Model-based Behavioural Tracking and Scale Invariant Features in Omnidirectional Matching

THÈSE N$^{\text{O}}$ 5150 (2011)

PRÉSENTÉE LE 30 SEPTEMBRE 2011

À LA FACULTÉ ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE TRANSPORT ET MOBILITÉ
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Javier CRUZ MOTA

acceptée sur proposition du jury:

Dr J.-M. Vesin, président du jury
Prof. M. Bierlaire, Prof. J.-Ph. Thiran, directeurs de thèse
Dr I. Bogdanova Vandergheynst, rapporteur
Prof. A. Cavallaro, rapporteur
Prof. B. Macq, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2011

Todo pasa y todo queda;
pero lo nuestro es pasar,
pasar haciendo caminos,
caminos sobre la mar.

Antonio Machado

A las personas que me acompañan en mi camino

Acknowledgments

Although this is a section usually placed at the beginning of a PhD Thesis, it is commonly the last section to be written. This is probably because summarising in a few lines all the gratitude to the people that directly or indirectly have contributed to a thesis, is a really hard task.

I would like to start by thanking Professor Michel Bierlaire and Professor Jean-Philippe Thiran for giving me the opportunity of doing this co-supervised PhD. They opened me the doors of the Transport and Mobility Laboratory (TRANSP-OR) and the Signal Processing Laboratory 5 (LTS5), generating a stimulating working environment that together with their ideas and insights have definitely enriched this thesis. Furthermore, during these years I have had the valuable support of the Swiss National Science Foundation, who granted my research under the project number 200021-117823.

Although there is usually only one name as author, a PhD Thesis is commonly the fruit of collaborations between several researchers, and my case was not an exception. In addition to the support that I have received from my two supervisors, I had the chance of collaborating with very good researchers. I would like to express my sincere gratitude to Dr. Matteo Sorci, Dr. Iva Bogdanova, Dr. Thomas Robin and Benoît Paquier. Without any doubt, their contributions improved the quality of this thesis.

By nature, a laboratory (and EPFL in general) is a place where lots of people pass by without staying for long time, from PhD students and Postdocs, who stay several years, to interns and exchange students, that only spend a few months here. This allowed me to meet a lot of interesting people that left their footprint on me and my work, and what is maybe more important, that made hours at the office (and outside) funny and pleasant. For these reasons, I would like to thank Anne, Antonin, Aurélie (aka Séverine), Bilal (aka Dr. B), Bilge, Carolina, Emma, Gunnar (aka Darth Vader), Ilaria, Jeffrey, Jingmin, Kamran, Mamy, Marianne, Matteo, Niklaus, Nitish, Olga (aka Olja), Olivier, Prem, Ricardo (aka Quiet man), Sohrab, Willem and Zachary, for the good moments at TRANSP-OR, at EPFL and at the outside World! But it is well known that the depth of a footprint depends on the weight of the person, and in this sense I have met two very “fat” persons: Michaël “Schumacher” Thémans and Thomas “Canard Français” Robin. Michaël was my first office-mate and although we only shared the office during a few months, it was enough for starting a very good friendship. Thank you Michaël for all the good moments...and all the good coffees! After “quitting” Michaël, I started to share the office with Thomas. I do not know if it was because we had a joint research project, because we lived very close, because we went running together, because we both had chairs with armrests (only allowed to professors at EPFL!) or because he drew me a lot of comic strips, but the fact is that his friendship and his personal and professional support has been crucial for being able to finish this thesis, thank you Thomas!

The support from people outside the lab is also very important. I would like to thank our neighbours from “La PPE”, for their kind welcome to Préverenges and the great barbecues in the garden. I am also particularly grateful to my “Famille Lausannoise”, a small (but exclusive!) group of Spanish people living in Lausanne that have supported me during all these years. Although only a few of us are still in Switzerland, the friendship that links us is too strong to lose contact. Thank you Ceci + David = Àlex Jr., Raquel + Pablo = Juliette, and Àlex, for always being there and for all the good moments that we have passed together...and that we will certainly live together in the future!

But of course, all this has been possible thanks to the unconditional support and guidance of my family. Especially of my parents, Juan and Inés, and my sisters, Cristina and Silvia. Although some kilometres are between us, you are always in my heart and I always feel that I have all your support. Thank you! I am also very grateful to Eva, Pablo and Magalí, for accepting me in their family...and for accepting my theft of Verònica! And then, like a dessert, I have left the best for the end, Verònica. You know better than nobody that getting to here has not been easy at all, but you have always been there. Thank you for appearing in my life, thank you for being my best friend, thank you for becoming my family, thank you for following my way.

Model-based Behavioural Tracking and Scale Invariant Features in Omnidirectional Matching

Abstract: Two classical but crucial and unsolved problems in Computer Vision are treated in this thesis: tracking and matching. The first part of the thesis deals with tracking, studying two of its main difficulties: object representation model drift and total occlusions. The second part considers the problem of point matching between omnidirectional images and between omnidirectional and planar images.

Model drift is a major problem of tracking when the object representation model is updated on-line. In this thesis, we have developed a visual tracking algorithm that simultaneously tracks and builds a model of the tracked object. The model is computed using an incremental PCA algorithm that allows to weight samples. Thus, model drift is avoided by weighting samples added to the model according to a measure of confidence on the tracked patch. Furthermore, we have introduced also spatial weights for weighting pixels and increasing tracking accuracy in some regions of the tracked object.

Total occlusions are another major problem in visual tracking. Indeed, a total occlusion hides completely the tracked object, making visual information unavailable for tracking. For handling this kind of situations, common in unconstrained scenarios, the Model cOrruption and Total Occlusion Handling (MOTOH) framework is introduced. In this framework, in addition to the model drift avoidance scheme described above, a total occlusion detection procedure is introduced. When a total occlusion is detected, the tracker switches to behavioural-based tracking, where instead of guiding the tracker with visual information, a behavioural model of motion is employed.

Finally, a Scale Invariant Feature Transform (SIFT) for omnidirectional images is developed. The proposed algorithm generates two types of local descriptors, Local Spherical Descriptors and Local Planar Descriptors. With the first ones, point matching between omnidirectional images can be performed, and with the second ones, the same matching process can be done but between omnidirectional and planar images. Furthermore, a planar to spherical mapping is introduced and an algorithm for its estimation is given. This mapping allows to extract objects from an omnidirectional image given their SIFT descriptors in a planar image.

Keywords: Tracking, Matching, PCA, On-Line Learning, Pedestrian Tracking, SIFT, Omnidirectional Vision.

Model-based Behavioural Tracking and Scale Invariant Features in Omnidirectional Matching

Résumé: Dans cette thèse, nous traitons de deux problèmes cruciaux et non-résolus à ce jour, dans le domaine de la vision assistée par ordinateur: le suivi et la correspondance. La première partie de la thèse traite du suivi, par l'étude de ses deux principales difficultés: la corruption des modèles de représentation d'objets et les occlusions totales. Dans la seconde partie de la thèse, nous nous intéressons au problème de la correspondance de points entre des images omnidirectionnelles, et entre des images omnidirectionnelles et plannaires.

Concernant le suivi, la corruption des modèles est un problème majeur, notamment lorsque la mise à jour des modèles de représentation d'objets se fait en temps réel. Dans cette thèse, nous avons développé un algorithme de suivi visuel, qui en plus de sa fonction première, construit simultanément un modèle d'apparence de l'objet. Le modèle est calculé en utilisant une Analyse en Composante Principale (ACP) incrémentale, qui permet la pondération des échantillons. Ainsi, la corruption du modèle est évitée par l'ajout des échantillons pondérés, basé sur une mesure de confiance de l'objet suivi. Nous avons également proposé une pondération spatiale, dans le but de pouvoir pondérer des pixels et d'augmenter la précision du suivi pour certaines zones de l'objet considéré.

Les occlusions totales représentent un autre problème majeur du suivi visuel. En effet, l'occlusion totale cache entièrement l'objet, et rend indisponible toute information utile au suivi. Pour gérer ces situations, communes dans les scénarios non-constraints, nous proposons un cadre théorique appelé MOTOH pour "Model cOrruption and Total Occlusion Handling". Dans ce cadre théorique, et en plus du schéma d'évitement de corruption des modèles décrit précédemment, nous proposons une procédure de détection des occlusions totales. Lorsqu'une occlusion totale est détectée, l'algorithme de suivi passe à un suivi basé sur le comportement. Plus précisément, l'algorithme utilise un modèle de mouvement comportemental, à la place des informations visuelles.

Enfin, une transformation des informations ne dépendant pas de l'échelle (SIFT), est développée pour les images omnidirectionnelles. L'algorithme génère deux types de descripteurs: les descripteurs sphériques locaux et les descripteurs plannaires locaux. Avec les premiers, nous pouvons établir la correspondance entre des points dans des images omnidirectionnelles; avec les seconds, la correspondance entre des points peut être établie entre des images omnidirectionnelles et plannaires. En outre, une transformation pour passer d'images plannaires à sphériques est proposée, et son estimation est donnée. Cette transformation permet d'extraire des objets d'une image omnidirectionnelle, étant donné leurs descripteurs SIFT dans une image plannaire.

Mots Clés: Suivi, Correspondance, ACP, Apprentissage en Direct, Suivi de Piétons, Vision Omnidirectionnelle.

List of Abbreviations

AR	Augmented Reality
CNL	Cross Nested Logit
CV	Computer Vision
DCM	Discrete Choice Model
DCPM	Discrete Choice Pedestrian Model
EKF	Extended Kalman Filter
EM	Expectation-Maximisation
EVD	EigenValue Decomposition
FN	False Negative
FP	False Positive
HCI	Human-Computer Interaction
HL	Hybrid Localisation
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradients
IPCA	Incremental Principal Component Analysis
ITWPCA	Incremental Temporally Weighted Principal Component Analysis
ITWVT	Incremental Temporally Weighted Visual Tracking
ITWVTSP	Incremental Temporally Weighted Visual Tracking with Spatial Penalty
IVT	Incremental Visual Tracking
KF	Kalman Filter
KLT	Kanade-Lucas-Tomasi
LPD	Local Planar Descriptor
LSD	Local Spherical Descriptor
MC	Monte Carlo
MHL	Multiple-Hypothesis Localisation

MOTOH	Model cOrruption and Total Occlusion Handling
MS	Mean Shift
OF	Optical Flow
PCA	Principal Component Analysis
PF	Particle Filter
RMSE	Root Mean Squared Error
SHL	Single-Hypothesis Localisation
SIFT	Scale Invariant Feature Transform
SIR	Sampling Importance Resampling
SIS	Sequential Importance Sampling
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
VT	Visual Tracking

Contents

Chapter 1

Introduction and Motivations

Contents

1.1 Context

Visual perception is the capacity of capturing the information contained in visible light. Human beings have developed an advanced vision, which is the capacity of interpreting the information captured in this visual perception process. Indeed, complex visual processes are performed by humans without any special effort, such as recognising subjects or following a moving object.

The engineering field that studies the development of vision capacities in a computer is known as Computer Vision (CV). Nowadays, CV is experiencing an important development thanks to the increasing complexity of the algorithms developed by the CV community, which allows to perform more and more complex visual tasks. The capacity of achieving the current degree of complexity has been favoured by the continuous increase of computational power, as well as the decrease in the cost of visual perception devices, i.e. cameras. This decrease of costs has allowed a growth in the number of installed surveillance cameras and a generalisation in the use of cameras in almost any new device, from video-game consoles to mobile phones.

Matching of features, points, regions or objects is a core vision task. It can be studied in a dynamical context, where the matching is performed across time in a video sequence, or in a static context, where the matching is performed between different images without a relevant temporal relationship. The first is the kernel of Visual Tracking (VT), the second is performed in applications such as object detection or image registration.

On one side, Visual Tracking is an important vision task required by a wide variety of CV applications, such as video-surveillance, human-computer interaction, traffic monitoring or medical imaging. Tracking consists of following a target over time. VT denotes the group of techniques that employ visual information for performing this tracking. The difficulty of VT lies in the dynamic nature of the problem, which generates changes in the tracked object and in the scene context. On the one hand, noise, illumination changes or camera motion generate changes in the scene that is being observed. On the other hand, the tracked entity changes due to occlusions, deformations or pose changes.

The use of robust features for tracking can minimise the effects of most of the aforementioned variabilities. However, some other aspects cannot be considered only by using visual information. For instance, when we see a person walking and passing behind a column, we know that this person will very likely appear on the other side of the column after some period of time. We do not need any visual cue on the other side of the column to infer this, instead we use our previous experience in similar situations and our experience about human behaviour. This kind of information is not strictly visual, but provides useful information to vision.

Mathematical models allow to exploit this complementary dimension of information in a CV application. Techniques such as Hidden Markov Models (HMM), Gaussian networks, Discrete Choice Models (DCM) or Support Vector Machines (SVM) can be used for simulating human experience about the observed scene. Their combination with CV techniques based only on visual information can be a way of developing more robust and powerful CV algorithms, with a closer performance to the human vision system.

On the other side, applications such as object detection or image registration are crucial for the development of new interactive applications between humans and digital devices. This has encouraged a lot of research, producing a wide spectrum of algorithms robust to typical visual changes such as affine transformations, illumination changes or noise. However, the hardware sometimes imposes constraints to the final application that are hard to deal with, such as narrow fields of view. This has encouraged research on new architectures of visual sensors, such as depth cameras, plenoptic cameras or omnidirectional cameras. As a first approach, CV algorithms developed for standard projective cameras can be applied to these new types of cameras. Nevertheless, specific algorithms, taking into account their specificities, have to be developed to completely exploit all their advantages.

1.2 Motivations

A common step performed in a wide variety of CV applications is the matching of points, features, regions or objects, i.e. visual entities. For instance, in object detection applications, the objective is to match a given object into a given image, or in VT applications a tracked object is matched across frames. The critical point of this matching procedure is the variability of the visual entities that are being matched.

The visual information received from a visual entity can be considered as a realisation of a random process. Then, observing this visual entity, from different points of view, during a certain period of time, or from different types of visual sensors, produces different realisations of this random process. CV algorithms need to have the capacity of adapting themselves to these variations, making the procedure robust. This is hard due to the impossibility of modelling some of the aspects that introduce the variability. We divide the sources of visual information variability introduced in the realisation of the random process in two groups: *endogenous sources* and *exogenous sources.*

On the one hand, *endogenous sources* are those that are directly related to the visual entity itself, such as its motion or its deformation in the case of a deformable object. *Endogenous sources* of variability are, in general, not hard to consider if we have a good knowledge about the visual entity. For instance, in the case of a planar and rigid moving object observed with a regular projective camera, the most remarkable *endogenous sources* of variability are the affine transformations of the object, and it is not difficult to account for them in a CV application.

On the other hand, *exogenous sources* are those not directly related to the visual entity, but to other aspects of the scene. *Exogenous sources* of variability are not straightforward to consider in a CV application, since even a complete knowledge of the observed object (colour, texture, 3D information, etc) can be useless for handling them. Examples are partial occlusions, total occlusions, noise or illumination changes.

Some aspects of the variability introduced by *exogenous sources* can be reduced by improving the visual sensor. In this sense, omnidirectional visual sensors enlarge considerably the field of view compared to regular cameras, which can improve for instance some occlusion aspects. However, in some cases this variability cannot be handled by using only visual information, as for example in the case of a total occlusion. This motivates the study of interdisciplinary approaches combining CV and mathematical modelling techniques, for producing human-like systems where the lack of visual information is thwarted by experience on similar behaviour.

With the objective of dealing with challenging situations, which are mainly generated by this uncontrolled *exogenous sources* of variability, we have investigated three motivating problems in the dynamical and static contexts introduced above:

- **Handling non-persistent exogenous sources of variability in VT:**

Short-time occlusions, temporary changes in lighting or partial occlusions of an object that is being tracked are difficulties that a VT algorithm has to deal with. Usually, tracking algorithms use a model of the object being tracked for determining its position at each frame of the video. In general, it is useful to update the model with new information about the object, in order to keep it adapted to the tracking conditions at any moment. This update has the drawback that bad quality samples of the object can corrupt the model, causing a loss of track. In this thesis, we have investigated how the model can be updated without corrupting it with bad quality samples of the object, giving robustness to the updating process. Furthermore, we have developed a strategy for achieving more accuracy in important regions of the tracked object.

- **Total occlusions in VT:** A total occlusion of an object is an important problem in tracking, especially when this occlusion is somehow persistent in time. A total occlusion causes the complete loss of visual information from the tracked object during a period of time. Therefore, other sources of information have to be exploited for continuing the tracking. We have developed a framework where a probabilistic behavioural model replaces the visual information whenever a total occlusion occurs. The proposed framework has been applied to pedestrian tracking, where we have studied the use of a pedestrian walking behaviour model, calibrated and validated on real data.

