not provided in our recent algorithm, like sudden speed changes or movements in the opposite direction of the usual routes.

Acknowledgments. This paper has been partly funded by the Consellería de Industria. Xunta de Galicia through grant contract 10TIC009CT, and by the Ministerio de Ciencia e Innovación through grant contract TIN2011-25476.

References

1. BARD, behavioral analysis and recognition dataset, `http://www.varpa.org/bard/`
2. Atev, S., Masoud, O., Papanikolopoulos, N.: Learning traffic patterns at intersections by spectral clustering of motion trajectories. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4851–4856. IEEE (2006)
3. Bashir, F., Khokhar, A., Schonfeld, D.: Object trajectory-based activity classification and recognition using hidden markov models. IEEE Transactions on Image Processing 16(7), 1912–1919 (2007)
4. Biliotti, D., Antonini, G., Thiran, J.P.: Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences. In: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), vol. 2, pp. 50–57. IEEE Computer Society, Washington, DC (2005)
5. Buzan, D., Sclaroff, S., Kollios, G.: Extraction and clustering of motion trajectories in video. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 2, pp. 521–524. IEEE (2004)
6. Cancela, B., Ortega, M., Penedo, M., Novo, J., Barreira, N.: On the use of a minimal path approach for target trajectory analysis. Pattern Recognition 46(7), 2015–2027 (2013)
7. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22(1), 61–79 (1997)
8. Chen, Z., Wang, L., Yung, N.H.: Adaptive human motion analysis and prediction. Pattern Recognition 44(12), 2902–2914 (2011)
9. Cohen, L., Kimmel, R.: Global minimum for active contour models: A minimal path approach. International Journal of Computer Vision 24(1), 57–78 (1997)
10. Hu, W., Xie, D., Fu, Z., Zeng, W., Maybank, S.: Semantic-based surveillance video retrieval. IEEE Transactions on Image Processing 16(4), 1168–1181 (2007)

11. Karypis, G., Han, E.H., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. Computer 32(8), 68–75 (1999)
12. Keogh, E., Pazzani, M.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 285–289. ACM (2000)
13. Kimmel, R., Amir, A., Bruckstein, A.: Finding shortest paths on surfaces using level sets propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(6), 635–640 (1995)
14. Li, X., Hu, W., Hu, W.: A coarse-to-fine strategy for vehicle motion trajectory clustering. In: The 18th International Conference on Pattern Recognition, ICPR 2006, vol. 1, pp. 591–594 (2006)
15. Morris, B., Trivedi, M.M.: An adaptive scene description for activity analysis in surveillance video. In: ICPR, pp. 1–4. IEEE (2008)
16. Morris, B., Trivedi, M.M.: Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In: CVPR, pp. 312–319. IEEE (2009)
17. Piciarelli, C., Foresti, G.: On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters 27(15), 1835–1842 (2006)
18. Sethian, J.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences 93(4), 1591–1595 (1996)
19. Wang, X., Ma, K.T., Ng, G.W., Grimson, W.E.L.: Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. International Journal of Computer Vision 95(3), 287–312 (2011)
20. Wang, X., Ma, X., Grimson, W.E.L.: Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(3), 539–555 (2009)

3.4 Conference Paper: Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths

Author #1: Brais Cancela Barizo
Affiliation: Universidade da Coruña, Spain

Co-author #2: Marcos Ortega Hortas
Affiliation: Universidade da Coruña, Spain

Co-author #3: Manuel Francisco González Penedo
Affiliation: Universidade da Coruña, Spain

Article title: Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths
Conference: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
Location: Columbus, Ohio, USA
Date: June, 2014

2014 IEEE Conference on Computer Vision and Pattern Recognition

Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths.

B. Cancela, A. Iglesias, M. Ortega, M. G. Penedo
VARPA Group, Universidade da Coruña
Campus de Elviña s/n, A Coruña, Spain
http://www.varpa.org/

Abstract

This paper presents a novel methodology for modelling pedestrian trajectories over a scene, based in the hypothesis that, when people try to reach a destination, they use the path that takes less time, taking into account environmental information like the type of terrain or what other people did before. Thus, a minimal path approach can be used to model human trajectory behaviour. We develop a modified Fast Marching Method that allows us to include both velocity and orientation in the Front Propagation Approach, without increasing its computational complexity. Combining all the information, we create a time surface *that shows the time a target need to reach any given position in the scene. We also create different metrics in order to compare the* time surface *against the real behaviour. Experimental results over a public dataset prove the initial hypothesis' correctness.*

1. Introduction

Modelling human behaviour is a huge field of study in computer vision. One of the most active topics is related with the trajectory analysis, that is, how is the people usual behaviour, having a special interest in areas like video surveillance. However, a lot of questions remain unsolved. For instance, what information makes people decide which path they should use to reach their goal?. Or what makes a movement abnormal? The answer for the questions are crucial in order to obtain a good trajectory modelling.

There exist two different approaches in the literature to model human trajectory behaviour. On one hand there are the Computer Vision techniques. They are mainly focused in classifying all the trajectories in the scene, mostly using clustering techniques, like Hybrid [12], agglomerative [3], where we merge clusters until we obtain the desired number; divisive [1], Graph-based [14], Spectral [11] or direct [17], using techniques such as k-means or fuzzy *c* means. These techniques require every path to have the same number of detections to update the patterns, making the updating procedure very difficult. More recently, new techniques are used, like the use of non-parametric Bayesian models [27], [26], [13] or using models to predict the motion behaviour [6].

On the other hand there are the techniques based in *social force* models [10]. They are based in the idea that some stimuli, like the scene properties and other people interactions, affect the pedestrian trajectory [9], [2]. This approach is often used in computer graphic schemes, developing a set of different *forces* that are added to infer the new movement [23], [20], [24]. The main drawback of these techniques is that although they are good approaches to model the usual human behaviour, there exist infinite solutions to model a normal behaviour. Thus, how can we use these kind of systems to decide whether a trajectory is abnormal or not?

More recently, techniques that try to merge computer vision techniques with social models are arising. For instance, some works introduce the *social force* model to detect abnormal behaviour [15], [19]. Flow models were also included to predict crowd behaviour [16]. More recently, the use of minimal paths were introduced to model the usual behaviour [5], [4]. However, this approach does not take into account velocity patterns or orientation in the propagation procedure.

Based on the latter, we propose a new methodology to model pedestrian trajectory behaviour. Our solution is based in the hypothesis that, typically, when people try to reach a destination, they use the path that takes less time. We take into account both the velocity and the orientation of the usual motion to create a *time surface*, where each node shows the time needed to reach it if the person behaviour is usual. We create a modified Fast Marching Method (FMM) [22] which includes the mentioned extra information without increasing the computational cost. Using this technique, only a potential and a velocity surfaces are needed in order to establish the so-called *time surface*. We also present different metrics to test a path's "degree of abnormality", in

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.327

IEEE computer society

order to experimentally prove that our initial hypothesis is correct. Due to the lack of datasets providing some ground truth, we provide a theoretic statistical probe of correctness, along with some visual confirmation, as other approaches do. No quantitative results against other techniques are provided, since no abnormal detection results were provided in any other paper, which focus in clustering techniques our algorithm do not need.

This paper is organized as follows: section 2 describes our modified FMM and the metrics created to model the pedestrian behaviour; section 3 shows implementation details about our algorithm; section 4 shows some experimental results and section 5 offers conclusions and future work.

2. The Governing Equations

In this section we develop our model for pedestrian trajectory modelling. We begin to set the hypothesis that are used to establish our methodology. We subsequently introduce our modified FMM that allows us to create the *time surface*. Finally, we introduce some metrics in order to test the pedestrian behaviour. We defer the implementation of these techniques to section 3. For coherence with previous approaches, we are going to follow the notation introduced in [7].

Hypothesis 1 *Each person tries to reach a geographic goal.*

Since this model tries to model pedestrian trajectories, it is needed that targets have the intention to reach some goal within or outside the scene. As a consequence, people that are stopped or moving erratically are considered as abnormal movements.

Hypothesis 2 *The trajectory used to reach the goal is ruled by the common pedestrian behaviour.*

This is a crucial point in this algorithm. In [5], the minimal path is ruled by a potential image that contains, for every node, the number of people that reach it. However, in our algorithm, we also include the velocity of the targets. Despite this, our model will also be driven by the people count estimation.

Hypothesis 3 *People move at the maximum speed possible.*

That is, the speed of every person is defined by a velocity field.

Having this hypothesis in mind, we can conclude that the usual path can be modelled as a minimal path approach. We propose a modified FMM in order to create a *time surface*, which is created taking into account the information about both the people frequency and their velocity. In algorithm 1 the method is explained. In essence, the structure is similar to the FMM. It only differs in the updating procedure, since

Algorithm 1 Time Surface Fast Marching method

Definitions:

- p_0: the initial point, the first time a target is tracked.
- U: surface of minimal action, driven by the people frequency.
- T: time surface: every node contains the time needed to reach it starting in the initial point.
- *Alive* set: points of the grid for which U has been computed and it will not be modified.
- *Trial* set: next points in the grid to be examined (4-connectivity) for which a estimation of U is computed using the points in *alive* set.
- *Far* set: the remaining points of the grid for which there is not an estimate for U.

Initialization:

- For each point in the grid, let $U_{i,j} = \infty, T_{i,j} = \infty$ (large positive value).
  Put all points in the *far* set.
- Set the start point $(i, j) = p_0$ to be zero:
  $U_{p_0} = 0, T_{p_0} = 0$, and put it in the *trial* set.

Marching loop:

- Select $p = (i_{min}, j_{min})$ from *trial* with the lowest value of U.
- Put p in *alive* and remove it from the *trial* set.
- For each of the p's neighbours (k, l) of (i_{min}, j_{min}):
  - If (k, l) belongs to *far* set, then put (k, l) in *trial* set.
  - If (k, l) is not in *alive* set, then set $U_{k,l}$ with Equation 3, and $T_{k,l}$ with Equation 4.

now we have two different output maps: U, the classical minimal path surface; and T, the *time surface*, where the time needed to reach every point is stored. T surface is driven by the U surface. That means we use the classical minpath updating procedure in order to update the surface. Thus, we need the best horizontal and vertical neighbours to do that. So, having the point $p_N = (i, j)$ to be updated we define the points $p_H = p \in \{(i+1, j), (i-1, j)\} | \min U(p)$ and $p_V = p \in \{(i, j+1), (i, j-1)\} | \min U(p)$. Then, we have the points

$$p_a = p \in \{p_H, p_V\} | \min U(p) \quad (1)$$

$$p_b = p \in \{p_H, p_V\} | \max U(p), \quad (2)$$

which are the two points used to perform the updating procedure. Hence, to update the minimal action surface U we

use the following equation

$$U_{p_N} = \begin{cases} \dfrac{U_{p_a} + U_{p_b} + \sqrt{\Delta_1}}{2} & \text{if } \tilde{P}_{p_N} > (U_{p_b} - U_{p_a}) \\ U_{p_a} + \tilde{P}_{p_N} & \text{otherwise} \end{cases}, \tag{3}$$

being $\tilde{P}(p_N)$ the potential surface, which satisfies that $\tilde{P} > 0$; and $\Delta_1 = 2\tilde{P}^2_{p_N} - (U_{p_b} - U_{p_a})^2$ the discriminant. To update the time surface, we use the same points p_a and p_b defined in the minimal action surface procedure. The equation is similar to the previous equation, that is,

$$T_{p_N} = \begin{cases} \dfrac{T_{p_a} + T_{p_b} + \sqrt{\Delta_2}}{2} & \text{if } \tilde{P}_{p_N} > (U_{p_b} - U_{p_a}) \\ T_{p_a} + \upsilon_{p_N} & \text{otherwise} \end{cases}, \tag{4}$$

being $\upsilon_{p_N} = \frac{1}{V_{p_N}}$ the velocity inverse; and $\Delta_2 = 2\upsilon^2_{p_N} - (T_{p_b} - T_{p_a})^2$ the discriminant. Note the condition $\tilde{P}_{p_N} > (U_{p_b} - U_{p_a})$ refers to the first equation, which guarantees that $\Delta_1 > 0$. However, in the second equation we cannot guarantee this situation in the discriminant. To solve that, we introduce a restriction in the procedure that allows the *time surface* to change the minimal action surface U. If we found that $\Delta_2 < 0$, we update both the minimal action surface U and the *time surface* T with the default condition. The computational complexity of the algorithm remains equal to the FMM, that is, $\mathcal{O}(M \log M)$, being M the numbers of nodes in the potential surface $\tilde{P}$.

2.1. Behavioural Metrics

Once we have computed the surface T, we have to establish whether a trajectory is abnormal or not. We can define a path as $P = \{p_0, p_1, \ldots, p_M\}$, where each point represents positions that are reached for the target, and its associated time as

$$P_t = \{t_{p_0}, t_{p_1}, \ldots, t_{p_M}\}. \tag{5}$$

Intuitively, without any information about the environment, the human brain detect as abnormal behaviour erratic movements, such as sudden orientation changes or zigzag movements. However, taking into account the scene properties, it may be the only way to reach the target, causing the path to be usual. An example of this behaviour could be a mountain road, climbing to the top like a snake. However, in our assumption, we only take into account the initial and final point of the trajectory. According with our assumptions, the average time required to reach a goal in the scene, starting at any given position, is stored in the surface T. Thus, a new hypothesis is established, that is,

Hypothesis 4 *If a path P have an usual behaviour, then $\forall p \in P, T_p \approx P_t(p)$,*

that is, the relation between the real and the expected time is $\frac{T_p}{P_t(p)} \approx 1$. Note that this idea is similar to the distance hypothesis performed in [4]. However, we think that the metrics used are not well suited, since the division between the two factors could obtain lower results when the numerator is too small compared with the denominator. Having this in mind, we develop a metric that allows us to measure a path's "abnormality". We called it Time Log-Likelihood (TL), and is defined by

$$TL(p_N) = \| \log(T_{p_N}) - \log(t_{p_N} - t_{p_0}) \|, \tag{6}$$

where $1 < N \leq M$. Values close to 0 mean the path is correct, while higher values could indicate an abnormal behaviour. Note that this final point mentioned in this metric is not necessarily the moment when the target leaves the scene. It is only a moment when we decide to evaluate a trajectory. This strong advantage allows this method to be used in real-time systems, since the computational complexity of the metric, once the surface T is computed, is $\mathcal{O}(1)$.

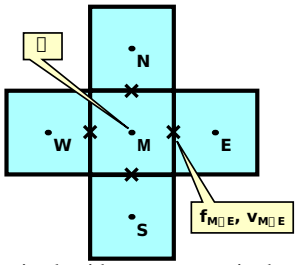


Figure 1. Discretized grid structure. ρ is the number of times a target reaches the node, whilst f and v are both the number of times a target crosses the path in each direction, and their most common speed.

3. Implementation

The model described in the previous section needs an implementation of the different structures that are used in the algorithm. Specifically, the potential $\tilde{P}$ and the velocity V surfaces. To compute these surfaces, we follow a similar approach that can be found in [23]. Since this method uses digital images, we discretized space into a regular grid. For each node/pixel, we store a set of properties, according to the schema shown in Fig. 1, into a 2D array we call P, being ρ the number of people that reach the node. We store anisotropic fields with four floats per cell corresponding to the east, north, west and south faces of each pixel ($\theta = \{0, 90, 180, 270\}$). $f_{M \to \{E,N,W,S\}}$ are the number of people that crosses each one of the pixel faces, while $\upsilon_{M \to \{E,N,W,S\}}$ are their most common speed. With this information, we are able to create $\tilde{P}$ and V surfaces.

To select the most common speed velocity, we use a Kernel Density Estimator [18]. To estimate the bandwidth, we use the *Rule of Thumb* method [8], [21]. Once we have the probability density function, we select the most common speed as the value with higher probability.

3.1. Potential Surface $\tilde{P}$

To model the potential surface we use the ρ parameter. Thus, having a point p, its potential value is defined by

$$\tilde{P}(p) = \frac{1}{\rho_p} + \omega, \tag{7}$$

being ω the regularization parameter, typically $\omega = 0$. We also have to define the p neighbourhood. The FMM uses a 4-connectivity procedure. However, in our case, we restrict the front-propagation technique to the directions that are commonly used, that is, being $M = \{p_E, p_N, p_W, p_S\}$ the usual p 4-connectivity neighbours,

$$\forall m \in M, m \text{ is } p\text{'s neighbour} \Leftrightarrow f_{p \to m} > \alpha, \tag{8}$$

being α a manual threshold. Note that this sentence does not imply the opposite. For instance, p_S can be a neighbour of p but p could not be a neighbour of p_S. With this restriction we can include orientation in the front-propagation procedure.

3.2. Velocity Surface V

As defined in Eq. 4, having any given point p to be updated, we have to find the value $v_p = \frac{1}{V_p}$ in order to update the time surface front. When the default condition is used, that is, only the point p_a is used to update the front it is easy to obtain the velocity, since it is $V_p = v_{p_a \to p}$.

However, when dealing with the first condition, we have two different velocities that have to be combined, $v_{p_a \to p}$ and $v_{p_b \to p}$. In order to establish an accurate solution to this problem, we define another surface D to be computed. This surface stores, for any given node p, the distance needed to reach it starting in p_0 and following the front-propagation method defined in Algorithm 1. This surface can be computed at the same time the surface of minimal action U is computed, as the time surface T does. To update the distance surface we follow the equation

$$D_{p_N} = \begin{cases} \dfrac{D_{p_a} + D_{p_b} + \sqrt{\Delta_3}}{2} & \text{if } \tilde{P}_{p_N} > (U_{p_b} - U_{p_a}) \\ D_{p_a} + 1 & \text{otherwise} \end{cases}, \tag{9}$$

being $\Delta_3 = 2 - (D_{p_b} - D_{p_a})^2$ the discriminant, and p_a and p_b the best neighbours defined in the minimal action surface procedure. To obtain the value V_p we make use of the gradient descendant in D, that is,

$$V_p = \frac{\nabla D_{p_a} v_{p_a \to p} + \nabla D_{p_b} v_{p_b \to p}}{\nabla D_{p_a} + \nabla D_{p_b}}, \tag{10}$$

where $\nabla D_{p_{\{a,b\}}} = \| D_p - D_{p_{\{a,b\}}} \|$.

4. Experimental Results

When trying to test any trajectory analysis, the same problem arises: there is a total absence of ground truth information, except in the BARD dataset [5]. However, this dataset is too small (only over 600 trajectories) and does not have any time information included. Thus, we decided to use a dataset that included a high number of trajectories, the single camera MIT trajectory dataset [25]. It contains $40,453$ different trajectories obtained from a parking lot scene within five days. We use half the trajectories as training, and the other half as testing.

Having all of this information, it is very hard to define whether a trajectory is normal or not. In related papers, they use some visual information to probe its effectiveness [25], [27], [26], [28]. In our approach, we also focus the solution as a statistical problem. Since all the earlier attempts to model the human trajectory behaviour have used some learning methods to determine the usual behaviour, we extract the idea that, having no information about the environment, every method consider the most usual paths as normal movements, being the outliers the abnormal ones.

Ideally, we expect an abnormal behaviour measure to have an asymptotic curve, like $\frac{1}{x}$, where the most part of the trajectories are normal, with a few abnormal movements. That is, the more erratic a trajectory is, the lower frequency it has. Thus, we can establish an inverse correlation between this two properties. In the top density function in Fig. 2-(a) we can see our metric has this behaviour. However, we have to check that, as we assume, lower values correspond to normal trajectories, while higher values means the opposite. To that end, we length-normalize every trajectory and perform a Fuzzy C means clustering into a large number of clusters. When we visualize the results, we saw that some trajectories detected as normal has some erratic movements in the middle. So, we decided to include two additional metrics: the Mean TL (MTL), which plots the mean of all the TL measures within the same trajectory, and the Maximum TL (MaxTL), which indicates its higher value. In Fig. 2-(a) we can see the density function of these metrics. We found that these two metrics perform really bad compared with the TL metrics, especially the MaxTL. Additionally, Fig. 2-(b) shows that the number of results near to ∞ is higher in the new metrics. This is really interesting, because when plotting the results of the clusters, according with the MaxTL value, we see some pattern (Fig. 3). The bad accuracy of the MaxTL metric is related to failures in the tracking system. That is, when due to a tracking failure a bad match is provoked, the system can detect it. This is an outstanding property, as it can be used to increase tracking performance.