- **Point matching between omnidirectional images and planar and omnidirectional images:** Standard visual perception devices can be approximated by the pinhole camera model. In this model, the 3D world is projected into a 2D plane. For this reason standard cameras are also called planar cameras. As commented before, new architectures of visual sensors can solve or minimise some problems of planar images. The problem is that, omnidirectional visual perception devices capture visual information as if the sensor were placed in a curved surface, whose geometry is embedded in the captured image. This can distort considerably the visual information of the observed objects, which entails two main consequences. First, the same object at different points of an omnidirectional image can present very different appearances. Secondly, the image of the same object captured by a planar sensor and by an omnidirectional sensor can also differ considerably. This justifies the development of algorithms that take into account this geometry and, when working with planar and omnidirectional images, work in a common framework for combining both types of images. With this in mind, we have studied the problem of point matching between omnidirectional images and between omnidirectional and planar images, developing an algorithm that performs both tasks successfully, using a spherical scale-space representation of omnidirectional images.

1.3 Contributions

Several contributions are presented in this dissertation. A summary of them is exposed below:

- **Weighted incremental Principal Component Analysis (PCA) algorithm:** We have developed an incremental PCA algorithm where the contribution of each data sample (columns of the data matrix) can be weighted. The technique used for weighting the data samples cannot be adapted for weighting variables (rows of the data matrix). However, in the context of VT we have developed a way of applying spatial weights for increasing tracking accuracy in predefined regions of the tracked object.

- **Model corruption avoidance in VT:** The unsupervised update of a model with new data is a very sensitive procedure. In VT, bad data samples have to be detected in order to adapt their contribution to the model accordingly. In this sense, we have combined our weighted incremental PCA algorithm with a measure of the quality of a tracking, in order to weight the contributions of the tracked samples to the model of the tracked object.

- **General framework for dealing with total occlusions in VT:** Total occlusions are one of the most challenging problems in VT. In this thesis, we have developed a tracking framework where the lack of visual information caused by a total occlusion is counteracted by a probabilistic behavioural model of motion. The framework is very flexible, allowing its configuration from a general purpose tracking to an object specific tracking, where the motion pattern is known.

- **Complete VT framework:** Combining the model corruption avoidance strategy and the management of total occlusions, we have developed a complete and robust VT framework. The tracking approach builds and updates a model of the tracked object while tracking, avoids model corruption by weighting the contribution of samples, increases, if needed, tracking accuracy in some regions of the tracked object and handles total occlusions.

- **Novel interdisciplinary approach to VT:** The complete VT framework proposed in this thesis has been applied to pedestrian tracking using a pedestrian walking behaviour model, estimated and validated on real data. This approach combines two fields, CV and mathematical modelling of behaviour. The obtained results show the advantages of using behavioural data in addition to visual data in CV applications. The pedestrian walking behaviour model was published in [Robin 2009].

- **Scale Invariant Feature Transform (SIFT) on the sphere for omnidirectional images:** A scale invariant feature transform for omnidirectional images on the sphere has been developed. Two types of feature descriptors have been introduced. The first one, called Local Spherical Descriptor (LSD), is defined on the sphere, as the omnidirectional image. The second one, Local Planar Descriptor (LPD), is defined in a tangent plane to the sphere. This work has been submitted to the International Journal of Computer Vision and is nowadays in the second revision round.

- **Matching between points in omnidirectional images and points in planar and omnidirectional images:** By performing the SIFT on the sphere aforementioned, LSDs and LPDs can be computed. On the one hand, LSDs can be used for matching points between omnidirectional images, which can be useful for instance in motion estimation applications. On the other hand, LPDs allow to match points between planar images and omnidirectional images. This is useful for instance in CV applications requiring an object detection procedure in omnidirectional images, since the great majority of databases of objects consists of planar images. In addition, a planar to spherical mapping is also introduced for transferring segmentations performed in planar images to spherical images. This work has been submitted to the International Journal of Computer Vision and is nowadays in the second revision round.

1.4 Organisation of the thesis

This thesis is organised as follows. Chapter 2 is an introduction to the theoretical aspects of visual tracking. There, the current state-of-the-art in tracking is discussed, making special emphasis in Particle Filter (PF) based methods, given its importance in subsequent chapters. The base VT method used in the thesis is also described in detail.

In Chapter 3, we introduce the techniques that we have developed to avoid model corruption in on-line model updating. These techniques are based on weighting the samples where PCA is performed. A method for simulating a weight applied to individual pixels, in the context of VT, is also introduced in this Chapter. This pixel weighting strategy can increase the accuracy of the performed tracking in some specific regions of the tracked object.

Then, in Chapter 4, we study the problem of total occlusion, developing a method for handling it in VT applications. This allows to define the complete VT framework, whose use in the problem of pedestrian tracking is introduced, showing how to exploit visual and behavioural data for tracking.

Afterwards, in Chapter 5, the scale-invariant feature transform on the sphere is introduced, developing the matching algorithms between points in omnidirectional images and between points in planar and omnidirectional images.

Finally, in Chapter 6 we discuss the conclusions that can be extracted from this thesis, as well as possible lines of future research.

Chapter 2

Visual Tracking in Computer Vision

Contents

2.1 Introduction

Visual Tracking is a core problem in many CV applications. For mentioning some examples, in the context of Human-Computer Interaction (HCI), [Polat 2003] develop a tracking algorithm of hands and faces for collaborative HCI; [Santis 2009] use an eye-tracking algorithm for communicating with disabled people; and [An 2011] use a tracking algorithm to detect finger gestures and control a mobile phone. In traffic monitoring, [Reinartz 2006] use a vehicle tracking algorithm for monitoring traffic using airbone cameras, and [Semertzidis 2010] use a VT algorithm for real-time monitoring of traffic in a network of calibrated cameras. In the context of video-surveillance, [Huang 2008] use a tracking algorithm for night tracking of moving objects in an autonomous outdoor surveillance system, and [Baseggio 2010] develop an autonomous surveillance system where a defined perimeter is autonomously patrolled and mobile targets tracked. Or in the recent and fast growing field of Augmented Reality (AR), where [Marimon 2007b] develop a hybrid marker/feature tracker for recovering camera position in Augmented Reality applications.

The main task of a tracking algorithm is to assign consistent labels to tracked objects along all the frames of a video sequence. Given a video sequence S composed of image frames I_k, i.e.

$$S = \{I_k | k \in K \subseteq \mathbb{N}\}, \tag{2.1}$$

where k is a temporal index, a tracking algorithm estimates for every tracked object j, a time series

$$x^{(j)} = \{x_k^{(j)} | k \in K \subseteq \mathbb{N}\}, \tag{2.2}$$

$j \in J$, where J denotes the set of objects being tracked. Each element $x_k^{(j)}$ of the time series $x^{(j)}$ denotes the state of object j at time k and defines its trajectory over time.

Visual Tracking in unconstrained scenarios is an unsolved CV problem. From a static point of view, the main difficulty of VT is clutter. Clutter is the phenomenon of similarity between features on target regions and features on non-target regions (see Figure 2.1). From a dynamic point of view, the temporal dimension in VT implicitly entails changes in the object of interest due to noise, deformations, occlusions, illumination changes, etc. Handling these changes and dealing with clutter are the core difficulties of VT, and the ability of an algorithm for dealing with one or more of these problems, makes the difference between VT algorithms. Since there is no universal tracking algorithm, the knowledge about the faced problem helps in choosing, adapting or developing VT algorithms that are robust to the specific difficulties of an application. In this Chapter we give an introduction to the structure of VT algorithms, as well as an overview of the state-of-the-art on this topic.

A common preliminary operation to tracking is object detection, although some algorithms perform a simultaneous detection and tracking [Czyz 2007, Breitenstein 2009]. Indeed, when a tracking algorithm is intended to follow some precise objects, these objects need to be previously detected. In this thesis we do not deal with this problem. In Chapter 3 and Chapter 4 we always consider that the starting bounding boxes are given. We only tackle object detection indirectly in Chapter 5, where object detection is one of the potential applications of the developed algorithm. The interested reader in object detection algorithms is referred, as a starting point, to [Yang 2002], [Mundy 2006], [Enzweiler 2009], [Galleguillos 2010], [Gerónimo 2010] and references inside them.

This Chapter is organised as follows. In Section 2.2 we give a general overview of the state-of-the-art in VT, describing the components of a general VT algorithm and the main techniques applied on each of these components. In Section 2.3, we introduce Particle Filters, given their importance in tracking and their use in posterior chapters of the thesis. In Section 2.4 we introduce in detail the Incremental Visual Tracking algorithm, which is the starting point of the tracking algorithms developed in this thesis. Then, in Section 2.5, performance evaluation

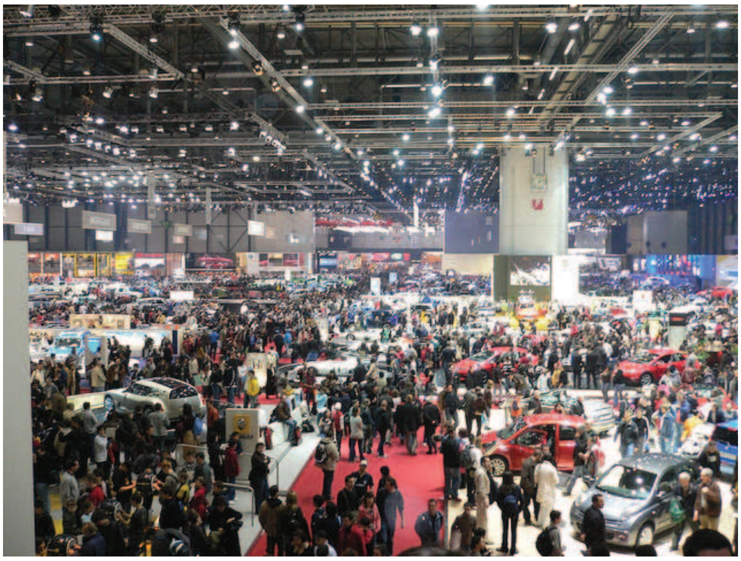


Figure 2.1: Example of clutter in an image.

of VT algorithms is introduced. Finally, in Section 2.6 we conclude this chapter by giving some insights of the developments of this thesis with respect to the contents exposed in this chapter.

2.2 Visual Tracking Algorithms

From a bottom-up point of view, a VT algorithm can be roughly defined by describing three main blocks [Yilmaz 2006, Maggio 2010]: the feature extraction block, the object representation block and the object localisation block.

A generic VT algorithm can be seen then as the application of these three blocks according to the schematic representation in Figure 2.2. Given a frame of a video sequence, the first block performs a feature extraction on its captured visual information. These extracted features, together with the model of the object computed in the object representation block, feed the object localisation block for estimating the new state of the object of interest. Optionally, the new computed state can feed the object representation block for updating the model. In the following sections, each one of these three blocks is described more in detail, giving a snapshot of their respective state-of-the-art.

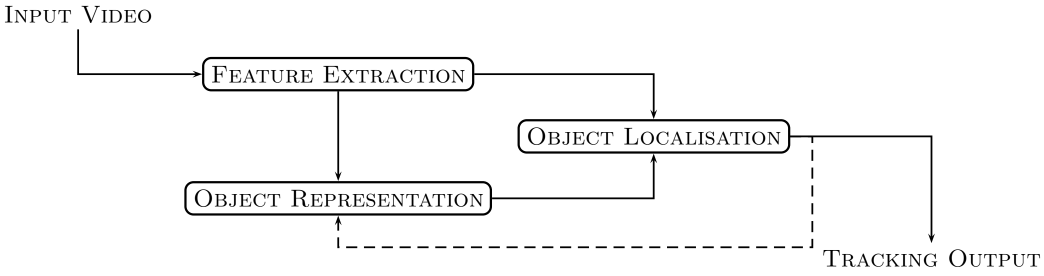


Figure 2.2: Schematic view of a general VT algorithm.

2.2.1 Feature Extraction

A visual sensor receives visual information from the scene that it is observing. Robust features need to be extracted from this rough visual information in order to work with this data. In VT applications, robustness in features basically means persistence along time and uniqueness of the features placed on the object of interest, i.e. the object being tracked. The uniqueness characteristic has two objectives: distinguish the object of interest from other similar objects in the scene and deal with clutter.

The robustness of the performed tracking is directly related with the robustness of the used features. In this sense, the features that are extracted from the visual information of the object of interest are a critical point of any VT algorithm. In general, these features can be divided into three main families [Maggio 2010]:

- **Low-level features:** Low-level features consider the local information provided by individual pixels. These are, in general, the easiest features in terms of computation complexity, but also the most affected by the exogenous sources of variability described in Chapter 1.
- **Mid-level features:** Mid-level features are those computed using information provided by groups of pixels.
- **High-level features:** High-level features consider whole objects as features. These are the most complex in terms of computation, but are supposed to be the most robust.

Among low-level features, colour is one of the most popular due to its simplicity and its suitability to be used with deformable objects. There are a wide variety of colour spaces, each one with its advantages and disadvantages. In [Wang 2008a], the authors use RGB, HSV and normalised RGB as features for tracking. RGB is not discriminative enough, therefore they considered also HSV and normalised RGB, that are more discriminative and robust to illumination changes, although more sensitive to noise. Another set of common low-level features are image derivatives. Derivatives encode information about local intensity changes, which are meaningful features especially at the boundaries of the object of interest. In [Birchfield 1998], boundary of faces is tracked using gradient information of the image combined with colour histograms. Finally, motion is another low-level feature useful in tracking. It presents the difficulty that only apparent motion can be estimated from a video sequence. Apparent motion is the observed motion due to changes on the intensity information of the image. It is related to real motion, but it also depends on noise and illumination, which makes the problem of estimating real motion from apparent motion hard. Motion information is usually treated using the Optical Flow (OF), which is a vector field that defines the translations of pixels in a region [Horn 1981, Lucas 1981, Reddy 1996, Zach 2007]. In [Shin 2005] a tracking algorithm using OF features is presented. The method can deal with partial occlusions thanks to the motion information given by OF. This motion is used to predict positions of occluded features.

Edges are a typical example of mid-level features. Several edge detector approaches exist in the literature. The interested reader is referred to [Bowyer 2001]. A popular approach is the Canny edge detector [Canny 1986], that is used in [Wang 2010] as feature extractor for tracking in a contrast media injection control application in computed tomography angiography. Another type of mid-level features are interest points and regions. Interest points are points on an image that contain good image features, in the sense of accurate localisation and repeatability. In [Wang 2008b], Harris corner features [Harris 1988] are used as facial features for face tracking. If interest points are computed using a scale-space representation of the image, then they are called interest regions. One of the most well-known interest region detector is the one defined in the Scale Invariant Feature Transform (SIFT) [Lowe 2004]. This detector is based on the localisation of some particular maxima and minima of difference of Gaussian filters applied to the image (see Chapter 5 for further details). In [Zhou 2009], SIFT features and colour are combined for tracking. Finally, uniform regions, which are regions of an image sharing some predefined property, are also mid-level features. In [Yamane 1998], the authors combine OF with uniform brightness regions for tracking pedestrians.

High-level features are usually the output of an object detector trained to detect foreground or background regions. Foreground detectors detect directly the object of interest, while background detectors detect everything that is not of interest. In [Verma 2003], [Hidaka 2006] and [Meynet 2008], face detectors are used for tracking faces, and in [Leibe 2008] several object detectors are combined for tracking pedestrians and vehicles. In [Kalal 2010], an object detector adapted to the object of interest is learnt on-line and used for tracking. In [Stauffer 2000] and [Kim 2008], it is the output of a background detector what is used for tracking moving objects. Note that background detectors are usually restricted to fixed camera environments.

A VT algorithm does not necessarily use only one type of features. Indeed, VT approaches that combine several types of features are common and some examples have already been cited. This strategy is intended to make the tracking more robust. For instance, in [Maggio 2007] and [Wang 2008a], gradient and colour features are considered, and in [Wu 2008] colour features and SIFT descriptors are combined for tracking.

2.2.2 Object Representation

Feature extraction defines the space where the object of interest will be defined, i.e. the space where the characteristics of the tracked object will be defined. For this, a mathematical model of the object of interest, the object representation, is employed. This model contains shape and appearance information about the object of interest in the feature space. Shape information encodes the shape of the tracked object, while appearance information encodes the visual information, in the defined feature space, inside the shape. In Figure 2.3 a schematic classification of the most common methods for encoding shape and appearance information is shown.

Shape representations can be divided into three main families: basic geometric representations, articulated shape representations and deformable shape representations. Basic shape representations are those that consider the shape of the object of interest as a basic geometric shape. These basic geometric shapes can go from a single point [Veenman 2001] to the volumetric schematic representation of the shape of an object [Roller 1993]. Point shape approximations are suitable for tracking objects that cover a small region of the image, either because they are small or because they are far from the sensor. Also, tracking algorithms using object detectors use commonly point shape representations, since the object detector already accounts for size and appearance of the object. Area approximations are also common in the literature, usually considering rectangles or ellipses. In [Ross 2008], a rectangle, and all its affine transformations, represents the shape of the object of interest. In [Comaniciu 2003] and [Jepson 2003], it is an ellipsoidal region the chosen area that delimits the shape of the object.