4.1. Crowded Scenes

We demonstrate how this method can detect and classify every trajectory. However, at this point it can be argued

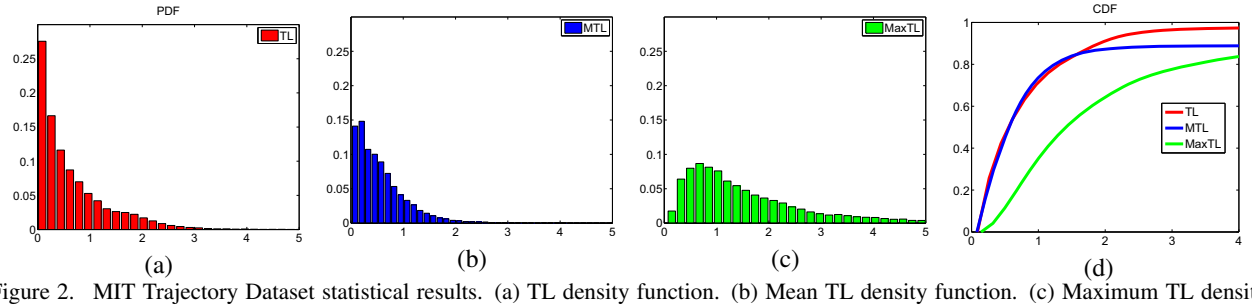


Figure 2. MIT Trajectory Dataset statistical results. (a) TL density function. (b) Mean TL density function. (c) Maximum TL density function. (d) Cumulative density function of each metric. Although the TL metric obtains good results, both its mean and its max value has poor quality. This seems to infer the trajectories are not accurate.

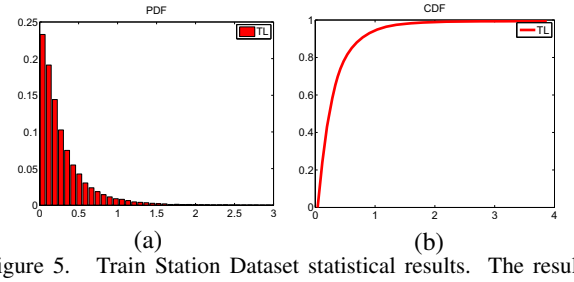


Figure 5. Train Station Dataset statistical results. The results suggest the effect of the rest of the people in a crowded scene has little impact in a target behaviour.

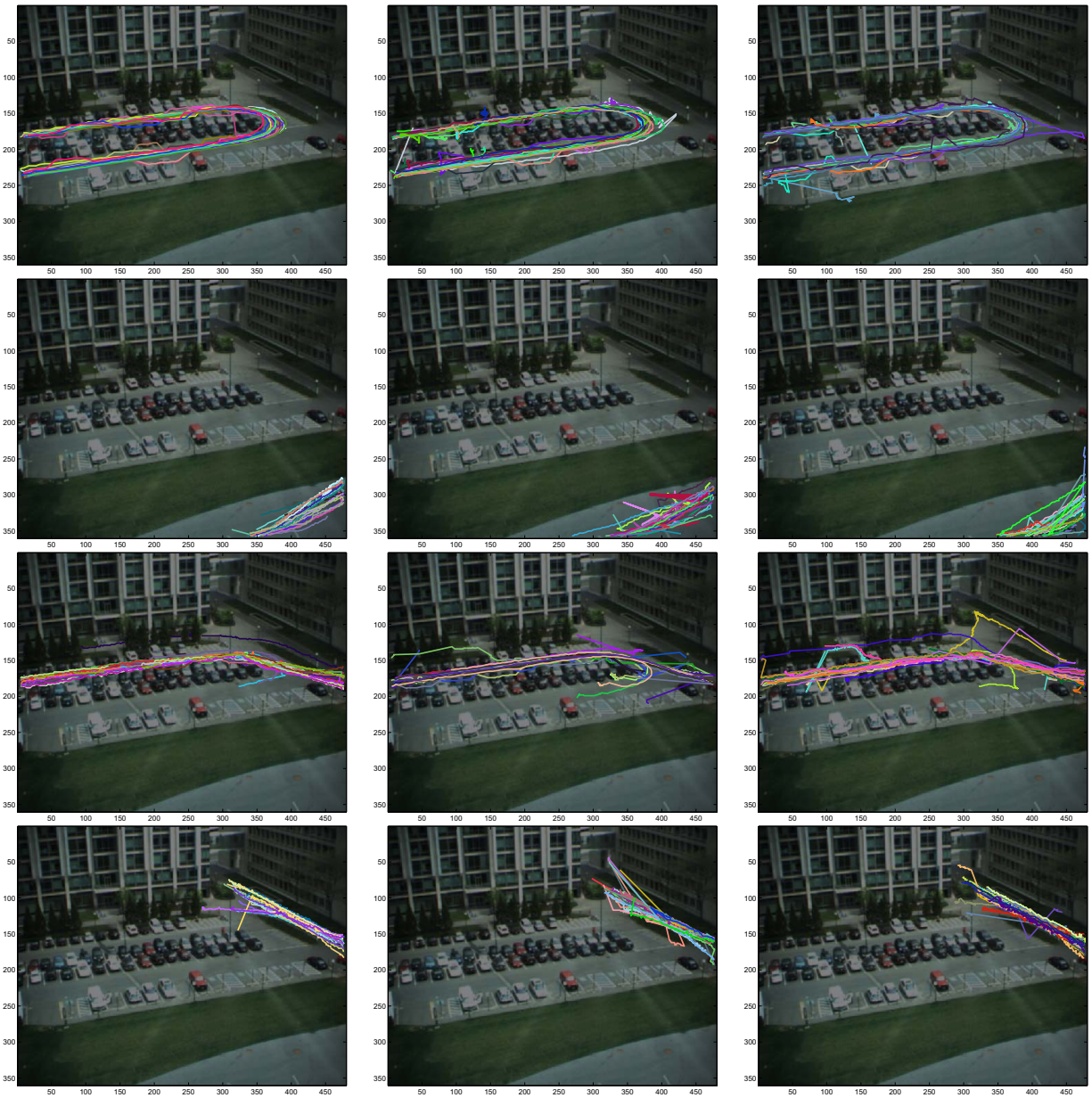
that the method could not work in crowded scenes, where the rest of the targets can affect a target's behaviour, making that usual direction may not be chosen. To that end, we select the Train Station Dataset [28], where $42,821$ trajectories were recorded in only 33 minutes. Performing the same experiments as in the previous dataset we obtain similar results (Fig. 5). That is really interesting, because we can conclude that the effect caused in a target by the rest of the people is not as huge as one may think. And with this hypothesis, we can pre-compute the time surface at the beginning without significantly increasing the metric error. Having this in mind, computing the behaviour at each position can be done in constant time.

In Fig. 4 we can see some trajectories the system detect as normal behaviour. As we can see, some movements were caused because of other people interaction, but the system remains robust against it. In a similar way, in Fig. 6 we can see some abnormal trajectories, and also when the system is changing its decision. This images show how our method can detect the degree of 'normality' at every moment a target is detected, providing a very powerful technique for surveillance scenarios.

The results obtained both in normal and high dense scene are better than expected. The fact that the number of people in the scene does not have a high impact in the target's behaviour makes this system very useful in very different situations. Previous methods group the scene in different regions with same behaviour [21] or cluster the trajectories in different patterns [25]. In our case, we cluster every 'time surface' taking into account each target's initial position. This idea is very promising, since in the most part of scenes the number of entrances is low. Thus, in future research it would be interesting to create only the surfaces related with these regions, causing the system to detect abnormal behaviours in constant time. To our knowledge, this is the first method that can be able to obtain it, since we do not have to compare the new trajectory against all the different pattern stored in the system, like the other methods do.

5. Conclusions

We proposed a novel idea to classify human trajectories, based in the idea that a person uses the path that takes less time to reach its target. Using the information about previous targets, we define a new potential field that is used to create a 'time surface' where, starting at any given point in the image, can predict the average time needed to reach any other position, assuming the target has an usual behaviour. Having this information, we introduce a new temporal metric to decide whether a trajectory is abnormal or not. Selecting two complicated scenarios, we prove that our initial hypothesis is correct, having an important contribution to establish a new way to define trajectory behaviour. The lack of datasets providing ground truth make impossible the task of evaluating our algorithm against other state of the art techniques. However, our method offers many advantages against techniques based in clustering or bayesian models, since it is able to compute the degree of 'normality' at every trajectory instant, and it is robust against the influence of other targets. Furthermore, it can detect tracking errors, making it suitable to improve any tracking system that can be integrated with our system. In future work we aim to further refine this method, using statistical techniques to model both velocity and time, instead of only using the most usual speed for every position. We also aim to expand this methodology to multiple camera frameworks.



Low $MaxTL$ trajectories | High $MaxTL$ trajectories | Trajectories with $MaxTL = \infty$

Figure 3. MIT Trajectory Dataset results. In the first column, trajectories with low $MaxTL$ value. In the second, higher values. In the third, trajectories for which a minimal action surface cannot be created. Trajectory colours are randomly selected.