Articulated shape representations combine several basic shape representations that are joined under motion constraints at connection points. This type of shape representations is usually used for human tracking [Lan 2004, Sundaresan 2009].

Finally, deformable shape representations relax the shape rigidity imposed by basic and articulated shape representations. Active contours are a good example of deformable shape representations. They are usually characterised by a set of control points where some constraints are imposed [Roh 2007], or by using a level set approach [Yilmaz 2004]. Another example of deformable shapes are point distribution models, where a set of points, placed on the boundary as well as

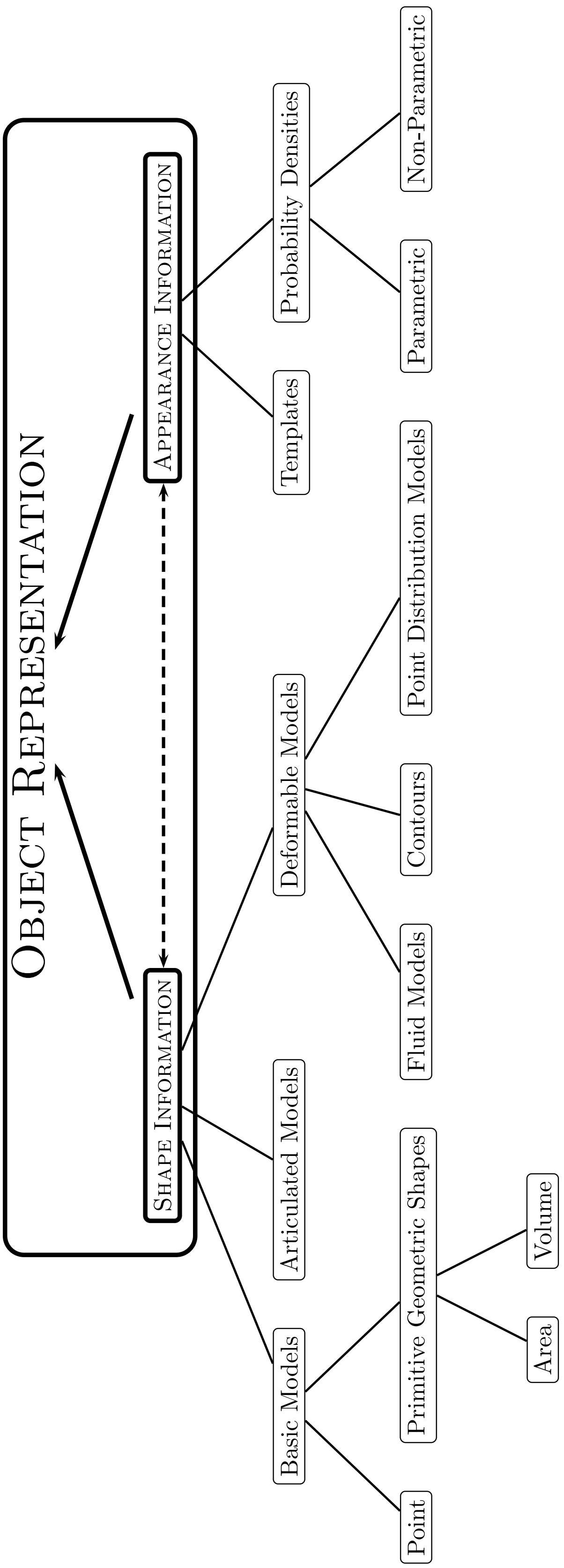


Figure 2.3: Object representation methods

inside the boundary of the object of interest, are tracked. The most remarkable representative of this group are Active Shape Models [Cootes 1995].

Appearance representations can be divided basically in template representations and probability density representations. A template is an appearance representation method that maps pixel values into a predefined coordinate system according to their position within the region of interest. This coordinate system is known as the template coordinate system. For instance, in [Ross 2008], a rectangular template is considered and the mapping between the pixels inside the tracked region and the template is simply an affine transformation.

Probability density representations are the other big family of appearance representations. Probability densities can be parametric or non-parametric. Among parametric methods, Gaussian density estimation is a common and convenient method, since only two parameters (mean and variance) define precisely the density function. In [Yang 1996], a Gaussian is used for modelling face colour in a face tracking application. For handling multimodal distributions, mixtures of Gaussians are used in the literature [McKenna 1999, Papadourakis 2010].

On the group of non-parametric probability density representations, we can find techniques such as Parzen windows [Chen 2010], although the most important are histograms. Histograms are a powerful appearance representation, invariant to transformations of the interest region shape. In [Lu 2001, Moreno 2002, Leichter 2010], histograms of colour features in the tracked region are used for appearance representation. In [Marimon 2007a], the features used for computing the histogram are gradients. One of the drawbacks of histograms is the lack of spatial information, which reduces their discriminative power. To account for that, the region delimited by the shape is sometimes divided in subregions where histograms are computed. This way, the object representation encodes spatial information as well as histograms of features. A well known appearance information representation computed using this technique is the Histogram of Oriented Gradients (HOG) [Dalal 2005]. HOG divides the region delimited by the shape in subregions where individual histograms of gradient angle values are computed. In [Lu 2006, Bilinski 2009], the authors encode appearance information using HOG for tracking pedestrians. Let us note that classifiers trained on HOG descriptors using Support Vector Machines (SVM) are a powerful pedestrian detection technique [Dalal 2005].

Shape representations are related to appearance information in the sense that they delimit the information that is considered as belonging to the tracked object. In some cases this relation is strong and fixes the appearance, as for instance when point representations are considered for shape. In this case, the considered appearance information is necessarily a template composed of just a point in the feature space. Note however that a point in the feature space can represent a region in the image, like for instance when using object detectors.

With respect to when the object representation model is computed, three strategies can be adopted in tracking:

- **A priori computation** with respect to the beginning of the tracking. In this case, the model is computed a priori.
- **Predefined computation** with respect to the beginning of the tracking, i.e. one or several of the starting snapshots of the object of interest are used for computing the model.
- **During** the tracking. In this case, the snapshots of the object of interest obtained by the tracking algorithm are constantly used for computing and updating the model.

A model computed a priori lightens the computational cost of the tracking algorithm, but it is not well adapted to the object of interest in the real conditions of the tracking. A model computed using some predefined samples of the tracking increases slightly the computational cost of the algorithm but obtains a better adapted model. However, changes in the conditions of the object after the initialisation period are not taken into account in the model. The last strategy achieves a complete adaptation to the object of interest by continuously updating the model, with the disadvantage of increasing computational cost and with the risk of model corruption (drift).

2.2.3 Object Localisation

The final block of a VT algorithm is the one in charge of object localisation, i.e. the block in charge of the tracking. This block, at a given frame, gets the information of the feature extraction and the object representation model, and gives as output an estimation of the state of the object of interest. This estimation is in general based on the hypothesis of a smooth change of position, shape and appearance.

Object localisation methods can be classified in two big families: Single-Hypothesis Localisation (SHL) methods and Multiple-Hypothesis Localisation (MHL) methods.

In SHL methods, only one track is evaluated at each time step. The target state is computed analytically as the solution of an optimisation problem where a cost function is minimised. These methods can arise from considering a deterministic problem or a stochastic problem. In [Veenman 2001], the authors developed a point tracker that considers three underlying motion models: an individual motion model, a combined motion model and a global motion model. The tracking problem is solved as a minimisation problem of the deviation with respect to the combination of these three motion models. In [Hager 1998], a general parametric model for motions and deformations of the target region is introduced, and the tracking is solved as a least squares error minimisation problem. In [Comaniciu 2003] the well known Mean Shift (MS) tracker is introduced. MS is an iterative algorithm [Cheng 1995] for locating local maxima of density functions. The Kanade-Lucas-Tomasi (KLT) tracker [Lucas 1981, Tomasi 1991] is another good example of SHL method. KLT searches local displacements as the solution of a first order Taylor approximation of the image. Another example of SHL method is Kalman Filter (KF) [Li 2010] and Extended Kalman Filter (EKF) [Ndiour 2010, Haj 2010] based approaches. However, the performance of these methods is limited due to the strong hypothesis on which they are based (see [Arulampalam 2002] for details).

On the contrary, MHL methods evaluate simultaneously multiple candidate tracks (hypothesis) per object of interest per time step. Each one of these hypothesis is validated against visual information and motion models. It can be seen [Doucet 2009] that the number of hypothesis needed to correctly sample the state space grows exponentially with its dimensionality. This is the main drawback that has to be paid in order to have the higher flexibility that MHL methods provide.

Several MHL approaches exist, like for instance approximate grid-based methods, but the most important and used methods are those based on Particle Filters (PF). A high number of VT algorithms in the literature use PF approaches for estimating the state of the object [Isard 1998, Zhou 2004, Ross 2008]. For this reason and given the importance of PF in the algorithms developed in Chapter 3 and Chapter 4, PF are described in detail in Section 2.3.

There exist also methods that combine aspects of SHL methods with aspects of MHL methods. They are known as Hybrid Localisation (HL) methods and try to reduce computational cost of MHL by improving the sampling with SHL techniques. A common approach is to combine MS with PF. Indeed, the MS optimisation step moves particles towards local peaks in the likelihood function, which improves the sampling efficiency [Shan 2007, Wang 2009, Maggio 2009].

2.3 Particle Filters

A theoretical Bayesian solution of a tracking problem would construct the posterior probability density function of the state based on all available information and measurements. In a real tracking problem, an estimate has to be computed for every frame, i.e. for every new measurement. This is the scenario of application of a recursive filtering approach like Particle Filters (PF).

Let us consider the state sequence $x = \{x_k | k \in \mathbb{N}\}$ (Equation (2.2) discarding the superindex). The evolution of states can be denoted by

$$x_k = f_k(x_{k-1}, v_{k-1}), \tag{2.3}$$

where $f_k : \mathbb{R}^{N_x} \times \mathbb{R}^{N_v} \to \mathbb{R}^{N_x}$ is a possibly non-linear function, $\{v_k | k \in \mathbb{N}\}$ is an i.i.d. process noise sequence, and N_x and N_v are the dimensions of the state and the process noise vector, respectively. For each time step k, the obtained measurement can be denoted by

$$z_k = h_k(x_k, n_k), \tag{2.4}$$

where $h_k : \mathbb{R}^{N_x} \times \mathbb{R}^{N_n} \to \mathbb{R}^{N_z}$ is a possibly non-linear function, $\{n_k | k \in \mathbb{N}\}$ is an i.i.d. measurement noise sequence, and N_z and N_n are the dimensions of the measurement and the measurement noise vector, respectively.

The problem of tracking tries to find filtered estimates of x_k based on the set of available measurements up to time k, i.e. $z_{1:k} = \{z_i | i = 1, \ldots, k\}$. These estimates are computed using the belief on a given state x_k, i.e. $p(x_k | z_{1:k})$, with a given initial prior $p(x_0 | z_0) \equiv p(x_0)$. Then, $p(x_k | z_{1:k})$ can be computed by applying recursively a prediction and an update step.

Let us suppose that $p(x_{k-1} | z_{1:k-1})$ is known, then the prediction step estimates the prior of the state x_k as

$$p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | z_{1:k-1}) dx_{k-1}. \tag{2.5}$$

This expression is obtained by applying the Chapman-Kolmogorov equation, that states that in a Markov process, the conditional probability $p(x_n | x_k)$ can be expressed as

$$p(x_n | x_k) = \int p(x_n | x_m) p(x_m | x_k) dx_m, \; \forall n > m > k \in \mathbb{N}. \tag{2.6}$$

Then, when measurement z_k becomes available, the update step updates the posterior as

$$p(x_k | z_{1:k}) = \frac{p(z_k | x_k) p(x_k | z_{1:k-1})}{p(z_k | z_{1:k-1})}, \tag{2.7}$$

where

$$p(z_k | z_{1:k-1}) = \int p(z_k | x_k) p(x_k | z_{1:k-1}) dx_k. \tag{2.8}$$

Equations (2.5) and (2.7) cannot be computed analytically, except for some particular cases. For instance, if the posterior density at each time step is Gaussian and Equations (2.3) and (2.4) are known and linear, then Equations (2.5) and (2.7) can be computed analytically and the Kalman Filter is obtained. Grid-based methods are another example of exact solutions, and are obtained when the state space is discrete and finite.

Methods that compute the exact solution are known as optimal algorithms, but in general, except for the cases cited above, sub-optimal algorithms have to be used. PF are among these sub-optimal algorithms, and their main representative is the Sequential Importance Sampling (SIS) algorithm. SIS algorithm implements a recursive Bayesian filter by simulation. The idea behind the method is to represent the posterior density function by a finite set of sample points, the particles, with an associated weight. Thus, the posterior can be approximated as

$$p(x_k|z_{1:k}) \simeq \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i). \tag{2.9}$$

where w_k^i is the weight of particle i at time step k, N_s is the number of particles, $\sum_{i=1}^{N_s} w_k^i = 1$ and x_k and x_k^i are the state at time k and the state of particle i at time k, respectively. The $\delta(y)$ function in Equation (2.9) equals 1 for $y = 0$ and 0 otherwise.

Suppose now that $p(x)$ is a hard to evaluate probability density function although, up to proportionality, it can be evaluated through a function $\pi(x)$, i.e. $p(x) \propto \pi(x)$. Then, given an easy to compute Importance Density $q(x)$ and a set of samples $x^i \sim q(x),\ i = 1, \ldots, N_s$, the importance sampling principle states that an approximation of the density $p(x)$ can be computed as

$$p(x) \simeq \sum_{i=1}^{N_s} w^i \delta(x - x^i), \tag{2.10}$$

where $w^i \propto \frac{\pi(x^i)}{q(x^i)}$. Thus, in the conditional case of Equation (2.9), the weights are defined as

$$w_k^i \propto \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})}. \tag{2.11}$$

If the importance density is chosen such that

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}), \tag{2.12}$$

it can be seen that

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})}. \tag{2.13}$$

Furthermore, if the importance density only depends on x_{k-1} and z_k, which is a common situation, then this last equation reduces to

$$w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}. \tag{2.14}$$

Taking all these elements into account, the SIS algorithm can be derived as described in Algorithm 1. A well-known tracking algorithm in CV using PF is the Condensation algorithm [Isard 1998]. It considers $w_k^i \propto p(z_k|x_k^i)$, i.e. the weights

Algorithm 1 Sequential Importance Sampling (SIS) algorithm (extracted from [Arulampalam 2002]).

```
$[\{x_k^i, w_k^i\}_{i=1}^{N_s}] = \mathrm{SIS}[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, z_k]$
for $i = 1 : N_s$ do
    Draw $x_k^i \sim q(x_k | x_{k-1}^i, z_k)$
    Assign to particle $i$ a weight $w_k^i \propto w_{k-1}^i \frac{p(z_k|x_k^i) p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}$
```

of the particles are directly the likelihood of the state of the particle at this time step, and prior and importance information in Equation (2.14) are discarded.

The main problem of the SIS algorithm is degeneracy, which makes that after some iterations only a few particles have a non-negligible weight. In [Bergman 1999], a measure of degeneracy was introduced by estimating the effective sample size

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2}. \tag{2.15}$$

The real effective sample size is defined as

$$N_{eff} = \frac{N_s}{1 + \mathrm{Var}(w_k^{*i})}, \tag{2.16}$$

although it cannot be computed, since the true weight values, denoted by w_k^{*i}, are unknown.

A common practice for fighting against degeneracy is by resampling with replacement the state space, in order to generate a new set of particles. This resampling can be done when Equation (2.15) becomes smaller than a given threshold or directly at each time step. The last produces the Sampling Importance Resampling (SIR) algorithm. For a detailed review of PF theory and algorithms, we refer the reader to [Arulampalam 2002] and [Doucet 2009].

2.4 The Incremental Visual Tracking Algorithm

Object representations computed a priori, or with some starting snapshots of the object of interest, are not robust against changes along time on the appearance of the tracked object. In [Ross 2008], the authors introduced the Incremental Visual Tracking (IVT) algorithm, a VT algorithm where the object representation, built using PCA on grayscale templates, is constantly updated with the samples of the object of interest obtained by the tracker. For achieving this, they introduced an Incremental Principal Component Analysis (IPCA) algorithm with mean update. Since newer samples of the object of interest are usually more representative of its appearance, the authors used a forgetting factor $f \in [0, 1]$

intended to reduce the contribution of old samples to the computed PCA ($f = 1$ means infinite memory and $f = 0$ forces to consider only new samples). In this Section, we are going to describe the IVT algorithm in detail, given its importance in the following chapters of this dissertation. Let us start with the IPCA algorithm.