References

[1] D. Biliotti, G. Antonini, and J. P. Thiran. Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)*, volume 2, pages 50–57. IEEE, 2005. 1

[2] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525, 2001. 1

[3] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 521–524. IEEE, 2004. 1

[4] B. Cancela, M. Ortega, A. Fernández, and M. G. Penedo. Trajectory similarity measures using minimal paths. In *Image Analysis and Processing–ICIAP 2013*, pages 400–409. 2013. 1, 3

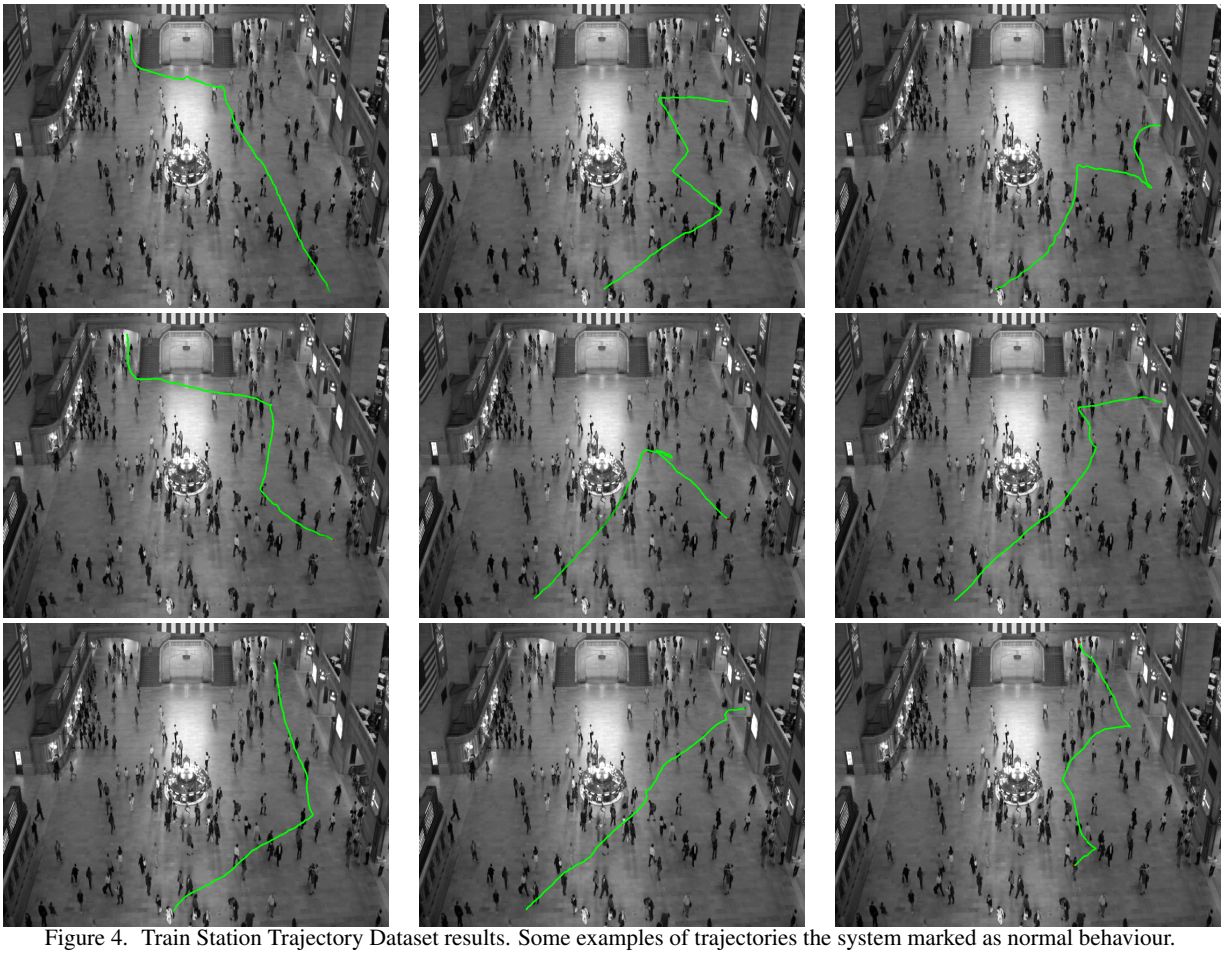


Figure 4. Train Station Trajectory Dataset results. Some examples of trajectories the system marked as normal behaviour.

[5] B. Cancela, M. Ortega, M. Penedo, J. Novo, and N. Barreira. On the use of a minimal path approach for target trajectory analysis. *Pattern Recognition*, 46(7):2015–2027, 2013. 1, 2, 4

[6] Z. Chen, L. Wang, and N. H. Yung. Adaptive human motion analysis and prediction. *Pattern Recognition*, 44(12):2902–2914, 2011. 1

[7] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *International journal of computer vision*, 24(1):57–78, 1997. 2

[8] K. Dehnad. Density estimation for statistics and data analysis. *Technometrics*, 29(4):495–495, 1987. 3

[9] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000. 1

[10] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 1

[11] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based surveillance video retrieval. *Image Processing, IEEE Transactions on*, 16(4):1168–1181, 2007. 1

[12] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999. 1

[13] J. Li, S. Gong, and T. Xiang. Learning behavioural context. *International journal of computer vision*, 97(3):276–304, 2012. 1

[14] X. Li, W. Hu, and W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 591 –594, 2006. 1

[15] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009. 1

[16] B. E. Moore, S. Ali, R. Mehran, and M. Shah. Visual crowd surveillance through a hydrodynamics lens. *Communications of the ACM*, 54(12):64–73, 2011. 1

[17] B. Morris and M. M. Trivedi. An adaptive scene description for activity analysis in surveillance video. In *ICPR*, pages 1–4. IEEE, 2008. 1

[18] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. 3

[19] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target

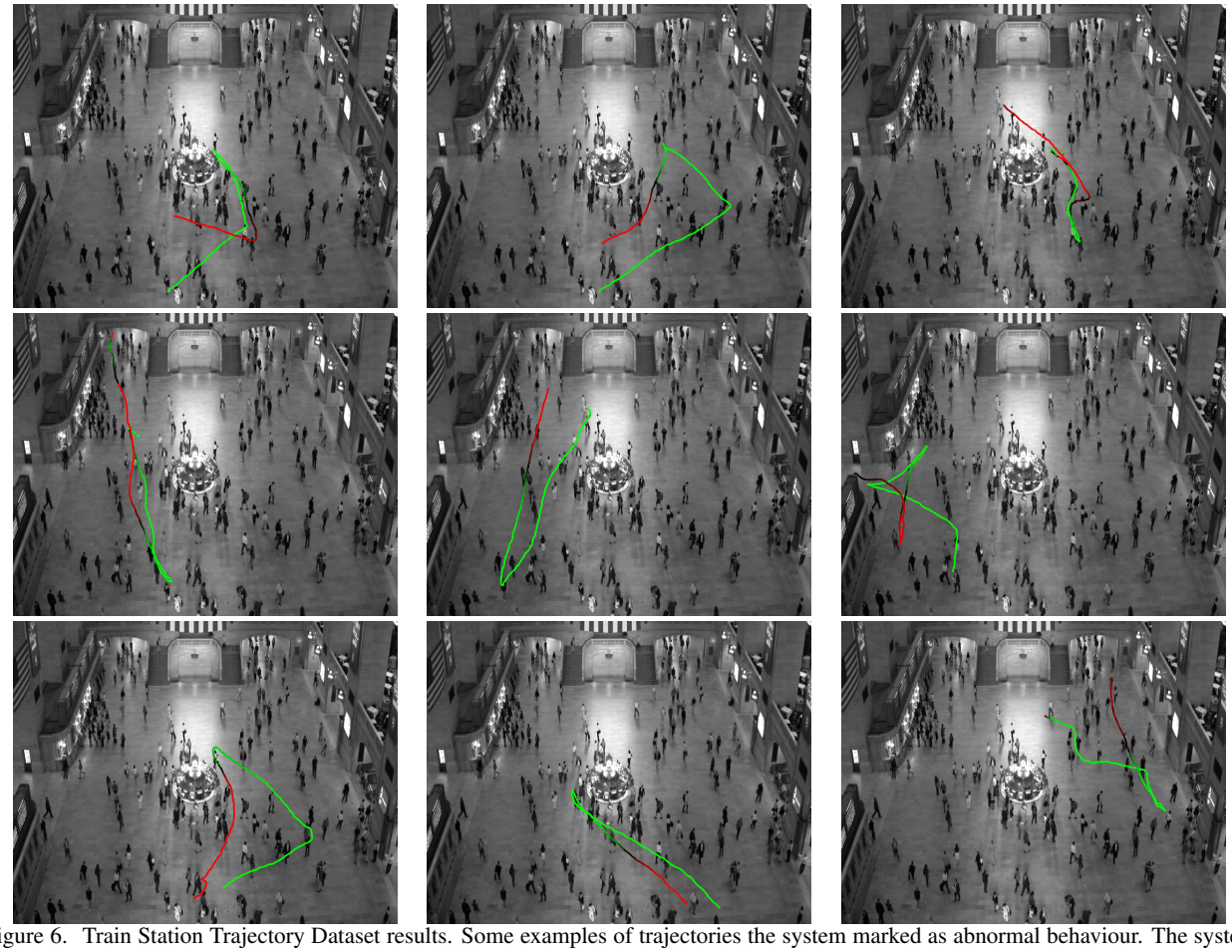


Figure 6. Train Station Trajectory Dataset results. Some examples of trajectories the system marked as abnormal behaviour. The system is able to determine, at any time, the degree of a bnormality (red means the trajectory is abnormal).

tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009. 1

[20] C. Reynolds. Big fast crowds on ps3. In *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, pages 113–121. ACM, 2006. 1

[21] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1472–1485, 2009. 3, 5

[22] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 1

[23] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1160–1168. ACM, 2006. 1, 3

[24] J. van den Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin. Interactive navigation of multiple agents in crowded environments. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 139–147. ACM, 2008. 1

[25] X. Wang, K. T. Ma, G.-W. Ng, and W. Grimson. Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008. 4, 5

[26] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *International journal of computer vision*, 95(3):287–312, 2011. 1, 4

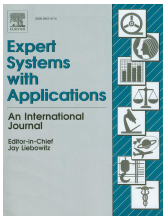
[27] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):539–555, 2009. 1, 4

[28] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2871–2878, 2012. 4, 5

Appendix A

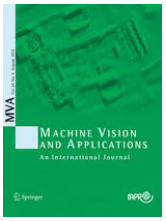
Publications and other mentions

JCR Journals



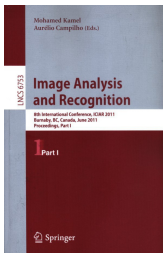
B. Cancela, M. Ortega, A. Fernández, M. G. Penedo. Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios. Expert Systems with Applications, 40(4), 1116-1131, 2013.

B. Cancela, M. Ortega, M. G. Penedo, J. Novo, N. Barreira. On the Use of a Minimal Path Approach for Target Trajectory Analysis. Pattern Recognition, 46(7), 2015-2027, 2013.

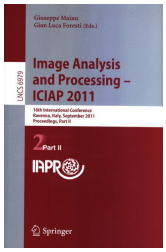


B. Cancela, M. Ortega, M. G. Penedo. Multiple Human Tracking System for Unpredictable Trajectories. Machine Vision and Applications, 25(2), 511-527, 2014.

Chapters in Book Series

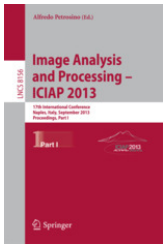


B. Cancela, M. Ortega, M. G. Penedo, A. Fernández. Solving Multiple-Target Tracking Using Adaptive Filters. Lecture Notes in Computer Science: International Conference on Image Analysis and Recognition (ICIAR), 6753, 416-425, 2011.



B. Cancela, M. Ortega, A. Fernández, M. G. Penedo. Path analysis in multiple-target video sequences. Lecture Notes in Computer Science: International Conference on Image Analysis and Processing (ICIAP) , 6979, 50-59, 2011.

B. Cancela, M. Ortega, M. G. Penedo. Path Analysis Using Directional Forces. A Practical Case: Traffic Scenes. Lecture Notes in Computer Science: Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) , 7887, 366-373, 2013.



B. Cancela, M. Ortega, A. Fernández, M. G. Penedo. Trajectory Similarity Measures using Minimal Paths. Lecture Notes in Computer Science: (ICIAP), 8156, 400-409, 2013.

International Conferences

B. Cancela, M. Ortega, M. G. Penedo. Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths. Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, 2014.

B. Cancela, T. Hospedales, S. Gong. Open-world Person Re-Identification by Multi-Label Assignment Inference. British Machine Vision Conference (BMVC), Nottimgham, UK, 2014.

Appendix B

Resumen

Durante los últimos años, el análisis de comportamiento humano es un área en constante expansión. La proliferación de cámaras en la sociedad aumentan exponencialmente las posibilidades de uso de este tipo de técnicas. Por poner el ejemplo más paradigmático, facilitan las tareas de seguridad. No obstante, el aumento de la información que se le proporciona a un operario de seguridad también conlleva una mayor dificultad a la hora de localizar comportamientos que puedan ser catalogados como sospechosos. Por tanto, se necesita de la creación de herramientas automáticas que sirvan de apoyo al operario para facilitar su labor.

Uno de estos sistemas de ayuda es la detección automática de comportamiento en secuencias de vídeo. Por seguir con el mismo ejemplo de la vigilancia, el sistema se encargaría de reducir toda la información de vídeo que recibe el operario a un pequeño número de movimientos sospechosos. No obstante, queda pendiente definir qué se considera como un movimiento sospechoso. Para una persona, esto se reduce a la detección de ciertos comportamientos que se salen de la norma, aquellos que son poco usuales.

Podríamos concluir que una persona, ante el comportamiento que realiza otro sujeto, es capaz de identificar rápidamente si se trata de un comportamiento normal o, por el contrario, si es un comportamiento sospechoso. No obstante, ¿qué pasaría si, en lugar de una persona, estamos presenciando una multitud? ¿Cómo distinguir, de entre todas las personas que estamos viendo, cuales son aquellas que realizan un comportamiento extraño? Hay que tener en cuenta que, en una escena concurrida, es muy difícil visualizar el cuerpo completo de una persona sin que existan otros sujetos que interfieran, colocándose entre la cámara y el susodicho. Por tanto, la tarea de identificación se tiene que realizar en un nivel más abstracto, como por ejemplo, el análisis de las trayectorias realizadas por las personas.

B.1 Análisis Automático de Comportamiento de Personas

Todo sistema automático de análisis de comportamiento de personas puede, de una manera ideal, ser dividido en tres módulos:

- *Detección:* Usando una secuencia de vídeo como entrada, el módulo de detección se encargaría de detectar a toda persona que aparezca durante todo el tiempo que aparezca.
- *Seguimiento:* Partiendo de la información proporcionada por el módulo de detección, el módulo de seguimiento se encarga de distinguir unívocamente a todas las personas que aparezcan en el vídeo. Esto significa agrupar todas las detecciones que pertenezcan a la misma persona bajo un único identificador.
- *Análisis de comportamiento de alto nivel:* Este módulo utiliza la información obtenida por los dos módulos anteriores para catalogar el comportamiento de cada sujeto.

Detección: Es un campo que lleva muchos años ofreciendo múltiples soluciones. Sin embargo, aún persiste un grave problema: la relación entre la calidad de la detección y el coste computacional. Como modelo más básico nos encontramos con la sustracción de fondo. La idea es simple: se realiza una estimación de cómo es el fondo de la escena (objetos inamovibles) y luego se compara con el frame del vídeo que se esté evaluando. Los píxeles que no se asemejen a lo esperado son considerados como parte de un objeto de interés. Finalmente, se agrupan dichos píxeles resaltados, dando a lugar a las detecciones.

Como se puede observar, se trata de un método muy rápido. Sólo requiere de una pasada sobre todos los píxeles de la imagen, para posteriormente agrupar todos los píxeles que no forman parte del fondo. No obstante, contiene muchas limitaciones. La más importante, cuando hablamos de entornos concurridos, es que no es capaz de distinguir entre dos personas si estas están solapadas. Las cataloga como un único ente indivisible. Por tanto, se requiere de técnicas de más alto nivel para poder hacer esta distinción.

Otra técnica muy utilizada es el flujo óptico. Partiendo de un frame del vídeo dado, se trata de localizar distintos patrones característicos, que tratarían de ser localizados en el frame siguiente. Dichos patrones puede ser desde niveles de gris de un píxel a texturas complejas. Lo más utilizado son los detectores de bordes o esquinas, puestos que son simples, rápidos de ejecución y obtienen buenos resultados.

Esta técnica introduce una ventaja con respecto a la substracción de fondo: nos indica la dirección del movimiento. Por tanto, el sistema es capaz de discernir personas que se solapan, pero que se mueven en direcciones contrarias. Sin embargo, en el caso que su movimiento sea en la misma dirección, nos encontramos con el mismo problema. Por tanto, aún se debe subir un peldaño más en el nivel de las técnicas.

Lo más fiable para detecciones de personas es la búsqueda de texturas comunes a todas ellas. En el caso de entornos concurridos, esto se reduce a la detección de cabezas, o, en su defecto, la detección del patrón creado por la cabeza y los hombros de una persona. Para este cometido se utilizan técnicas como el Histograma de Gradientes (HoG) en combinación con un clasificador tal como la máquina de vectores soporte (SVM) o el clasificador de Viola-Jones. No obstante, el coste computacional se incrementa hasta tal punto que para estas técnicas no resulta rentable ejecutarlas sobre toda la imagen. Se necesitan técnicas adicionales que restrinjan las regiones de uso, para así poder usarlas en sistemas de tiempo real.

Seguimiento: Para la realización del seguimiento se necesita de una métrica que mida cuan similares son dos detecciones en dos instantes de tiempo diferentes. La complicación viene en qué es lo que se debe medir. Una idea inicial podría ser cuánto se parecen las dos detecciones en términos de apariencia: color, textura, etc. Sin embargo, no siempre puede ser utilizada. Volviendo al ejemplo anterior de los entornos concurridos, la mayoría de las personas se suele detectar parcialmente ocluida, haciendo muy complicada una comparación sobre la apariencia.

Por tanto, se requiere el uso de modelos más básicos, centrados en características espaciales. Por ejemplo, la distancia entre dos detecciones. Sin embargo, ¿qué pasa si lo que intentas comparar son dos detecciones en dos instantes de tiempo separados en varios frames? La posición del sujeto puede haber cambiado significativamente, y el cálculo de la distancia puede inducir a error.

Para paliar esta deficiencia se desarrollaron los métodos de predicción. Su cometido es el siguiente: sabiendo la posición y velocidad de una persona en un instante de tiempo, ¿puedo predecir en dónde se va a encontrar dicha persona posteriormente? El método más utilizado para la predicción de objetos es el filtro de Kalman. Se trata de un estimador linear cuadrático que es capaz de realizar predicciones robustas ante información con ruido. Sin embargo, presenta un contratiempo: sólo devuelve una predicción, un lugar concreto, con lo que presenta muchos fallos ante oclusiones, esto es, cuando dejamos de localizar a una persona durante un largo tiempo.

Haciendo uso de filtros lineales tipo Kalman, se pueden extender lanzando una serie de predicciones a las que se le introduce ruido aleatorio, dando como resultado unas muestras que llamaremos partículas. Dichas muestras serían evaluadas con el objetivo de buscar cuál es el lugar más probable en dónde se encuentra el sujeto. Esta generalización es lo que se conoce como filtros de partículas. Su inconveniente, otra vez, es el coste computacional: usar pocas partículas hace que el sistema tenga los mismos inconvenientes que el filtro lineal, mientras que usar muchas aumenta sobremanera el tiempo de ejecución del sistema, haciéndolo inviable para su utilización en entornos donde se requieren sistemas de tiempo real.

Análisis de comportamiento: Centrándonos en trayectorias, el análisis del comportamiento se centra en la detección de caminos y en su catalogación en usuales o erráticos. Con esto como base, la mayoría de las aproximaciones que tratan de resolver este problema se centran en el *clustering*: el agrupamiento de trayectorias similares para formar un conjunto finito de las denominadas 'trayectorias usuales'. Cualquier trayectoria que no sea similar a alguna de las incluidas en este conjunto será catalogada como errática.

Se consideran tres maneras distintas para la definición de los caminos usuales. En primer lugar, basada en *centroides*. Para ello, se agrupan todas las trayectorias similares en una única trayectoria, entendiendo como tal una sucesión de puntos conexos linealmente. Este método se extendió para dar lugar a las técnicas de *envoltura*. En ellas, se substituye la trayectoria única por una región espacial por la que pasan, en mayor o menor medida, todos las trayectorias contenidas en ese agrupamiento. Para ello se utilizan desde conjuntos de Gaussianas hasta regiones de interés. Finalmente, también existen las técnicas basadas en *tramos*. Dado que varios conjuntos de trayectorias pueden compartir algunas secciones de sus caminos, se dividen todas las trayectorias en pequeñas tramas y se conectan entre ellas mediante grafos, formando de esta manera las trayectorias completas.

Recientemente está emergiendo una nueva vertiente para el análisis de comportamiento en entornos concurridos. Son lo que se ha dado en llamar los modelos de *fuerza social*. Se centran en revisar el comportamiento de las personas en base a cómo interacciona con el entorno y con el resto de personas. Dichos sistemas introducen una serie de ecuaciones, una serie de fuerzas, que, en su conjunto, tratan de determinar el comportamiento general en la escena.