Given a set of N data samples $Z = [z_1, \ldots, z_N] \in \mathbb{R}^{M \times N}$, where each sample is represented as a vector $z \in \mathbb{R}^M$, PCA performed on these data gives a projection matrix $U \in \mathbb{R}^{M \times K}$, $K \leq N$, that minimises the squared reconstruction error

$$\xi = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\widehat{z}_{ij} - \sum_{p=1}^{K} u_{ip} \sum_{q=1}^{M} u_{qp} \widehat{z}_{qj} \right)^2 , \tag{2.17}$$

where a_{bc} represents the element at row b and column c of matrix A and $\widehat{Z}$ is the matrix obtained by subtracting the sample mean to each column of Z. The columns of the matrix U are the principal components of the data matrix $\widehat{Z}$, i.e. the eigenvectors of the autocovariance matrix of $\widehat{Z}$. These eigenvectors are usually computed using the Singular Value Decomposition (SVD) of the matrix $\widehat{Z}$. In our context, data samples are snapshots of the object of interest, i.e. images. These data samples arrive sequentially in time, therefore we refer to the mean amongst images, i.e. $\mu = \frac{1}{N} \sum_{j=1}^{N} z_j$, as the temporal mean.

Let us consider that we have performed a PCA on a data matrix $Z^{(1)} = [z_1^{(1)}, \ldots, z_{N^{(1)}}^{(1)}] \in \mathbb{R}^{M \times N^{(1)}}$, where each column is a data sample represented as a vector, $z_j^{(1)} \in \mathbb{R}^{N^{(1)}}$. The temporal mean of $Z^{(1)}$ is

$$\mu^{(1)} = \frac{1}{N^{(1)}} \sum_{j=1}^{N^{(1)}} z_j^{(1)}, \tag{2.18}$$

and so the zero mean data matrix is

$$\widehat{Z}^{(1)} = [z_1^{(1)} - \mu^{(1)}, \ldots, z_{N^{(1)}}^{(1)} - \mu^{(1)}]. \tag{2.19}$$

The PCA gives us a projection matrix $U^{(1)}$, composed of the eigenvectors of $Cov(\widehat{Z}^{(1)})$, and a diagonal matrix $\Sigma^{(1)}$, being the elements on the diagonal the eigenvalues of $Cov(\widehat{Z}^{(1)})$. For instance, imagine that we have $Z^{(1)} = [z_1^{(1)} z_2^{(1)} z_3^{(1)} z_4^{(1)}] \in \mathbb{R}^{\nvDash \times \ntrianglerighteq}$ where $z_1^{(1)} = (1,1)^\top$, $z_2^{(1)} = (-1,-1)^\top$, $z_3^{(1)} = (0.5, 0.5)^\top$ and $z_4^{(1)} = (-0.5, -0.5)^\top$. Then, $\mu^{(1)} = (0,0)^\top$, $U^{(1)} = (\sqrt{2}/2, \sqrt{2}/2)^\top$ and $\Sigma^{(1)} = (1.67)$. In this very simple example, the second eigenvector, $(-\sqrt{2}/2, \sqrt{2}/2)^\top$, is discarded because its corresponding eigenvalue is zero.

Suppose now that a new data matrix $Z^{(2)}$, whose sample mean is $\mu^{(2)}$, is received. Based on the works presented in [Levy 2000], Ross et al. introduced an algorithm that provides the projection matrix $U^{(1,2)}$ without computing the whole PCA, where $U^{(1,2)}$ corresponds to the PCA matrix of

$$Z^{(1,2)} = [Z^{(1)} Z^{(2)}]. \tag{2.20}$$

Defining the scatter matrix of a given matrix as the outer product of the zero mean data matrix, it can be seen that the scatter matrix of $Z^{(1,2)}$ is given by

$$S_{Z^{(1)}Z^{(2)}} = S_{Z^{(1)}} + S_{Z^{(2)}} + \frac{N^{(1)}N^{(2)}}{N^{(1)} + N^{(2)}}(\mu^{(1)} - \mu^{(2)})(\mu^{(1)} - \mu^{(2)})^\top, \tag{2.21}$$

where $S_{Z^{(1)}}$ and $S_{Z^{(2)}}$ are the scatter matrices of $Z^{(1)}$ and $Z^{(2)}$, respectively.

Using Equation (2.21) for considering the change of mean, the IPCA algorithm is described in Algorithm 2. Note that eigenvectors corresponding to small eigenvalues can be suppressed, but we use an abuse of notation and denote the final projection matrix and eigenvalues (after suppression), $U^{(1,2)}$ and $\Sigma^{(1,2)}$, respectively.

Algorithm 2 Incremental Principal Component Analysis (IPCA) with mean update. Algorithm introduced in [Ross 2008].

```
Set $\widehat{Z}^{(2)} = [z_1^{(2)} - \mu^{(2)}, \dots, z_{N^{(2)}}^{(2)} - \mu^{(2)}, \sqrt{\frac{N^{(1)}N^{(2)}}{N^{(1)}+N^{(2)}}}(\mu^{(1)} - \mu^{(2)})]$
Compute ${}^{\perp}\widehat{Z}^{(2)} = orth(\widehat{Z}^{(2)} - U^{(1)}U^{(1)\top}\widehat{Z}^{(2)})$, where "$orth()$" denotes orthogonalisation via QR decomposition.
Compute $R = \begin{bmatrix} f\Sigma^{(1)} & U^{(1)\top}\widehat{Z}^{(2)} \\ 0 & {}^{\perp}\widehat{Z}^{(2)\top}(\widehat{Z}^{(2)} - U^{(1)}U^{(1)\top}\widehat{Z}^{(2)}) \end{bmatrix}$, where $f \in [0,1]$ is a forgetting factor that decreases the contribution of old blocks of data.
Compute $SVD(R) = U'\Sigma'V'^\top$
Then, $U^{(1,2)} = [U^{(1)}\ {}^{\perp}\widehat{Z}^{(2)}]U'$, $\Sigma^{(1,2)} = \Sigma'$ and $\mu^{(1,2)} = \frac{fN^{(1)}}{fN^{(1)}+N^{(2)}}\mu^{(1)} + \frac{N^{(2)}}{fN^{(1)}+N^{(2)}}\mu^{(2)}$.
```

A probabilistic interpretation of PCA [Tipping 1999] allows to combine this object representation with a particle filter approach for object localisation. The state-space where particles are placed is composed of the six parameters of an affine transformation: translation (2 parameters), rotation angle, scale, aspect ratio and skew direction. This affine transformation maps the region of interest onto a rectangular template. The particles represent a sample of the posterior density function of the state given the observations. A dynamical model drives the motion of these particles and an observation model assigns their weights.

The dynamical model defines the dynamics between states, and is modelled as a Brownian motion,

$$p(x_k|x_{k-1}) \sim N(x_k; x_{k-1}, \Theta), \tag{2.22}$$

where x_k denotes a point in the state-space at time k and Θ is a diagonal covariance matrix containing the variances of the affine parameters.

The observation model gives a measure of how likely an image region z_i belongs to the subspace generated by the projection matrix U, i.e. Span(U). A similar approach to Condensation [Isard 1998] is adopted, assigning as weight to the particles directly their likelihood, i.e. $w_k^i \propto p(z_k|x_k^i)$. Given an image patch z_i, a projection matrix U, a mean μ and a diagonal matrix of eigenvalues Σ, then

$$\log p(z_i \in \mathrm{Span}(U)) \propto -(d_t + d_U), \tag{2.23}$$

where d_t is the Euclidean distance of $z_i - \mu$ to the subspace Span(U), and d_U is the Mahalanobis distance within the subspace, i.e. the symmetric bilinear form defined by the inverse of the autocovariance matrix of the data. These two distances can be computed as

$$d_t = \frac{1}{\sigma^2}(z_i - \mu)^\top (I - UU^\top)(z_i - \mu), \tag{2.24}$$

where I denotes the identity matrix, and

$$d_U = (z_i - \mu)^\top U \Sigma^{-1} U^\top (z_i - \mu). \tag{2.25}$$

Note that since the principal components define a basis where the data is uncorrelated, the autocovariance matrix reduces to the diagonal matrix of eigenvalues Σ. The σ^2 term can be seen as the average variance lost in the projection:

$$\sigma^2 = \frac{1}{N - N_b} \sum_{i=N_b+1}^{N} \lambda_i. \tag{2.26}$$

In this last equation, N_b denotes the index of the last considered eigenvector, N the total number of eigenvectors and λ_i the eigenvalue corresponding to the i-th eigenvector.

At each time step, the weights of the particles and the dynamical model are utilised to propagate the particles. Then, the observation model assigns weights to these particles. The point in the state-space where the particle with the highest weight is placed, is chosen as defining the image window that contains the tracked object. In order to increase performance, new data added to the PCA is computed in blocks of a prefixed size (every five frames in [Ross 2008]). The maximum total number of eigenvectors in matrix U is also prefixed. In Algorithm 3, the complete IVT algorithm is described.

2.5 Performance Evaluation of Visual Tracking Algorithms

In tracking, performance evaluation of an algorithm is a difficult task. Two types of evaluations are usually performed: visual evaluation and objective evaluation. On

Algorithm 3 Summary of the Incremental Visual Tracking algorithm (extracted from [Ross 2008]).

```
1: Locate the target object in the first frame, either manually or by using an
   automated detector, and use a single particle to indicate this location.
2: Initialise the eigenbasis $U$ to be empty, and the mean $\mu$ to be the appearance
   of the target in the first frame. The effective number of observations so far is
   $n = 1$.
3: Advance to the next frame. Draw particles from the particle filter, according
   to the dynamical model.
4: For each particle, extract the corresponding window from the current frame,
   and calculate its weight, which is its likelihood under the observation model.
5: Store the image window corresponding to the most likely particle. When the
   desired number of new images have been accumulated, perform an incremental
   update (with a forgetting factor) of the eigenbasis, mean, and effective number
   of observations.
6: Go to step 3.
```

the one hand, visual evaluation is a subjective evaluation of the performance of a VT algorithm by observing its behaviour when faced to some particular difficulties. It does not provide a score of its performance, but it is a useful tool when testing the algorithm faced to a specific difficulty. For instance, when a tracked object suffers an occlusion, a ground truth cannot be used for evaluating the algorithm, but a visual evaluation can determine the capacity of the algorithm for handling this kind of situations.

On the other hand, objective evaluation consists on computing a set of numerical performance scores that objectively describe the quality of the algorithm. A common approach for defining these performance scores is by treating the problem as a classification problem. In such a problem, two performance scores are typically defined, namely Precision and Recall. Their computation is based on the concepts of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) samples. In a classification problem with respect to a hypothetical class X, a TP denotes an object from X correctly classified, a TN denotes an object not belonging to X correctly classified, a FP denotes an object not belonging to X incorrectly classified and a FN denotes an object from X incorrectly classified. Then, Precision and Recall scores can be defined as [Olson 2008]:

$$Precision = \frac{|TP|}{|TP| + |FP|}, \tag{2.27}$$

$$Recall = \frac{|TP|}{|TP| + |FN|}, \tag{2.28}$$

where $|TP|$ denotes the number of TPs and so on.

The definition of TP and FP is usually performed using the intersection over union

$$\rho = \frac{|A_T \cap A_{GT}|}{|A_T \cup A_{GT}|}, \tag{2.29}$$

where A_T denotes the region tracked, A_{GT} the region defined by the ground truth and $|\cdot|$ the number of pixels inside a region. Then, if ρ is bigger than a given threshold ρ_{Th}, the sample is considered as a TP sample. If $\rho < \rho_{Th}$ then the sample is classified as a FP.

With respect to TNs and FNs, if there is no object of interest and the tracker does not output any tracked region, the sample corresponds to a TN. FNs are those samples with the object of interest present in the scene but the tracker not tracking it, i.e. without giving any output. Note that TNs and FNs, and therefore the recall score, are related to object detection capabilities of the tracking algorithm.

Another important measure is the lost track ratio

$$\lambda = \frac{N_s}{N_t}, \tag{2.30}$$

where N_s denotes the number of frames of unsuccessful tracking and N_t the total number of frames in the sequence. For using this measure, a definition of unsuccessful tracking is needed. For instance, a threshold T_λ compared to the dice error D_k at a given frame k can be used as an indicator of unsuccessful tracking [Maggio 2010]:

$$D_k = \frac{|FP_k| + |FN_k|}{2|TP_k| + |FP_k| + |FN_k|} > T_\lambda, \tag{2.31}$$

where the subindex k denotes the frame index and $|\cdot|$ denotes the number of pixels. Typical values for T_λ are in the interval $[0.8, 1]$.

The values of λ of several VT algorithms applied on a given sequence, are already an indicator of performance and can be used for comparing them. In case of similar obtained values, precision and recall are then used for comparison.

The definition of an evaluation protocol for comparing correctly and fairly several VT algorithms is not straightforward. An evaluation protocol is composed of a set of performance scores, like for instance those defined above, and a dataset where run the evaluation tests. Datasets for evaluation have the purpose of defining a common scenario of test in order to allow comparison between results of different tracking algorithms. This is probably the most difficult part on the design of an evaluation protocol. The design of a general purpose dataset, with enough variability for correctly evaluating a tracker, even in an application specific scenario, is far from being trivial. In addition, the difficulty of defining a ground truth, or even the impossibility for instance when an occlusion occurs, makes the objective evaluation of VT algorithm with standard evaluation protocols even more complicated. ETISEO [Nghiem 2007], USF-DATE [Kasturi 2009] or PFT [Nawaz 2011] are some of the evaluation protocols present in the literature.

2.6 Conclusions

In this Chapter, the main elements of a VT algorithm have been described. For each one of these elements, representative works present in the literature have been referenced. Special emphasis has been done on the description of techniques that are relevant for next chapters. Indeed, Particle Filtering and the Incremental Visual Tracking algorithm have been described in detail.

Keeping the object representation adapted to the object of interest is a difficult task in VT. This problem is treated in Chapter 3. Another difficulty of VT is the disappearance of the tracked object due to a total occlusion. This problem is tackled in Chapter 4. There, a complete VT framework is introduced. This framework considers the techniques for model update introduced in Chapter 3, as well as the total occlusion handling introduced in Chapter 4. In tracking, a matching process is performed over time, under the assumptions of a smooth change of position, shape and appearance, which are usually true given a frame rate high enough. In Chapter 5, static matching between omnidirectional images, without any smoothness assumption, is studied.

Common approaches for performance evaluation of VT algorithms have been also described. Difficulties of this performance evaluation have been discussed and common measures of performance introduced. For evaluating our proposed algorithms in common situations we will use these measures. In hard specific situations that our algorithms can handle, such as total occlusions, we will perform visual evaluations due to the impossibility of defining a ground truth.

Chapter 3

Sample and Pixel Weighting Strategies for Avoiding Model Drift and Increasing Accuracy in Visual Tracking

Contents

3.1 Introduction

There exists a wide variety of techniques for computing models encoding shape and/or appearance information of a tracked object. For instance PCA [Cootes 2001, Lee 2005, Ross 2008], mixtures of Gaussians [Stauffer 1999, Papadourakis 2010], histograms [Birchfield 2005, Peng 2005], bayesian networks [Park 2004] or boosting techniques [Grabner 2006, Iwahori 2008].

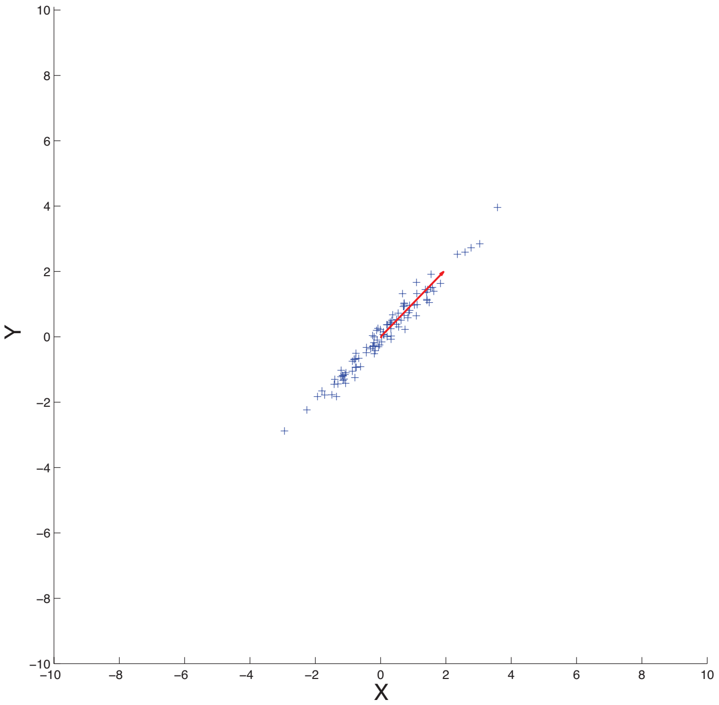
The difficulty with these models is that the appearance of the tracked object is continuously changing, and the model needs to be either built for dealing with these changes or to have the capacity of being adapted to them. In the first option, the changes have to be predicted and taken into account in the model estimation process, which is performed a priori or during an initialisation period. For this reason, the second option is the most effective in terms of adaptability, since the type of changes that the model has to handle does not need to be known beforehand. This strategy is also interesting in terms of dealing with exogenous sources of variability (see Chapter 1) since most of the generated changes are not predictable in advance. However, the adaptation procedure is very sensitive due to the possibility of corrupting the model with bad samples of the object of interest, causing a model drift and the consequent loss of track.