B.2 Tesis

El objetivo de esta introducción es el desarrollo de un novedoso sistema totalmente automático para el análisis de comportamiento en escenas concurridas. No existe ninguna técnica en la literatura existente que pueda, simultáneamente, responder a todos los problemas de este tipo de sistemas. Los objetivos específicos de esta tesis son:

1. Introducción de nuevas metodologías para el seguimiento de multitudes que pueda solucionar, total o parcialmente, el mayor problema que acecha a estos sistemas: la relación entre la precisión y el coste computacional.

2. Definir una serie de hipótesis fijas que expliquen, de una manera sencilla, el significado de trayectorias 'usuales'.

3. Definición, en base a las hipótesis previamente definidas, de un sistema novedoso para el análisis de trayectorias humanas. Su cometido será catalogar si una persona está realizando un comportamiento usual o errático. Para ello, se hará uso de información de la escena.

4. Introducir mejoras en el sistema para que pueda ser utilizado para catalogar el comportamiento de cualquier tipo de objeto.

5. Mejorar el coste computacional lo máximo posible, con objeto de que pueda ser utilizado en entornos concurridos.

Dado que se trata de una tesis defendida por compendio de publicaciones, a continuación se resumirán, paper por paper, las aportaciones introducidas en este trabajo.

Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios: En este trabajo se propone la creación de un sistema automático de seguimiento de objetos, con objeto de hacerlo lo más rápido posible, esto es, que pueda ser ejecutado en tiempo real en un ordenador estándar. Para ello se hace uso de una estructura jerárquica, contando con dos módulos independientes que realizan la tarea de seguimiento, uno de bajo nivel para las detecciones entre frames consecutivos o próximos entre si, y otro de más alto nivel para la recuperación de la identificación de un sujeto si se pierde su localización durante un tiempo relativamente largo.

Con objeto de acelerar todo el proceso, se usaron técnicas de poco consumo de procesador: sustracción de fondo para la detección de objetos, filtros adaptativos

(Adalines) en lugar de filtros de Kalman para labores de predicción, representación del objetos mediantes elipses para el módulo de bajo nivel y con histogramas para el de alto nivel. La representación de elipses se utiliza, a mayores, para la detección de colisiones entre distintos objetos, creado la figura de agrupamientos y divisiones, que el sistema se encarga de resolver.

Está metodología se probó en dos bases de datos públicas (CAVIAR y CANDELA) obteniendo resultados muy prometedores. El sistema es capaz de funcionar en tiempo real en una máquina cualquiera, manteniendo una precisión muy alta en el seguimiento de objetos. No obstante, el uso de la sustracción de fondo limita el uso de esta metodología a entornos poco concurridos.

Multiple Human Tracking System for Unpredictable Trajectories: Con el objetivo de eliminar el problema principal de la aproximación anterior, se desarrolló una segunda metodología, esta vez ya centrada en el dominio del seguimiento de personas. Para ello, se combinaron dos técnicas de alto nivel, el clasificador de Viola-Jones y un SVM entrenado para histogramas de gradiente centrados en el patrón de omega que genera la figura de la cabeza y los hombros. La sustracción de fondo continúa siendo utilizada, pero únicamente para restringir los lugares en donde aplicar estos dos clasificadores. Los resultados experimentales muestran que su uso mejora tanto la calidad de la salida proporcionada como su uso de CPU, incrementando el número de frames por segundo que puede procesar esta metodología.

Asimismo, se diseñó un nuevo sistema para la re-identificación de personas, teniendo en cuenta las oclusiones que se producen en entornos muy concurridos. Para evaluar la metodología, se grabó un vídeo de un evento deportivo, en el que las trayectorias de las personas son totalmente impredecibles. Los resultados muestran que nuestro método es capaz de seguir a múltiples personas en entornos concurridos e impredecibles. Sin embargo, su coste computacional hace que no pueda ser ejecutada en tiempo real en un ordenador cualquiera.

Open-world person re-identification by multi-label assignment Inference: Uno de los problemas más graves de los sistemas de seguimiento es su incapacidad de recuperar la identificación de un objeto ante largas oclusiones. Es lo que se llama el problema de 're-identificación'. Las técnicas clásicas basadas en este problema se modulan como un problema de recuperación: se parte de un conjunto de detecciones conocidas; ante una nueva detección, se busca cuál es la detección que más se asemeja. Desafortunadamente, esto no es lo que ocurre en sistemas reales, donde no hay información a priori sobre todas las personas que pueden aparecer en el vídeo. Por tanto, en este trabajo se propuso, por primera vez, el uso de técnicas de re-

identificación en *mundo abierto*, donde no se contiene de antemano información de todos los sujetos que pueden aparecer en las distintas cámaras que pueda tener el entorno a evaluar.

Para ello, se ha hecho uso de un campo condicional aleatorio (CRF) ejecutado en dos pasos. En el primero, sólo se comparan detecciones que ocurren en la misma cámara. Posteriormente, la salida del CRF se toma como entrada del segundo paso, donde se permiten conexiones entre distintas cámaras, pero de una manera restringida. Esto se debe a que los cambios de iluminación y de orientación entre cámaras dificultan la detección. De esta manera se refuerzan las conexiones entre detenciones tomadas con la misma cámara en detrimento del resto. Los resultados obtenidos en una base de datos pública (SoftBIO) muestran que el uso del CRF incrementa sustancialmente el desempeño del sistema.

On the Use of a Minimal Path Approach for Target Trajectory Analysis: Una vez acometido el problema del seguimiento de personas, nos centramos en el análisis de comportamiento. Para ello, centrándonos en el análisis de trayectorias, nuestro cometido fue el de obtener un método que reuniera unas ciertas características: (i) poder ser usado sin ningún tipo de entrenamiento previo; (ii) que pudiera ser actualizado conforme le va llegando más información; (iii) que limitase el consumo de memoria del sistema; (iv) que fuera capaz de funcionar aún cuando el sistema de seguimiento no proporciona una información precisa; y (v) que pudiera evaluar el comportamiento en cualquier momento, sin necesidad de esperar a que la trayectoria de la persona se complete.

Para ello, buscamos una aproximación que pudiera ser lo más simple posible. Para ello, establecimos una serie de hipótesis para explicar lo que se entiende como 'comportamiento usual'. La primera es que asumimos que toda persona que aparece en la escena tiene intención de llegar a un lugar determinado, esté dentro o fuera de la escena. Por tanto, un comportamiento tal como el de deambular es considerado como anómalo. La segunda hipótesis nos dice que el camino que toma una persona para llegar a dicho lugar es el que suele tomar el resto de gente.

Con estas premisas, desarrollamos un método para la detección del camino usual mediante la utilización de caminos mínimos. Mediante un campo de frecuencias de paso de la gente, se crea una superficie que nos permite extraer el camino mínimo desde cualquier punto de la escena a un punto inicial, que normalmente será tomada como la primera detección tomada de dicha persona. Una vez obtenido el camino usual, se compara con la trayectoria real tomada por la persona para evaluar su comportamiento. Dicha trayectoria puede ser evaluada en cualquier momento, no hay que esperar a que el objeto deje la escena para obtener una estimación de su

comportamiento.

El campo de frecuencias puede ser iniciado a cero, con lo que el sistema es capaz de funcionar sin entrenamiento. Aún así, éste se le puede añadir si se desea. Asimismo, se pueden utilizar distintos potenciales, dependiendo, por ejemplo, de dónde empiece la trayectoria del objeto a evaluar, o de su tipo. La memoria utilizada se mantiene constante, puesto que el sistema únicamente necesita mantener el campo de frecuencias. Puede funcionar con fragmentos de trayectorias, puesto que el campo de densidades solo almacena información local de cada punto, y ninguna sobre cómo se conecta con otros lugares. Es el algoritmo de caminos mínimos el que se encarga de usar información local para convertirla en un análisis global.

Para evaluar las dos trayectorias, la usual y la real, se construyó un mapa de distancias para ver cuán lejos está la trayectoria real de la esperada. Se evaluó esta técnica en una base de datos creada para la ocasión, puesto que no existe ninguna base de datos anotada con comportamiento anómalo. Se comparó nuestra métrica contra las existentes en la literatura, probando que nuestro método es mucho más eficaz.

Path Analysis Using Directional Forces. A Practical Case: Traffic Scenes: Una vez probado que nuestro método es capaz de catalogar correctamente el comportamiento humano, nos propusimos extenderlo a cualquier tipo de objeto a seguir. El comportamiento de casi todos los objetos sigue las mismas reglas que las personas, con algunas excepciones. Dichas excepciones tienen que ver con comportamientos en los cuáles la dirección del movimiento es importante. El ejemplo más ilustrativo es el del tráfico. No es lo mismo ir por un carril en una dirección u en otra, el comportamiento cambia sustancialmente.

El método de caminos mínimos desarrollado no tiene en cuenta la dirección, haciéndolo inviable para ser usado en este tipo de entornos. Para subsanar esto, utilizamos un algoritmo más complejo para la obtención del camino mínimo. Este algoritmo (*Ordered Upwind Method* (OUM)) tiene en cuenta la orientación del movimiento. Para comprobar su rendimiento, se probó en una situación compleja: una rotonda. Si no se tiene en cuenta la dirección del carril no sería necesario dar una vuelta a la rotonda para coger la salida de la izquierda, sólo habría que tomar la rotonda en sentido hacia la izquierda. El nuevo algoritmo es capaz de sobreponerse a esta casuística, obteniendo el resultado esperado para una trayectoria usual.

Trajectory Similarity Measures Using Minimal Paths: Uno de los problemas con los que se encontraba nuestra metodología era el coste computacional del cálculo de la métrica ($\mathcal{O}(N \log N)$, siendo N el numero de nodos del mapa de

densidades). Así que nos propusimos mejorar este punto.

En lugar de crear un mapa de distancias desde el camino usual, lo creamos desde el punto inicial de la trayectoria. La diferencia es que este cálculo se puede realizar al mismo tiempo que para la obtención del camino mínimo. Para una trayectoria usual, la distancia real recorrida es similar a la distancia estimada. A tal efecto, se crearon diferentes técnicas para explotar esta nueva información. Los resultados mostraron que la métrica se comportaba de una manera similar (algo inferior) al mapa de distancias definido previamente. No obstante, la complejidad pasó de $\mathcal{O}(N \log N)$ a $\mathcal{O}(1)$, haciendo que esta métrica sea más útil para su utilización en entornos concurridos.