In the tracker introduced in Section 2.4, the model is built on-line and incrementally using a PCA. This allows to maintain the model permanently adapted to the object of interest and their current conditions. The problem with this procedure is that bad tracked samples or temporal modifications of the tracked object, such as occlusions, are also added to the model.

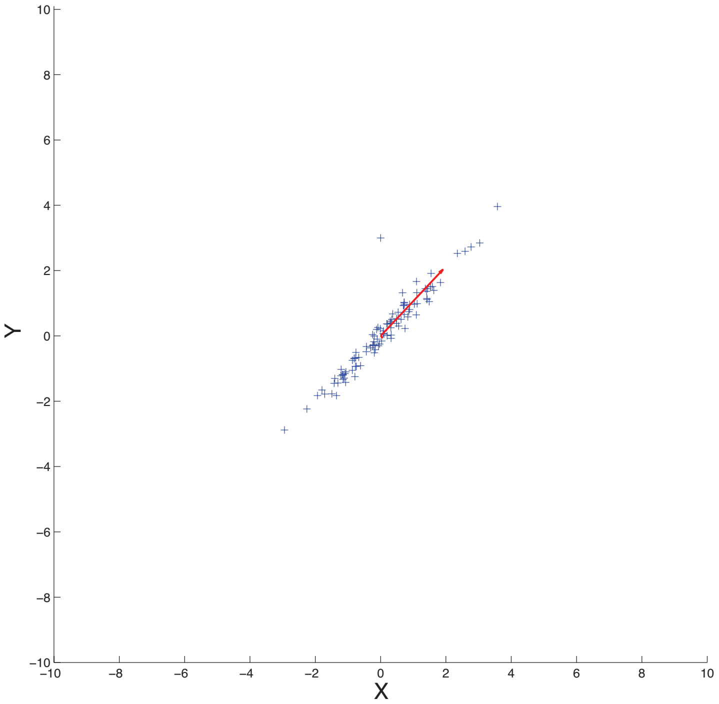
PCA is a well-known and commonly used technique for dimensionality reduction [Pearson 1901]. It consists of projecting the data onto the eigenvectors with biggest eigenvalues of the data covariance (or autocorrelation) matrix. In spite of its popularity and good performance, PCA presents two main problems: computational cost and sensitivity to outliers (see Figure 3.1 and Figure 3.2 for a low-dimensional example of the effect of outliers).

The computational cost can be split by considering data incrementally [Levy 2000, Brand 2006, Ross 2008] (see Section 2.4 for further details). This way, instead of computing a big PCA on a big data matrix, a PCA is performed on a small sub-matrix. This PCA is afterwards updated with blocks of the remaining data. Incremental procedures are also interesting when the whole dataset is not available at the beginning. In [Levy 2000], the sequential computation of PCA is tackled by updating an existing PCA with the components of the new data that are orthogonal to the previously generated subspace. Indeed, the process starts by computing, for the first block of data, its Singular Value Decomposition (SVD), which is an efficient way of computing the principal components of a matrix. Then, for each new block of data, the update process is based on a QR factorisation and a SVD of a small matrix. Their results, correctly combined, provide the principal components of the concatenation of the old and the new data matrices.

With respect to outliers, two strategies arise in a PCA computation procedure for minimising their effect on principal components. The first option is to discard samples that are supposed to be outliers. This strategy can be seen as a hard decision strategy, since samples labeled as outliers are not used at all. This forces to have a good outlier detection approach, in order to do not discard good samples. For instance, in [Jackson 2004] a minimum volume ellipsoid is fitted to data in order to discard, in the PCA computation, the samples that are outside; in [Hubert 2005] and [Hubert 2009], samples are discarded according to their projection in a subspace computed using a robust covariance estimation; and in [Zhou 2010], a convex optimisation procedure is performed to recover noise-free principal components. Methods using this hard decision strategy are dependent on

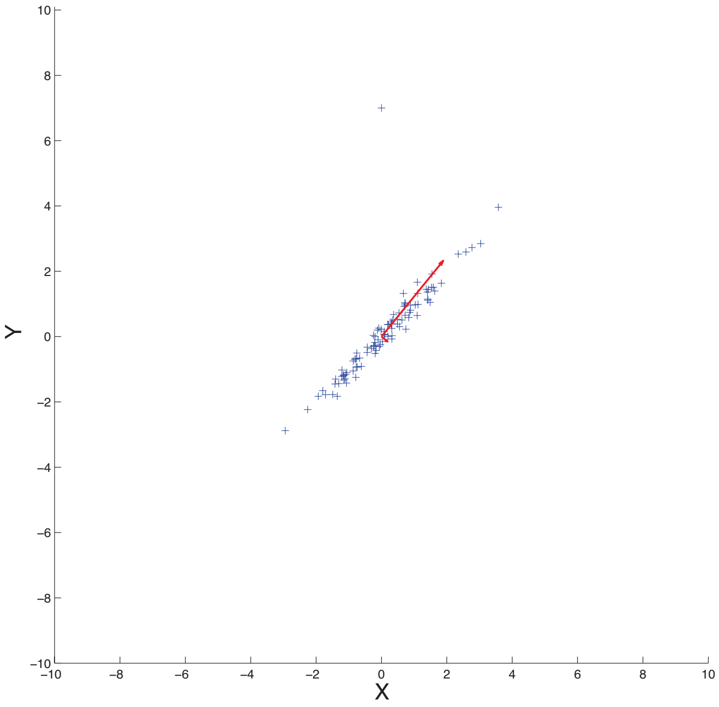


(a) 100 data points without outliers. The principal components are $v_1 = (0.70, 0.72)$ and $v_2 = (0.72, -0.70)$ with eigenvalues $\lambda_1 = 2.77$ and $\lambda_2 = 0.02$

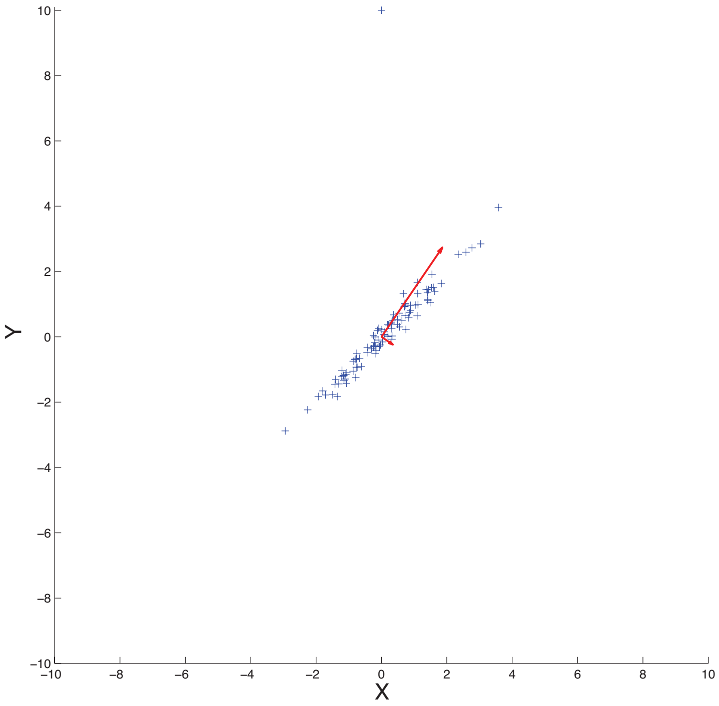


(b) 100 correct data points and one outlier at $(0, 3)$. The principal components are $v_1 = (0.68, 0.73)$ and $v_2 = (0.72, -0.68)$ with eigenvalues $\lambda_1 = 2.78$ and $\lambda_2 = 0.07$

Figure 3.1: Two-dimensional example of the effect on PCA of a single outlier against 100 correct samples. Each blue cross indicates a sample point. The red arrows are the principal components normalised to the value of the corresponding eigenvalue.



(a) 100 correct data points and one outlier at $(0, 7)$. The principal components are $v_1 = (0.63, 0.78)$ and $v_2 = (0.78, -0.63)$ with eigenvalues $\lambda_1 = 2.99$ and $\lambda_2 = 0.24$



(b) 100 correct data points and one outlier at $(0, 10)$. The principal components are $v_1 = (0.56, 0.83)$ and $v_2 = (0.83, -0.56)$ with eigenvalues $\lambda_1 = 3.31$ and $\lambda_2 = 0.42$

Figure 3.2: Two-dimensional example of the effect on PCA of a single outlier against 100 correct samples. Each blue cross indicates a sample point. The red arrows are the principal components normalised to the value of the corresponding eigenvalue.

the outlier detection method, and their performance strongly depends on it.

The second option is to weight the contribution of each value of the data matrix according to a measure of confidence. This is a soft decision strategy, in the sense that all the values in the data matrix are considered, although the contribution of the less confident ones is reduced. In this kind of approaches, the PCA computation is still dependent on the outlier detection method, but its dependency is considerably weakened. In [Kriegel 2008], the authors use a weighted covariance matrix for computing the principal components. The weight values are computed using a distance function to clusters of points of the dataset. In [Skočaj 2007], two kinds of weights are considered, temporal weights and spatial weights. Temporal weights adjust the contribution of each observation (a column in a data matrix), while spatial weights adjust the contribution of each variable (individual elements of each column). In [Skočaj 2008], the authors introduce a weighted incremental PCA algorithm based on the IPCA algorithm developed in [Hall 1998]. This algorithm updates the PCA for every new sample, using an EigenValue Decomposition (EVD), which is considerably slower than using SVD and updating the PCA per blocks of new samples, as shown in [Huang 2009].

In this Chapter we introduce an Incremental Temporally Weighted Principal Component Analysis (ITWPCA) algorithm based on SVD update, in Section 3.2. This algorithm allows to compute incrementally a robust low dimensional subspace representation (model). The robustness is based on the capacity of weighting the contribution of each single sample to the subspace generation. A spatial penalty approach has been also introduced in Section 3.3. This spatial penalty allows to assign more importance to some regions of a tracked object, penalising more the errors there, and increasing therefore the accuracy of tracking in those regions. A VT algorithm, based on the combination of the ITWPCA algorithm, the spatial weights and a particle filter, has been developed and introduced in Section 3.4. This algorithm allows to reduce the effect of bad tracked or bad quality samples on the appearance model that is computed simultaneously with tracking, at the same time that gives more importance to some regions of the tracked object. Finally, in Section 3.5, all the algorithms introduced in this Chapter are tested. A journal paper on the works presented in this Chapter is being prepared and will be submitted soon.

3.2 Incremental PCA with Weighted Samples: the ITWPCA Algorithm

In Section 2.4, the IPCA algorithm with mean update introduced in [Ross 2008] has been described. Here, we propose a weighted version of this algorithm. The weights modulate the contribution of data samples (columns of the data matrix) to the computed PCA. The same notation than in Section 2.4 is considered.

Given a set of N data samples $Z = [z_1, \dots, z_N] \in \mathbb{R}^{M \times N}$, where each sample is represented as a vector $z \in \mathbb{R}^M$, and a weight matrix with positive elements $\Omega \in \mathbb{R}^{M \times N}$, the goal of weighted PCA is to compute the projection matrix $U \in \mathbb{R}^{M \times K}$, $K \leq N$, that minimises the weighted squared reconstruction error

$$\widetilde{\xi} = \sum_{i=1}^{M} \sum_{j=1}^{N} \omega_{ij} \left(\widehat{z}_{ij} - \sum_{p=1}^{K} u_{ip} \sum_{q=1}^{M} u_{qp} \widehat{z}_{qj} \right)^2 . \tag{3.1}$$

The elements of the temporally weighted mean vector, μ_i, are computed as

$$\mu_i = \frac{1}{\sum_{j=1}^{N} \omega_{ij}} \sum_{j=1}^{N} \omega_{ij} z_j , \tag{3.2}$$

and so $\mu = [\mu_1, \dots, \mu_M]^\top$.

If only temporal weights are considered, i.e. $\omega_{ij} = \omega_{kj}\ \forall i, k \in [1, \dots, M], j \in [1, \dots, N]$, then the weights can be expressed by a vector ${}^t\omega = [\omega_1, \dots, \omega_N] \in \mathbb{R}^N$ and Equation (3.1) can be rewritten as

$$\widetilde{\xi} = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\widetilde{\widehat{z}}_{ij} - \sum_{p=1}^{K} u_{ip} \sum_{q=1}^{M} u_{qp} \widetilde{\widehat{z}}_{qj} \right)^2 , \tag{3.3}$$

where $\widetilde{\widehat{z}}_{ij} = \sqrt{\omega_j}\,\widehat{z}_{ij}$. Then, the matrix U that minimises $\widetilde{\xi}$ is composed by the K biggest eigenvalues of the covariance matrix of $\widetilde{\widehat{Z}}$, and can be computed by performing Singular Value Decomposition on this matrix, i.e. $\text{SVD}(\widetilde{\widehat{Z}}) = U \Sigma V^\top$, as introduced in [Skočaj 2007].

For introducing the incremental version, let us first note that a scatter temporally weighted matrix S_Z, defined as

$$S_Z = \sum_{i=1}^{N} \omega_i (z_i - \mu)(z_i - \mu)^\top , \tag{3.4}$$

differs from the weighted covariance matrix by only a scalar multiple, equal to $\sum_{i=1}^{N} \omega_i$. Therefore, eigenvectors of both matrices are the same and eigenvalues are scaled by this scalar multiple. This makes equivalent to work with the covariance matrix or the scatter matrix, in terms of PCA. Let us now introduce the following lemma:

Lemma 1. *Let* $Z^{(1)} = [z_1^{(1)}, \dots, z_{N^{(1)}}^{(1)}]$ *and* $Z^{(2)} = [z_1^{(2)}, \dots, z_{N^{(2)}}^{(2)}]$ *be two data matrices;* ${}^t\omega^{(1)} = [\omega_1^{(1)}, \dots, \omega_{N^{(1)}}^{(1)}]$ *and* ${}^t\omega^{(2)} = [\omega_1^{(2)}, \dots, \omega_{N^{(2)}}^{(2)}]$ *the weights corresponding*

to each sample in $Z^{(1)}$ and $Z^{(2)}$, respectively; $Z^{(1,2)} = [Z^{(1)} Z^{(2)}]$ the concatenation of matrices $Z^{(1)}$ and $Z^{(2)}$; and $\mu^{(1)}$, $\mu^{(2)}$ and $\mu^{(1,2)}$ the weighted means according to ${}^t\omega^{(1)}$ and ${}^t\omega^{(2)}$ of $Z^{(1)}$, $Z^{(2)}$ and $Z^{(1,2)}$, respectively.
Then, the weighted scatter matrix of $Z^{(1,2)}$, $S_{Z^{(1,2)}}$, can be computed as

$$S_{Z^{(1,2)}} = S_{Z^{(1)}} + S_{Z^{(2)}} + \frac{\|{}^t\omega^{(1)}\|_1 \|{}^t\omega^{(2)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1} (\mu^{(1)} - \mu^{(2)})(\mu^{(1)} - \mu^{(2)})^\top, \tag{3.5}$$

where $S_{Z^{(1)}}$ and $S_{Z^{(2)}}$ are the weighted scatter matrices of $Z^{(1)}$ and $Z^{(2)}$, respectively, and $\|\cdot\|_1$ denotes the 1-norm.

Proof. Note that

$$\mu^{(1,2)} = \frac{\|{}^t\omega^{(1)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1} \mu^{(1)} + \frac{\|{}^t\omega^{(2)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1} \mu^{(2)},$$

and so

$$\mu^{(1)} - \mu^{(1,2)} = \frac{\|{}^t\omega^{(2)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1} (\mu^{(1)} - \mu^{(2)}), \tag{3.6}$$

and

$$\mu^{(2)} - \mu^{(1,2)} = \frac{\|{}^t\omega^{(1)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1} (\mu^{(2)} - \mu^{(1)}). \tag{3.7}$$

Then,

$$\begin{aligned}
S_{Z^{(1,2)}} &= \sum_{i=1}^{N^{(1)}} \omega_i^{(1)} (z_i^{(1)} - \mu^{(1,2)})(z_i^{(1)} - \mu^{(1,2)})^\top \\
&+ \sum_{i=1}^{N^{(2)}} \omega_i^{(2)} (z_i^{(2)} - \mu^{(1,2)})(z_i^{(2)} - \mu^{(1,2)})^\top \\
&= \sum_{i=1}^{N^{(1)}} \omega_i^{(1)} (z_i^{(1)} - \mu^{(1)} + \mu^{(1)} - \mu^{(1,2)})(z_i^{(1)} - \mu^{(1)} + \mu^{(1)} - \mu^{(1,2)})^\top \\
&+ \sum_{i=1}^{N^{(2)}} \omega_i^{(2)} (z_i^{(2)} - \mu^{(2)} + \mu^{(2)} - \mu^{(1,2)})(z_i^{(2)} - \mu^{(2)} + \mu^{(2)} - \mu^{(1,2)})^\top \\
&= S_{Z^{(1)}} + S_{Z^{(2)}} + \sum_{i=1}^{N^{(1)}} \omega_i^{(1)} (\mu^{(1)} - \mu^{(1,2)})(\mu^{(1)} - \mu^{(1,2)})^\top \\
&+ \sum_{i=1}^{N^{(2)}} \omega_i^{(2)} (\mu^{(2)} - \mu^{(1,2)})(\mu^{(2)} - \mu^{(1,2)})^\top
\end{aligned} \tag{3.8}$$

Applying Equations (3.6) and (3.7) on Equation (3.8), we obtain

$$S_{Z^{(1,2)}} = S_{Z^{(1)}} + S_{Z^{(2)}} + \frac{\|{}^t\omega^{(1)}\|_1 \|{}^t\omega^{(2)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1} (\mu^{(1)} - \mu^{(2)})(\mu^{(1)} - \mu^{(2)})^\top$$

□

The results of Lemma 1 tell us how to express the temporally weighted scatter matrix of a big matrix by means of the weighted scatter matrices of two sub-matrices. Basically, the new scatter matrix is the sum of the other two, plus a rank-1 perturbation that depends on the difference of means. This rank-1 perturbation can be taken into account by adding a new column to the data matrix, as expressed in the Incremental Temporally Weighted PCA (ITWPCA) algorithm described in Algorithm 4. Note that the algorithm presented in [Ross 2008] can be seen as a particular case of the algorithm presented here, fixing $\omega_i^{(1)} = \omega_j^{(2)} = 1, \ \forall i, j$. Furthermore, the complexity of the procedure is almost unchanged. Indeed, only $M \times N^{(2)}$ additional multiplications and $N^{(2)}$ additional sums, whose computation time is negligible, are performed in order to consider the weights.

Algorithm 4 Incremental Temporally Weighted PCA (ITWPCA). Given $U^{(1)}$, $\Sigma^{(1)}$, $\|{}^t\omega^{(1)}\|_1$, a forgetting factor f, and a new data matrix $Z^{(2)}$, with its corresponding weights ${}^t\omega^{(2)}$, ITWPCA computes $U^{(1,2)}$ and $\Sigma^{(1,2)}$ from the total set of data.

```
Compute $\mu^{(2)} = \frac{1}{\|{}^t\omega^{(2)}\|_1} \sum_{i=1}^{N^{(2)}} \omega_i^{(2)} z_i^{(2)}$ and $\mu^{(1,2)} = \frac{f\|{}^t\omega^{(1)}\|_1}{f\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1}\mu^{(1)} + \frac{\|{}^t\omega^{(2)}\|_1}{f\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1}\mu^{(2)}$
Compute $\widetilde{\widehat{Z}}^{(2)} = [\sqrt{\omega_1^{(2)}}(x_1^{(2)} - \mu^{(2)}), \dots, \sqrt{\omega_{N^{(2)}}^{(2)}}(x_{N^{(2)}}^{(2)} - \mu^{(2)}), \sqrt{\frac{\|{}^t\omega^{(1)}\|_1 \|{}^t\omega^{(2)}\|_1}{\|{}^t\omega^{(1)}\|_1 + \|{}^t\omega^{(2)}\|_1}}(\mu^{(1)} - \mu^{(2)})]$
Compute ${}^{\perp}\widetilde{\widehat{Z}}^{(2)} = orth(\widetilde{\widehat{Z}}^{(2)} - U^{(1)}U^{(1)\top}\widetilde{\widehat{Z}}^{(2)})$
Compute $R = \begin{bmatrix} f\Sigma^{(1)} & U^{(1)\top}\widetilde{\widehat{Z}}^{(2)} \\ 0 & {}^{\perp}\widetilde{\widehat{Z}}^{(2)\top}(\widetilde{\widehat{Z}}^{(2)} - U^{(1)}U^{(1)\top}\widetilde{\widehat{Z}}^{(2)}) \end{bmatrix}$
Compute $SVD(R) = U'\Sigma'V'^{\top}$
Then, $U^{(1,2)} = [U^{(1)} \ {}^{\perp}\widetilde{\widehat{Z}}^{(2)}]U'$ and $\Sigma^{(1,2)} = \Sigma'$.
```

3.3 What about Weighted Variables in IPCA and VT?

In some applications, weights applied to variables, i.e. to rows, can be also useful. For instance, if each variable corresponds to data captured by sensors with different accuracies, weights applied to these variables can be used for giving more importance to the more accurate sensors. In a VT application, variables are pixels, and weights applied to them can be also interesting for giving more importance to some regions of the object of interest. We call weights applied to variables, spatial weights.

Temporal weights can be applied by preprocessing the data matrix as expressed in

Equation (3.3). This is not the case for spatial weights. Let us show why. Consider a vector of spatial weights ${}^{s}\omega = [\omega_1, \ldots, \omega_M] \in \mathbb{R}^M$, where at least two weights are different. Without loss of generality, suppose that $\omega_1 \neq \omega_2$, then

$$\begin{aligned} \widetilde{\xi} &= \sum_{i=1}^{M}\sum_{j=1}^{N} \omega_i \left(\widehat{z}_{ij} - \sum_{p=1}^{K} u_{ip} \sum_{q=1}^{M} u_{qp} \widehat{z}_{qj} \right)^2 \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N} \left(\sqrt{\omega_i}\widehat{z}_{ij} - \sum_{p=1}^{K} u_{ip} \sum_{q=1}^{M} u_{qp} \sqrt{\omega_i}\widehat{z}_{qj} \right)^2 \end{aligned} \tag{3.9}$$

From Equation (3.9) it can be observed, for instance, that ω_1 multiplies $\widehat{z}_{1j}$ $\forall j$ and ω_2 multiplies $\widehat{z}_{qj}$ $\forall q, j$. Therefore, $\widehat{z}_{1j}$ is multiplied by ω_1 in some terms of Equation (3.9) and by ω_2 in some others, which does not allow a preprocessing of the data matrix, similar to the temporal case, for considering these weights.

The impossibility of preprocessing the data matrix for considering spatial weights is the reason why, in [Skočaj 2007], the authors use an Expectation-Maximisation (EM) approach. In our case, as computational time is important and needs to be limited, an iterative approach as EM is not a good idea. Therefore, an adaptation of the EM approach for incremental spatially weighted PCA is not suitable.

Nevertheless, thinking in a VT application, the sensor of every variable (pixel sensors) are supposed to be identical. Moreover, the really important thing is the accuracy in the tracking of certain regions of the object of interest, not the accuracy of the model for these regions. For instance, if a face is being tracked, special care must be taken in order to correctly track the regions containing more information (eyes, nose and mouth), but a correct delimitation of the cheek is not as important, in general.

A spatial weighting strategy in the PCA computation on these regions is not necessary for increasing the accuracy of tracking. Furthermore, this spatial weighting would not be easily interpretable, since this weighting would give more importance to some pixel sensors than to others, while all pixel sensors in an image sensor are supposed to be identical. Instead, a higher tracking accuracy can be achieved by penalising the contribution of these pixels to the distances in Equations (2.24) and (2.25), i.e. by applying a spatial penalty to hypothesis.

Let us define a vector of positive values ${}^{s}\omega \in \mathbb{R}^M$ as the desired spatial weights, i.e. pixel weights. The higher the value applied to a pixel, the higher the penalty applied to this pixel and therefore more importance assigned to this pixel, since hypothesis fitting better these more penalised pixels will be favoured. Thus, let us redefine Equations (2.24) and (2.25) by considering spatial weights as

$$\widetilde{d_t} = \frac{1}{\sigma^2}(z-\mu)^\top diag({}^s\omega)(I-UU^\top)diag({}^s\omega)(z-\mu), \tag{3.10}$$

$$\widetilde{d_U} = (z-\mu)^\top diag({}^s\omega)U\Sigma^{-1}U^\top diag({}^s\omega)(z-\mu), \tag{3.11}$$

where $diag({}^s\omega)$ is a diagonal matrix with the spatial weights and σ^2 is defined in Equation (2.26). Equation (3.10) computes a weighted Euclidean distance to the subspace generated by the PCA, while Equation (3.11) computes a weighted Mahalanobis distance within this subspace.

Using these equations for computing particle weights considers the importance given beforehand to every pixel of the tracked region. This implies that the values of individual pixels of every hypothesis have different importance in the computation of the particle weight. However, an important thing to take into account is that an excessive increase of the weight applied to certain pixels can render the tracking algorithm unstable (as will be shown in Section 3.5). In the following Section, a complete VT algorithm with temporal weight for model computation and spatial weights for tracking is introduced.

3.4 Temporal and Spatial Weights in Visual Tracking

In Section 3.2, an incremental PCA algorithm capable of considering weights for samples has been introduced. This algorithm allows to build, in a VT application, a model of the object of interest while tracking, giving more importance to the more confident samples in order to avoid model corruption. From the VT algorithm described in Section 2.4, a measure of the quality of the tracked region for each video frame can be computed as follows.

Given a tracked patch expressed as a vector of pixel values $z \in \mathbb{R}^M$, the reconstruction error according to the PCA matrix at this time step gives information about the distance between the tracked patch and the subspace generated by the PCA. The difference between this patch and the PCA mean gives also information about how far is the new sample from the PCA subspace. Then, let us define the confidence on the tracked patch ω as

$$\omega = \begin{cases} 1 - \frac{\alpha}{M}\sum_{i=1}^{M} f(z_i,\varepsilon), & \text{if } \sum_{i=1}^{M} f(z_i,\varepsilon) \le \frac{M}{\alpha} \\ 0, & \text{otherwise,} \end{cases} \tag{3.12}$$

where $\varepsilon \in [0,1]$, $\alpha \in \mathbb{R}^+$ and two different options for $f(z_i,\varepsilon)$, namely

$$f(z_i,\varepsilon) = f_R(z_i,\varepsilon) = \begin{cases} 1, & \text{if } |(z_i-\mu_i)-\bar{z}_i| \ge \varepsilon \\ 0, & \text{otherwise,} \end{cases}, \tag{3.13}$$

and

$$f(z_i, \varepsilon) = f_M(z_i, \varepsilon) = \begin{cases} 1, & \text{if } |z_i - \mu_i| \geq \varepsilon \\ 0, & \text{otherwise}, \end{cases}, \tag{3.14}$$

being $\bar{z}_i$ the i-th component of the vector $\bar{z} = UU^\top(z - \mu)$. The measure proposed in Equation (3.12) gives more importance to the number of pixels with a significant error (ε) than to the amount of the error itself, weighting the contribution of the sample to the PCA according to this measure. Note that samples with more than $\frac{100}{\alpha}$% of the pixels with more than ε error are discarded ($\omega = 0$). This strategy tries to penalise samples containing big regions with a significant amount of error (reconstruction error or distance to the mean). Indeed, this is the typical situation in an occlusion. The neutral value of $\alpha = 2$ has been adopted in all the tests, i.e. samples with more than 50% of the pixels with a significant error are not considered in the PCA computation. Depending on the context of the application, this value can be increased, with the risk of being too restrictive and therefore becoming unadapted to the object of interest.

The confidence measure presented in Equation (3.12) is used in the VT algorithm introduced below for weighting samples supplied to the incremental PCA computation. Then, combining this temporal weighting with the spatial penalty introduced in Section 3.3, we can define the Incremental Temporally Weighted Visual Tracking with Spatial Penalty (ITWVTSP) algorithm. In Algorithm 5, a detailed description of the complete proposed visual tracking algorithm with incremental temporally weighted PCA and spatial error penalty is shown. Note that by fixing ${}^s\omega = \mathbb{1}_{M\times 1}$, we obtain what we call the Incremental Temporally Weighted Visual Tracking (ITWVT) algorithm. We denote by "/R" the use of Equation (3.13) and by "/M" the use of Equation (3.14), i.e. for instance we denote by ITWVT/M the ITWVTSP algorithm using $f_M(z, \varepsilon)$ and ${}^s\omega = \mathbb{1}_{M\times 1}$ for computing sample weights.

3.5 Tests and Results

We have performed several tests to our Incremental Temporally Weighted Visual Tracking (ITWVT) algorithm and our Incremental Temporally Weighted Visual Tracking with Spatial Penalty (ITWVTSP) algorithm, on several video sequences. For showing the improvement obtained by the weighting strategy, the results are compared with the results obtained by the IVT algorithm introduced in [Ross 2008][1]. For a general comparison against state-of-the-art algorithms, we compare also our results with the results obtained with the TLD algorithm introduced in [Kalal 2010][2].

[1] Implementation available at http://www.cs.toronto.edu/~dross/ivt/ (last visited in june 2011)

[2] Implementation available at http://info.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html (last visited in june 2011)

Algorithm 5 Incremental Temporally Weighted Visual Tracking with Spatial Penalty (ITWVTSP). The target region (image of the object in the first frame) is denoted by z_0; $N^{(2)}$ denotes the size of the processed blocks; and K denotes the maximum number of considered eigenvalues.

```
$\mu = z_0$, $n = 1$, and $U^{(1)}$, $\Sigma^{(1)}$, $Z^{(2)}$ and ${}^t\omega^{(2)}$ are empty
Set ${}^s\omega$ to the desired spatial weights (by default, ${}^s\omega = \mathbb{1}_{M\times 1}$)
for every frame of the video do
  Draw particles according to the dynamical model (Equation (2.22)) and the weight distribution of particles.
  For each particle, compute its weight according to the observation model and spatial weights (Equation (3.10) and Equation (3.11)).
  Store in $Z^{(2)}$ the image region corresponding to the most likely particle, and in ${}^t\omega^{(2)}$ its PCA weight (Equation (3.12))
  if there are $N^{(2)}$ stored images in $Z^{(2)}$ then
    if $n < K$ then
      ${}^t\omega_i^{(2)} = 1, \ \forall i = 1, \dots, N^{(2)}$
    Apply Algorithm 4 with $\|{}^t\omega^{(1)}\|_1 = n$, discarding the eigenvectors that exceed $K$.
    Set $U^{(1)} = U^{(1,2)}$, $\Sigma^{(1)} = \Sigma^{(1,2)}$ and $n = fn + \|{}^t\omega^{(2)}\|_1$
    Empty $Z^{(2)}$ and ${}^t\omega^{(2)}$
```

With IVT, ITWVT and ITWVTSP, we use the same parameters that those proposed in [Ross 2008], i.e. 600 particles, an eigenvector size of 32 × 32 pixels, a maximum number of 16 eigenvectors and a block update of 5 images. We only increase slightly the forgetting factor (from 0.95 to 0.97) since the temporal weights increase the quality of the model and a longer memory is beneficial. With these parameters, the tracker runs at 7 frames per second in a laptop with a 2.0GHz processor.

The standard deviations of the dynamical model (Equation (2.22)) in all the experiments are 9.0px for row and column displacements, 0.05 radians for rotation, 0.05 for scaling in the x direction, 0.001 for scaling in the y direction and 0.001 radians for the scaling angle defining x and y directions, which are similar values to those proposed in the implementation of IVT. By using the same parameters for the three algorithms, the performance improvement due to the temporal and spatial weighting strategy can be clearly perceived. With TLD, the standard parameters provided in the distributed implementation are used.

For visualisation of the tracking results, we use the same template as in [Ross 2008]: the first row contains the current frame with the tracked region, the second row contains the mean, the tracked window, the reconstruction error and the reconstructed image, and finally, the third and fourth rows contain the first ten

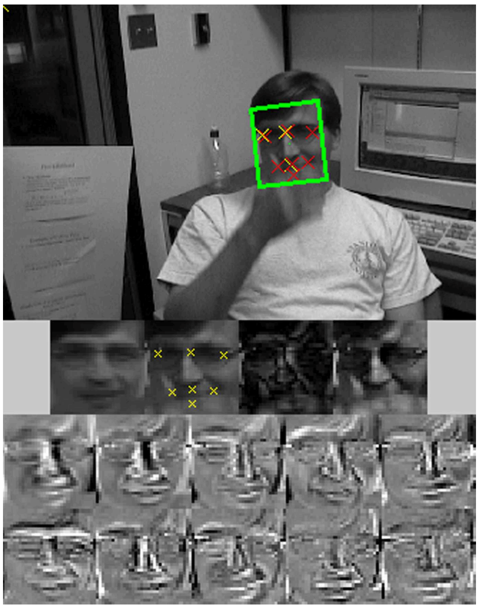


Figure 3.3: Frame of the Dudek sequence where an occlusion of the face is starting. First row contains the current frame with the tracked region. The second row contains the mean, the tracked window, the reconstruction error and the reconstructed image. Finally, the third and fourth rows contain the first ten eigenvalues.

eigenvalues. In Figure 3.3 an example is shown.