Unsupervised Trajectory Modelling using Temporal Information via Minimal Paths: Por último, nos propusimos evaluar la metodología en un entorno concurrido. Para ello, además del análisis de trayectorias basado en componentes espaciales, incluimos a mayores la velocidad. De la misma manera que el mapa de distancia creado en la aproximación anterior, ahora creamos a mayores un mapa de tiempos. Básicamente, nos devuelve una superficie que nos indica el tiempo necesario que le llevaría a una persona alcanzar cualquier punto de la escena, siempre y cuando su comportamiento sea usual. A mayores, introducimos direccionalidad en el campo de densidades definiendo cuatro direcciones (los cuatro puntos cardinales).

A la hora de comprobar el comportamiento de la metodología, nos encontramos con el mismo problema mencionado anteriormente: no existe una base de datos que contenga información de comportamientos anómalos. Por tanto, recurrimos a un análisis estadístico. Asumiendo que definimos como comportamiento usual aquel que es realizado por la mayoría de las personas, la métrica que utilicemos tendrá que indicar que la mayoría de la gente realiza un comportamiento correcto. Cuanto menos usual sea el comportamiento, más difícil es que éste aparezca. Al evaluar la métrica en dos bases de datos públicas (un parking y una estación de metro) nos encontramos que la métrica se comportaba de la manera esperada. Lo que indica que nuestra métrica no se ve influenciada por las interacciones que se producen entre personas en entornos altamente concurridos.

B.3 Conclusiones

En esta tesis se ha abordado la creación de un sistema automático de análisis de comportamiento humano. Para tal efecto, se han propuesto dos modelos de seguimiento de personas, con objeto de mejorar el delicado balance entre la precisión de dicho seguimiento y su coste computacional. Los resultados muestran la eficacia de dichos

métodos en entornos no controlados. Asimismo, se introdujo por primera vez el concepto de re-identificación en mundo abierto, proveyendo una primera aproximación a su resolución mediante la utilización de un campo aleatorio condicional.

Con respecto al análisis del comportamiento humano, se ha desarrollado una tecnología completamente novedosa basada en el análisis de caminos mínimos. Parte de la premisa de que toda persona intenta alcanzar un objetivo concreto de la escena, utilizando para ello el camino más comúnmente utilizado. Dicha aproximación fue extendida para poder ser utilizada con cualquier tipo de análisis de objetos (tráfico incluido). Posteriormente, se redujo la complejidad del algoritmo mientras se mantenía una calidad de la solución similar, para finalmente probar cómo esta metodología es capaz de analizar el comportamiento humano incluso en entornos concurridos, tales como una estación de metro.

Como trabajo futuro se plantea la implementación de los algoritmos de seguimiento mediante GPUs, con objeto de hacer posible que funcionen en tiempo real. Asimismo, se propone de la creación de un nuevo método para la obtención del camino mínimo con información direccional, que reduzca la complejidad de los métodos ya existentes.

Finalmente, se plantea la cuestión del *seguimiento basado en comportamiento*, esto es, realimentar los modelos de seguimiento con la información de comportamiento, con objeto de mejorar su precisión. Asimismo, se propone la creación de un método de predicción a larga distancia. Dicho de otro modo, un sistema de alto nivel que sustituya a los filtros lineales tipo Kalman.

Bibliography

Andriyenko, A., & Schindler, K. (2011). Multi-target tracking by continuous energy minimization. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 1265–1272).

Andriyenko, A., Schindler, K., & Roth, S. (2012). Discrete-continuous optimization for multi-target tracking. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on* (pp. 1926–1933).

BARD, behavioral analysis and recognition dataset. (Date accessed: february, 2015).
http://www.varpa.org/bard/.

Bashir, F. I., Khokhar, A. A., & Schonfeld, D. (2007). Object trajectory-based activity classification and recognition using hidden markov models. *Image Processing, IEEE Transactions on*, *16*(7), 1912–1919.

Benfold, B., & Reid, I. (2011). Stable multi-target tracking in real-time surveillance video. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 3457–3464).

Ben Shitrit, H., Berclaz, J., Fleuret, F., & Fua, P. (2011). Tracking multiple people under global appearance constraints. In *Computer vision (iccv), 2011 ieee international conference on* (pp. 137–144).

Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhyā: The Indian Journal of Statistics*, 401–406.

Bialkowski, A., Denman, S., Lucey, P., Sridharan, S., & Fookes, C. B. (2012). A database for person re-identification in multi-camera surveillance networks. In *Proceedings of the 2012 international conference on digital image computing techniques and applications (dicta 12)* (pp. 1–8).

Biliotti, D., Antonini, G., & Thiran, J. P. (2005). Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences. In *Proceedings of the ieee workshop on motion and video computing (wacv/motion'05)* (Vol. 2, pp. 50–57). IEEE.

Black, J., Ellis, T., & Rosin, P. (2002). Multi view image surveillance and tracking. In *Motion and video computing, 2002. proceedings. workshop on* (pp. 169–174).

Bose, B., Wang, X., & Grimson, E. (2007). Multi-class object tracking algorithm that handles fragmentation and grouping. In *Computer vision and pattern recognition, 2007. cvpr'07. ieee conference on* (pp. 1–8).

Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, *5*.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., & Van Gool, L. (2011). Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *33*(9), 1820–1833.

Brox, T., & Malik, J. (2011). Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *33*(3), 500–513.

Burstedde, C., Klauck, K., Schadschneider, A., & Zittartz, J. (2001). Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, *295*(3), 507–525.

Buzan, D., Sclaroff, S., & Kollios, G. (2004). Extraction and clustering of motion trajectories in video. In *Pattern recognition, 2004. icpr 2004. proceedings of the 17th international conference on* (Vol. 2, pp. 521–524).

Cancela, B., Hospedales, T. M., & Gong, S. (2014). Open-world person re-identification by multi-label assignment inference. *British Machine Vision Conference*.

Cancela, B., Iglesias, A., Ortega, M., & Penedo, M. (2014). Unsupervised trajectory modelling using temporal information via minimal paths. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2553–2560).

Cancela, B., Ortega, M., Fernández, A., & Penedo, M. G. (2011). Path analysis in multiple-target video sequences. In *Image analysis and processing–iciap 2011* (pp. 50–59). Springer.

Cancela, B., Ortega, M., Fernández, A., & Penedo, M. G. (2012). Hierarchical framework for robust and fast multiple-target tracking in surveillance scenarios. *Expert Systems with Applications*, *40*(4), 1116–1131.

Cancela, B., Ortega, M., Fernández, A., & Penedo, M. G. (2013). Trajectory similarity measures using minimal paths. In *Image analysis and processing–iciap 2013* (pp. 400–409). Springer Berlin Heidelberg.

Cancela, B., Ortega, M., Penedo, M., Novo, J., & Barreira, N. (2013). On the use of a minimal path approach for target trajectory analysis. *Pattern Recognition*, *46*(7), 2015–2027.

Cancela, B., Ortega, M., & Penedo, M. G. (2014). Multiple human tracking system for unpredictable trajectories. *Machine vision and applications*, *25*(2), 511–527.

Cancela, B., Ortega, M., Penedo, M. G., & Fernández, A. (2011). Solving multiple-target tracking using adaptive filters. In *Image analysis and recognition* (pp. 416–425). Springer Berlin Heidelberg.

Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., & Kasturi, R. (2010). Understanding transit scenes: A survey on human behavior-recognition algorithms. *Intelligent Transportation Systems, IEEE Transactions on*, *11*(1), 206–224.

CANDELA, content analysis and networked delivery architectures. (Date accessed: february, 2015). http://www.multitel.be/∼va/candela/.

CAVIAR, context aware vision using image-based active recognition. (Date accessed: february, 2015). http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

Cohen, L. D., & Kimmel, R. (1997). Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision*, *24*, 57-78.

Collins, R., Liu, Y., & Leordeanu, M. (2005). Online selection of discriminative tracking features. *PAMI*, *27*(10), 1631-1643.

Comaniciu, D., Ramesh, V., & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Computer vision and pattern recognition, 2000. proceedings. ieee conference on* (Vol. 2, p. 142 -149 vol.2).

Cucchiara, R., Grana, C., Piccardi, M., Prati, A., & Sirotti, S. (2001). Improving shadow suppression in moving object detection with hsv color information. In *Intelligent transportation systems, 2001. proceedings. 2001 ieee* (p. 334-339).

Dalal, N., & Triggs, B. (2005, june). Histograms of oriented gradients for human detection. In *Computer vision and pattern recognition, 2005. cvpr 2005. ieee computer society conference on* (Vol. 1, p. 886 -893).

Desai, C., Ramanan, D., & Fowlkes, C. (2009, oct). Discriminative models for multi-class object layout. In *Computer vision, 2009 ieee 12th international conference on* (p. 229 -236).

Doshi, A., & Trivedi, M. (2006, November). "hybrid cone-cylinder" codebook model for foreground detection with shadow and highlight suppression. In *Video and signal based surveillance, 2006. avss '06. ieee international conference on* (p. 19).

Farenzena, M., Bazzani, L., Perina, A., Murino, V., & Cristani, M. (2010). Person re-identification by symmetry-driven accumulation of local features. In *Computer vision and pattern recognition (cvpr), 2010 ieee conference on* (pp. 2360–2367).

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Image analysis* (pp. 363–370). Springer.

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010, sept.). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*(9), 1627 -1645.

Fleet, D. J., & Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, *5*(1), 77–104.

Gong, S., Cristani, M., Yan, S., & Loy, C. C. (2014). *Person re-identification*. Springer.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., & Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, *50*(2), 425–437.

Haritaoglu, I., Harwood, D., & Davis, L. (1998, April). W4: Who? when? where? what? a real time system for detecting and tracking people. In *Automatic face and gesture recognition, 1998. proceedings. third ieee international conference on* (p. 222-227).

Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W 4: Real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *22*(8), 809–830.

Heeger, D. J. (1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, *1*(4), 279–302.

Helbing, D., Farkas, I., & Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, *407*(6803), 487–490.

Hirzer, M., Roth, P. M., Köstinger, M., & Bischof, H. (2012). Relaxed pairwise learned metric for person re-identification. In *Computer vision–eccv 2012* (pp. 780–793). Springer.