The performed experiments are divided in two groups. In the first group, there are experiments performed on labelled video sequences, i.e. video sequences with a ground truth. In these experiments, quantitative performance scores are computed to show the performance of the algorithms. In the second group, the proposed algorithms are applied to several unlabelled video sequences in a variety of tracking applications, to show the polyvalence of the algorithms.

3.5.1 Labelled Video Sequences

First, we have performed a tracking of the face in the Dudek sequence [Jepson 2003] (see Figure 3.3) with IVT, TLD, ITWVT and ITWVTSP. This sequence is a very challenging video with changes in the tracked object, the camera position and the illumination. The ground truth of 7 manually labelled points on the face is available for this sequence. This allows to compare quantitatively the obtained results, by computing the Root Mean Squared Error (RMSE) of the tracked points with respect to the real ones. Given the implicit stochasticity of the algorithms, ten independent runs per algorithm are performed in all the tests.

The ITWVT algorithm has been tested using Equation (3.13) and Equation (3.14), for 25 different values of ε between 0.01 and 0.9. In Figure 3.4(a), the results obtained with ITWVT/M are shown, where only runs with a RMSE lower than 10.0 pixels are plotted. In Figure 3.4(b), the results corresponding to the ITWVT/R algorithm are shown, plotting again only those with a RMSE lower than 10 pixels. Both variants of the algorithm provide similar results, with small variance among runs for $\varepsilon \in [0.02, 0.12]$. This is due to the fact that small values of ε produce low temporal weights, avoiding a good adaptation of the model to the tracked face, and big values of ε produce big weights, making the performance similar to IVT (in terms of RMSE and number of track losses). For $\varepsilon \in [0.02, 0.12]$, the compromise between good model adaptation and corruption avoidance seems to be satisfied for the Dudek sequence. The complete statistics of the runs can be found in Appendix A.

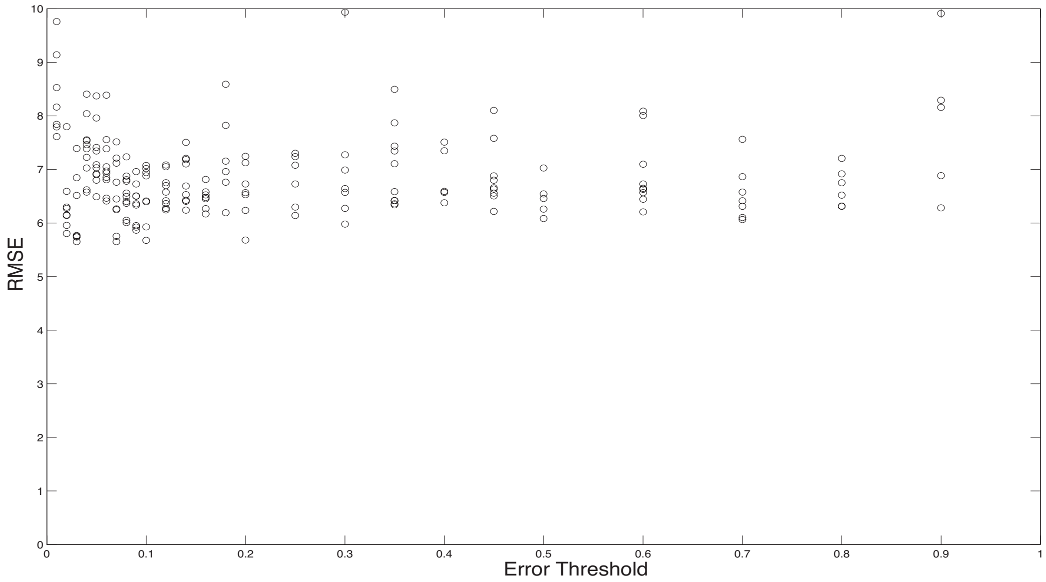
Looking at Figure 3.4, a reasonable value for the error threshold is $\varepsilon = 0.07$. For this value of ε, ITWVT/M obtains a RMSE = 5.6537px and ITWVT/R a RMSE = 5.8645px. The best run with IVT obtains a RMSE = 6.2324px, which shows that the use of temporal weights improves the performance of the tracking. In addition, only one out of the ten runs lost the track (RMSE higher than 10.0px) for both, ITWVT/M and ITWVT/R, while five are lost with IVT. This shows the improvement in the robustness of the tracking thanks to the better quality of the model.

Let us note that the value of ε is application-specific. Indeed, the appearance of a rigid object changes slightly, which allows to fix a more restrictive (smaller) ε. On the contrary, a deformable object like for instance a pedestrian, changes its appearance considerably, which forces to fix ε to higher values if we want to avoid unjustified small temporal sample weights. Faces are somehow in between highly deformable objects and rigid objects, which makes $\varepsilon = 0.07$ an appropriate candidate value when no information about the application is available.

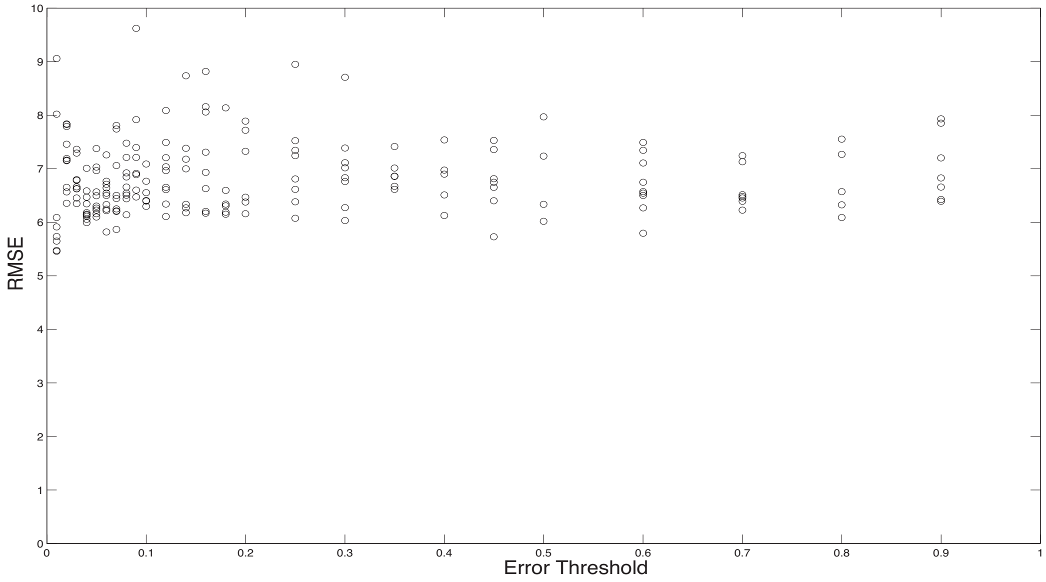
Performance of ITWVT/R and ITWVT/M are similar, although looking to the obtained temporal weights (see Figures 3.5(a) and 3.5(b)), those obtained using the reconstruction error are more robust. Indeed, only frames with an occlusion or high out-of-plane rotations of the face present a clearly reduced weight.

For observing the effect of spatial penalty, we have designed two ${}^{s}\omega$ vectors. The first one assigns higher weight values to pixels on important regions of the face (see Figure 3.6(a)), we call this the “spec” spatial weights and denote its use by “-spec”. The second one is a two-dimensional Gaussian shape, centred in the middle of the patch (see Figure 3.6(b)), we call this the “iso” spatial weights and denote its use by “-iso”.

In Figures 3.7(a), 3.7(b), 3.8(a) and 3.8(b) the results obtained varying the maxi-

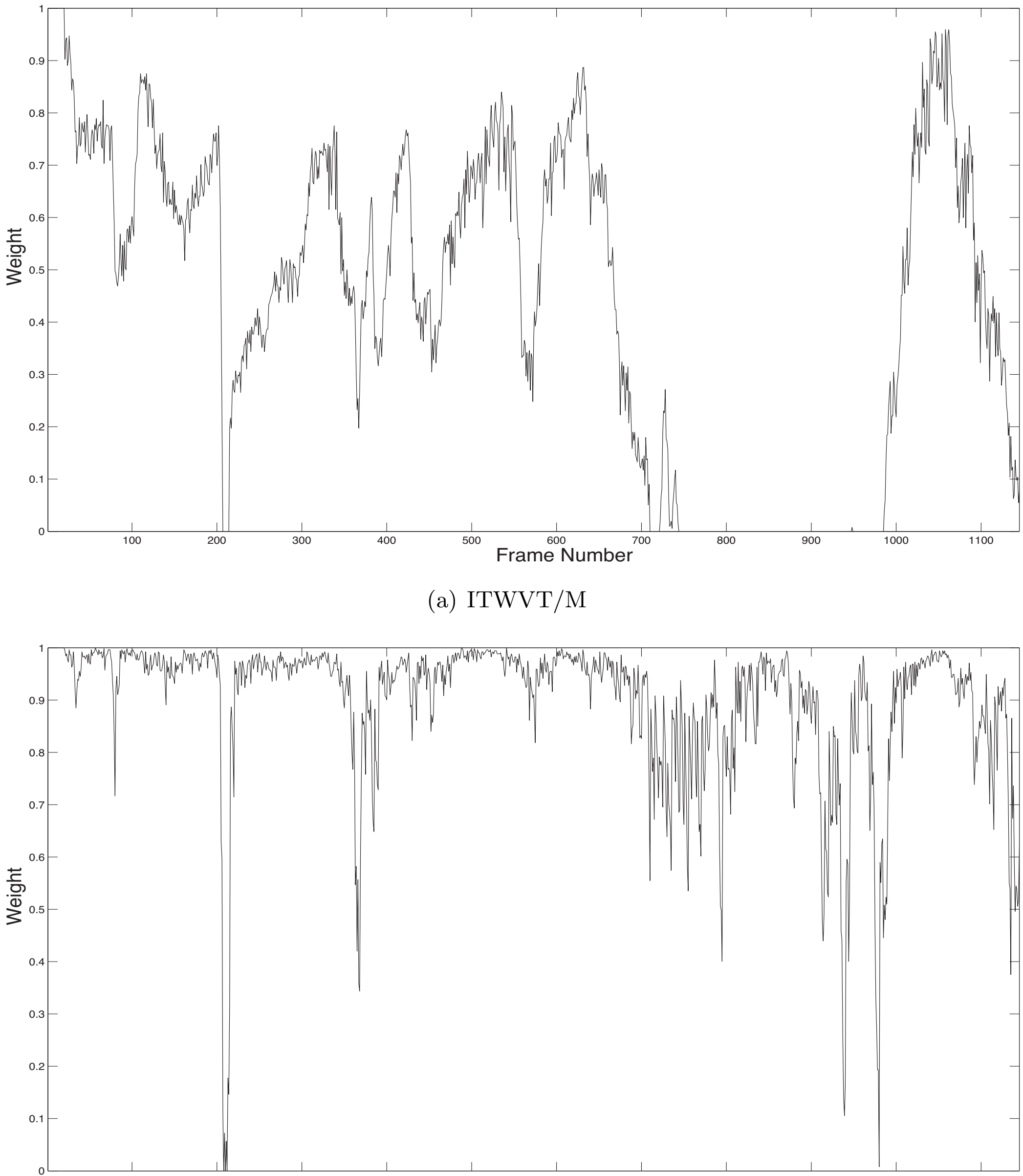


(a) ITWVT/M



(b) ITWVT/R

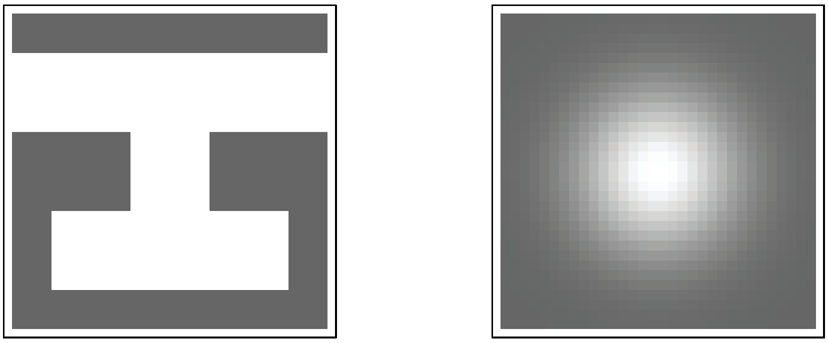
Figure 3.4: RMSE obtained with ITWVT on the Dudek sequence as a function of ε. For each value of ε, ten runs of the algorithm are executed. The values of RMSE higher than 10.0px are not plotted (considered as loss of track).



(a) ITWVT/M

(b) ITWVT/R

Figure 3.5: Weights applied to each tracked sample of the Dudek sequence using ITWVT/M and ITWVT/R with $\varepsilon = 0.07$. These weights correspond to the runs that gave the best RMSE value (5.6537px for ITWVT/M and 5.8645px for ITWVT/R). The frames that present an occlusion or the frames where the face is rotated out-of-the-plane are clearly noticeable (small weights).



(a) “Spec” spatial weights

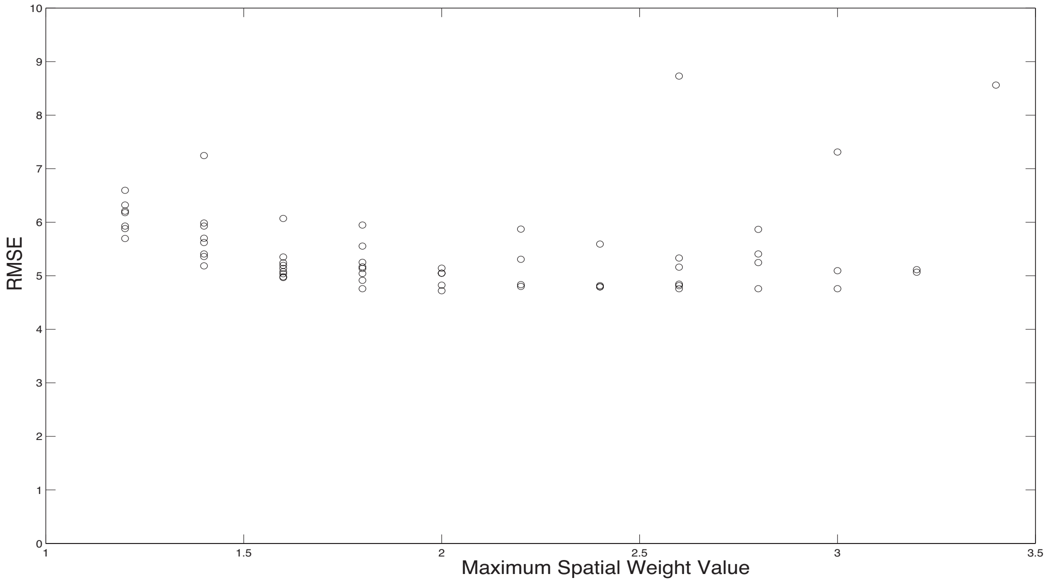
(b) “Iso” spatial weights

Figure 3.6: Spatial weights used in the experiments. Brighter regions correspond to high weight values, darker regions to spatial weights equal to 1.0.