Hlinka, O., Sluciak, O., Hlawatsch, F., Djuric, P. M., & Rupp, M. (2012). Likelihood consensus and its application to distributed particle filtering. *Signal Processing, IEEE Transactions on*, *60*(8), 4334–4349.

Horprasert, T., Hardwood, D., & Davis, L. S. (2000). A robust background substraction and shadow detection. In *4th accv, taipei, taiwan* (Vol. 1, p. 34-41).

Hu, W., Xiao, X., Xie, D., Tan, T., & Maybank, S. (2004). Traffic accident prediction using 3-d model-based vehicle tracking. *Vehicular Technology, IEEE Transactions on*, *53*(3), 677–694.

Hu, W., Xie, D., Fu, Z., Zeng, W., & Maybank, S. (2007). Semantic-based surveillance video retrieval. *Image Processing, IEEE Transactions on*, *16*(4), 1168–1181.

Huang, C., Wu, B., & Nevatia, R. (2008). Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th european conference on computer vision: Part ii* (pp. 788–801). Berlin, Heidelberg: Springer-Verlag.

Iwase, S., & Saito, H. (2004). Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *Pattern recognition, 2004. icpr 2004. proceedings of the 17th international conference on* (Vol. 4, pp. 751–754).

Jepson, A., Fleet, D., & El-Maraghi, T. (2003, October). Robust online appearance models for visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *25*(10), 1296-1311.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, *82*(1), 35–45.

Karaman, S., & Bagdanov, A. D. (2012). Identity inference: generalizing person re-identification scenarios. In *Computer vision–eccv 2012. workshops and demonstrations* (pp. 443–452).

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75.

Keogh, E., & Pazzani, M. (2000). Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth acm sigkdd international conference on knowledge discovery and data mining* (pp. 285–289).

Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, *11*(3), 172 - 185. (Special Issue on Video Object Processing)

Kostinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., & Bischof, H. (2012). Large scale metric learning from equivalence constraints. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on* (pp. 2288–2295).

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, *2*(1-2), 83–97.

Li, M., Zhang, Z., Huang, K., & Tan, T. (2008, dec.). Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *Pattern recognition, 2008. icpr 2008. 19th international conference on* (p. 1 -4).

Li, M., Zhang, Z., Huang, K., & Tan, T. (2009). Rapid and robust human detection and tracking based on omega-shape features. In *16th ieee international conference on image processing (icip)* (p. 2545 - 2548).

Li, X., Hu, W., & Hu, W. (2006). A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Pattern recognition, 2006. icpr 2006. 18th international conference on* (Vol. 1, p. 591 -594).

Li, Y., Huang, C., & Nevatia, R. (2009, june). Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (p. 2953 -2960).

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *Ijcai* (Vol. 81, pp. 674–679).

Luo, W., Zhao, X., & Kim, T.-K. (2014). Multiple object tracking: A review. *ArXiv e-prints*.

Maddalena, L., & Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on*, *17*(7), 1168 -1177.

Magee, D. R. (2004). Tracking multiple vehicles using foreground, background and motion models. *Image and vision Computing*, *22*(2), 143–155.

Makris, D., & Ellis, T. (2005). Learning semantic scene models from observing activity in visual surveillance. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *35*(3), 397–408.

Mehran, R., Oyama, A., & Shah, M. (2009). Abnormal crowd behavior detection using social force model. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 935–942).

Min, C., & Medioni, G. (2008). Inferring segmented dense motion layers using 5d tensor voting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *30*(9), 1589–1602.

Mittal, A., & Davis, L. S. (2003). M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, *51*(3), 189–203.

Moore, B. E., Ali, S., Mehran, R., & Shah, M. (2011). Visual crowd surveillance through a hydrodynamics lens. *Communications of the ACM*, *54*(12), 64–73.

Morris, B., & Trivedi, M. M. (2008). An adaptive scene description for activity analysis in surveillance video. In *Icpr* (p. 1-4). IEEE.

Morris, B., & Trivedi, M. M. (2009). Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Cvpr* (p. 312-319). IEEE.

Morris, B. T., & Trivedi, M. M. (2008). Learning, modeling, and classification of vehicle track patterns from live video. *Intelligent Transportation Systems, IEEE Transactions on*, *9*(3), 425–437.

Morris, B. T., & Trivedi, M. M. (2011). Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *33*(11), 2287–2301.

Naftel, A., & Khalid, S. (2006). Motion trajectory learning in the dft-coefficient feature space. In *Computer vision systems, 2006 icvs'06. ieee international conference on* (pp. 47–47).

Pellegrini, S., Ess, A., Schindler, K., & Van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer vision, 2009 ieee 12th international conference on* (pp. 261–268).

Piciarelli, C., & Foresti, G. L. (2006). On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, *27*(15), 1835–1842.

Popoola, O. P., & Wang, K. (2012). Video-based abnormal human behavior recognition–a review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *42*(6), 865–878.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and vision computing*, *28*(6), 976–990.

Prosser, B., Zheng, W.-S., Gong, S., Xiang, T., & Mary, Q. (2010). Person re-identification by support vector ranking. In *Bmvc* (Vol. 2, p. 6).

Rashwan, H. A., García, M. A., & Puig, D. (2013). Variational optical flow estimation based on stick tensor voting. *Image Processing, IEEE Transactions on*, *22*(7), 2589–2599.

Reynolds, C. (2006). Big fast crowds on ps3. In *Proceedings of the 2006 acm siggraph symposium on videogames* (pp. 113–121).

Rodriguez, M., Sivic, J., Laptev, I., & Audibert, J.-Y. (2011). Density-aware person detection and tracking in crowds. In *Proceedings of the international conference on computer vision (iccv).*

Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(4), 1591-1595.

Sethian, J. A., & Vladimirsky, A. (2003). Ordered upwind methods for static hamilton–jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis*, *41*(1), 325–363.

Shi, J., & Tomasi, C. (1994). Good features to track. In *Computer vision and pattern recognition, 1994. proceedings cvpr'94., 1994 ieee computer society conference on* (pp. 593–600).

Shizawa, M., & Maze, K. (1991). Unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *Computer vision and pattern recognition, 1991. proceedings cvpr'91., ieee computer society conference on* (pp. 289–295).

Song, B., Jeng, T.-Y., Staudt, E., & Roy-Chowdhury, A. (2010). A stochastic graph evolution framework for robust multi-target tracking. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision â eccv 2010* (Vol. 6311, p. 605-619). Springer Berlin / Heidelberg.

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Ieee computer society conference on computer vision and pattern recognition* (Vol. 2, p. 246-252).

Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D., & Soundararajan, P. (2007). The clear 2006 evaluation. In *Multimodal technologies for perception of humans* (pp. 1–44). Springer.

Tao, D., Jin, L., Wang, Y., Yuan, Y., & Li, X. (2013). Person re-identification by regularized smoothing kiss metric learning. *Circuits and Systems for Video Technology, IEEE Transactions on*, *23*(10), 1675–1685.

Treuille, A., Cooper, S., & Popović, Z. (2006). Continuum crowds. In *Acm transactions on graphics (tog)* (Vol. 25, pp. 1160–1168).

Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, *18*(11), 1473–1488.

van den Berg, J., Patil, S., Sewall, J., Manocha, D., & Lin, M. (2008). Interactive navigation of multiple agents in crowded environments. In *Proceedings of the 2008 symposium on interactive 3d graphics and games* (pp. 139–147).

Vezzani, R., Baltieri, D., & Cucchiara, R. (2013). People reidentification in surveillance and forensics: A survey. *ACM Computing Surveys (CSUR)*, *46*(2), 29.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer vision and pattern recognition, 2001. cvpr 2001. proceedings of the 2001 ieee computer society conference on* (Vol. 1, p. I-511 - I-518).

Wang, X., Ma, K. T., Ng, G.-W., & Grimson, W. (2008). Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In *Computer vision and pattern recognition, 2008. cvpr 2008. ieee conference on* (p. 1-8).

Wang, X., Ma, K. T., Ng, G.-W., & Grimson, W. E. L. (2011). Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *International journal of computer vision*, *95*(3), 287–312.

Wang, X., Ma, X., & Grimson, W. E. L. (2009). Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *31*(3), 539–555.

Wang, X., Tieu, K., & Grimson, E. (2006). Learning semantic scene models by trajectory analysis. In *Computer vision–eccv 2006* (pp. 110–123). Springer.

Weinland, D., Ronfard, R., & Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, *115*(2), 224–241.

Wu, B., & Nevatia, R. (2006, june). Tracking of multiple, partially occluded humans based on static body part detection. In *Computer vision and pattern recognition, 2006 ieee computer society conference on* (Vol. 1, p. 951 - 958).

Wu, B., & Nevatia, R. (2007, November). Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vision*, *75*, 247-266.

Xing, J., Ai, H., & Lao, S. (2009). Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 1200–1207).

Xu, X., Gong, S., & Hospedales, T. (2013). Cross-domain traffic scene understanding by motion model transfer. In *Proceedings of the 4th acm/ieee international workshop on analysis and retrieval of tracked events and motion in imagery stream* (pp. 77–86).

Yang, B., Huang, C., & Nevatia, R. (2011). Learning affinities and dependencies for multi-target tracking using a crf model. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 1233–1240).

Yang, B., & Nevatia, R. (2012). An online learned crf model for multi-target tracking. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on* (pp. 2034–2041).

Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., & Xu, L.-Q. (2008). Crowd analysis: a survey. *Machine Vision and Applications*, *19*(5-6), 345–357.

Zhang, L., Li, Y., & Nevatia, R. (2008, june). Global data association for multi-object tracking using network flows. In *Computer vision and pattern recognition, 2008. cvpr 2008. ieee conference on* (p. 1 -8).

Zheng, W.-S., Gong, S., & Xiang, T. (2011). Person re-identification by probabilistic relative distance comparison. In *Computer vision and pattern recognition (cvpr), 2011 ieee conference on* (pp. 649–656).

Zhou, B., Wang, X., & Tang, X. (2012). Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on* (p. 2871-2878).

Zhou, S. K., Chellappa, R., & Moghaddam, B. (2004). Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Processing, IEEE Transactions on*, *13*(11), 1491–1506.

Zimmer, H., Bruhn, A., & Weickert, J. (2011). Optic flow in harmony. *International Journal of Computer Vision*, *93*(3), 368–388.