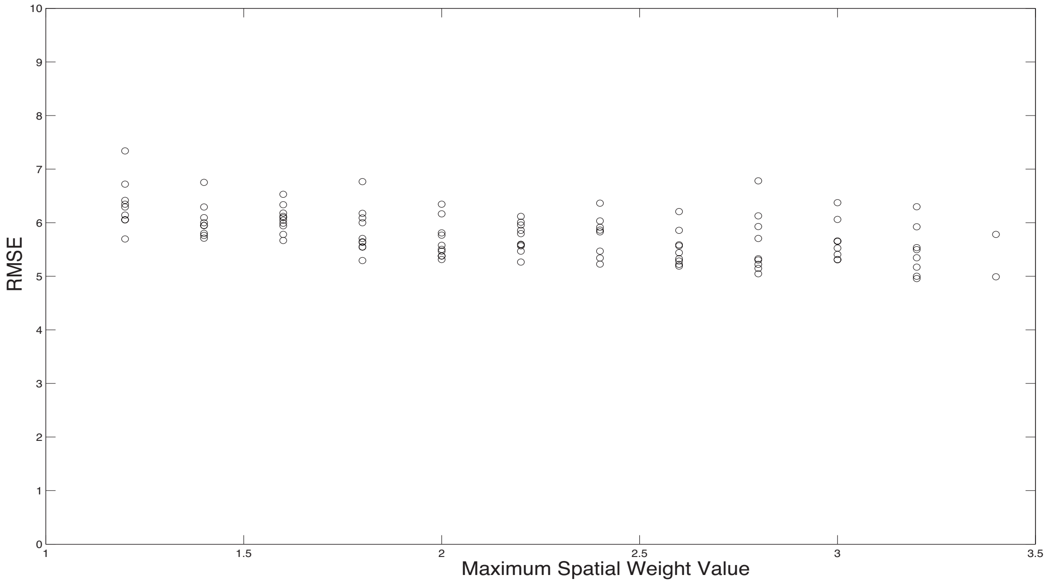
mum value of the spatial weight (${}^{s}\omega_{\max}$) using ITWVTSP/M-spec, ITWVTSP/M-iso, ITWVTSP/R-spec and ITWVTSP/R-iso respectively, are shown. As before, values of RMSE higher than 10.0px are considered as track losses and are not plotted. For values of ${}^{s}\omega_{\max} > 2.0$ using “spec”, the algorithm starts to be unstable, producing more losses of track than correct tracking among the ten performed runs. For the “iso” spatial weights, the gradual transition make the algorithm more stable allowing to go up to ${}^{s}\omega_{\max} = 3.2$. Looking at the results, good values for the maximum spatial weights are 1.8 for “spec” and 3.2 for “iso”, although smaller values can be used if we want to minimise the risk of track loss due to excessive spatial penalty. In Appendix A, the complete statistics of the runs are shown.

In Table 3.1, the best values of RMSE obtained with each algorithm for the parameter values commented above, are shown. As it can be observed, the ITWVTSP algorithm produces a considerable better tracking performance than IVT with an increased robustness (two out of ten track losses for ITWVTSP against five out of ten for IVT). In this sequence, TLD produces considerably higher RMSE than the other algorithms since it tends to enlarge or reduce the tracked region, which causes a displacement on the template of the tracked points and therefore a higher error. In Appendix B, plots of RMSE per frame for the best run of each algorithm are shown.

For testing the algorithms in a real situation with partial occlusions, we recorded the Rockstar sequence. In this sequence, composed of 171 frames, a subject in front of the camera is recorded. At a certain moment, the subject puts on a pair of sunglasses that he takes off later. This sunglasses generate an occlusion of the eyes of the subject, which is an important part of the face, clearly coded in the appearance model. The distance between the face of the subject and the camera, and therefore its size in the image, remains almost constant during the whole video. This allows to label the ground truth of the sequence by displacing

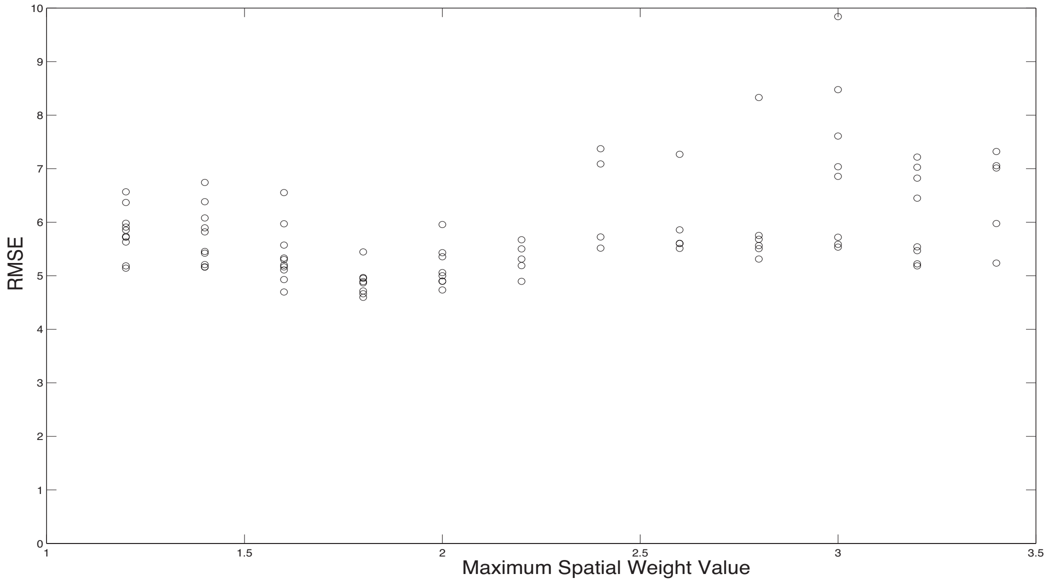


(a) ITWVTSP/M-spec

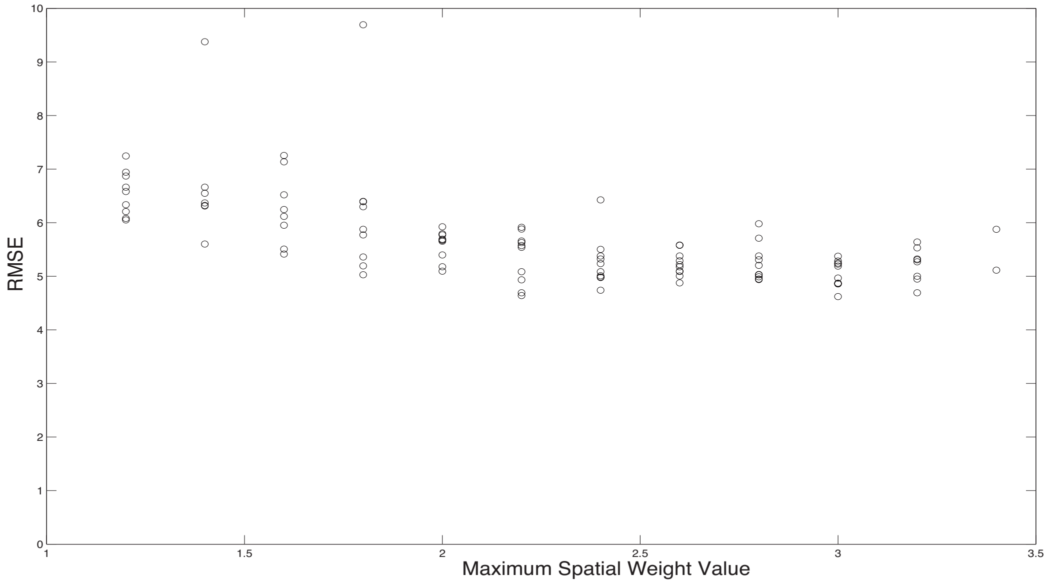


(b) ITWVTSP/M-iso

Figure 3.7: Obtained RMSE on the Dudek sequence varying ${}^{s}\omega_{\max}$ with ITWVTSP/M-spec and ITWVTSP/M-iso.



(a) ITWVTSP/R-spec



(b) ITWVTSP/R-iso

Figure 3.8: Obtained RMSE on the Dudek sequence varying ${}^{s}\omega_{\max}$ with ITWVTSP/R-spec and ITWVTSP/R-iso.

Algorithm	**Minimum RMSE**
TLD	13.6619
IVT	6.2324
ITWVT/R ($\varepsilon = 0.07$)	5.8645
ITWVT/M ($\varepsilon = 0.07$)	5.6537
ITWVTSP/M-iso ($\varepsilon = 0.07$, ${}^{s}\omega_{\max} = 3.2$)	4.9586
ITWVTSP/M-spec ($\varepsilon = 0.07$, ${}^{s}\omega_{\max} = 1.8$)	4.7596
ITWVTSP/R-iso ($\varepsilon = 0.07$, ${}^{s}\omega_{\max} = 3.2$)	4.6927
ITWVTSP/R-spec ($\varepsilon = 0.07$, ${}^{s}\omega_{\max} = 1.8$)	**4.5969**

Table 3.1: Best RMSE values for all the algorithms on the Dudek sequence.

the starting bounding box that contains the face, in order to keep eyes, nose and mouth centred along the whole video sequence. This has been done manually.

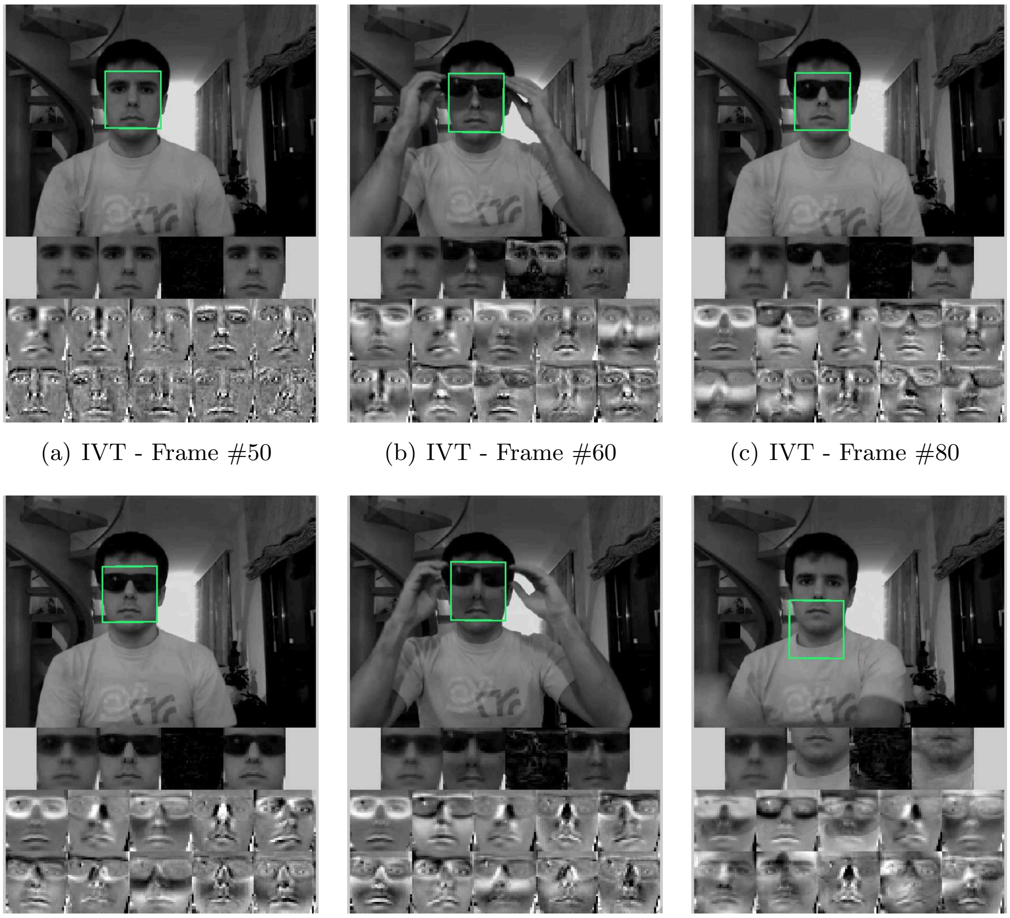
Ten runs of IVT, ITWVT, ITWVTSP and TLD have been performed on this sequence. For spatial weights, a conservative approach has been adopted, taking ${}^{s}\omega_{\max} = 2.0$ for "iso" and ${}^{s}\omega_{\max} = 1.6$ for "spec". Precision and lost track ratio scores have been computed for all the algorithms and the results are shown in Table 3.2. For computing precision score, the intersection over union criterion with a threshold value of 0.8 has been used. For the lost track ratio, dice error with a threshold value of 0.8 has been employed. The results show clearly the better performance of the family of algorithms introduced in this Chapter. However, the negative impact in this case of the spatial penalty can be observed too. Indeed, the persistence of the partial occlusion in an important region, in terms of spatial weights, seems to have a negative effect in the performance, although it is anyway better than with IVT and TLD. These two algorithm suffer from a displacement of the tracked region while the subject is wearing the sunglasses, which causes the bad precision and lost track ratio scores. Some selected frames of the best run using IVT, TLD and ITWVT/M are shown in Figure 3.9, Figure 3.10 and Figure 3.11, respectively.

3.5.2 Unlabelled Video Sequences

In Figure 3.12, several frames of the poster sequence are shown. In this sequence, a poster is recorded while several partial occlusions are generated. The total sequence is composed of 585 frames, and during the first 100 frames there are no occlusions. In order to see the effect of the temporal weights, we compute the deviation from the "correct" first eigenvector due to these occlusions in IVT, ITWVT/R and ITWVT/M. As "correct" eigenvectors we consider the eigenvectors at frame 100, with a forgetting factor fixed to 1.0 and the temporal weights up to frame 100 equal to 1.0. Note that the first eigenvector is the one with the highest

Algorithm	**Precision**				**Lost Track Ratio**			
	Best	**Worst**	**Mean**	**St.Dev.**	**Best**	**Worst**	**Mean**	**St.Dev.**
IVT	0.9064	0.3392	0.6526	0.1877	0.0	0.1637	0.0965	0.0831
TLD	0.6260	0.3493	0.5598	0.0833	0.1053	0.2398	0.1632	0.0671
ITWVT/R	0.9591	0.3392	0.8474	0.1859	**0.0**	**0.0117**	**0.0012**	**0.0037**
ITWVT/M	**0.9708**	**0.7661**	**0.9129**	**0.0739**	0.0	0.0760	0.0111	0.0241
ITWVTSP/M-iso	0.7895	0.3509	0.6018	0.1239	0.0	0.1579	0.0287	0.0518
ITWVTSP/M-spec	0.9532	0.7602	0.8111	0.0753	0.0	0.0877	0.0322	0.0391
ITWVTSP/R-iso	0.8187	0.3333	0.5468	0.1752	0.0	0.1579	0.0503	0.0744
ITWVTSP/R-spec	0.9825	0.6842	0.7754	0.0813	0.0	0.1637	0.0830	0.0733

Table 3.2: Obtained results on the Rockstar sequence. The parameters are $\varepsilon = 0.07$, ${}^{s}\omega_{\max} = 2.0$ for "iso" and ${}^{s}\omega_{\max} = 1.6$ for "spec".



(a) IVT - Frame #50

(b) IVT - Frame #60

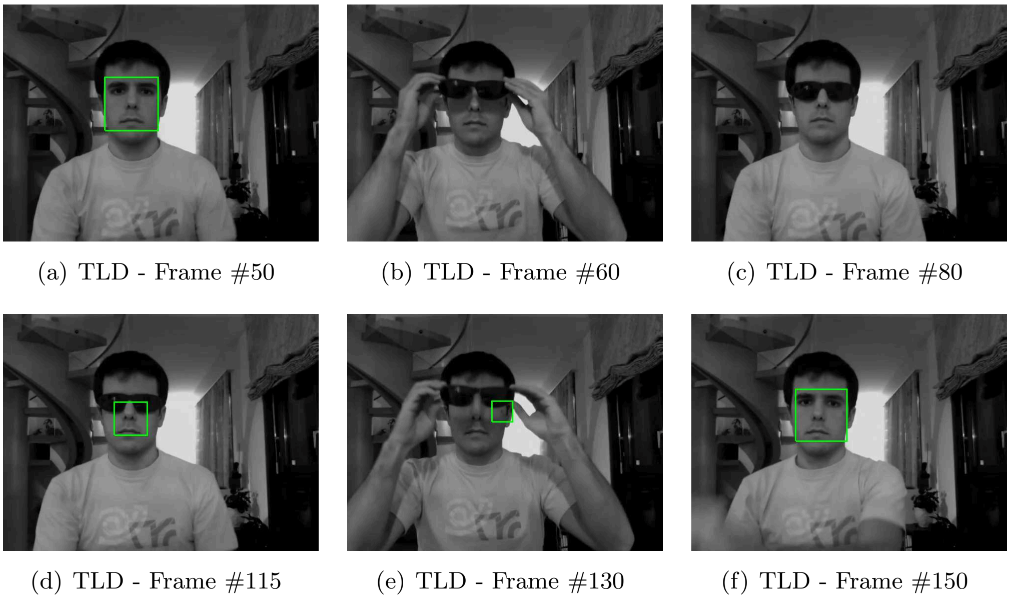
(c) IVT - Frame #80

(d) IVT - Frame #115

(e) IVT - Frame #130

(f) IVT - Frame #150

Figure 3.9: Results obtained with the IVT algorithm on the Rockstar sequence. The eigenvectors show how the sunglasses corrupt the model, avoiding a correct tracking continuation after taking them off.



(a) TLD - Frame #50

(b) TLD - Frame #60

(c) TLD - Frame #80

(d) TLD - Frame #115

(e) TLD - Frame #130

(f) TLD - Frame #150

Figure 3.10: Results obtained with the TLD algorithm on the Rockstar sequence.

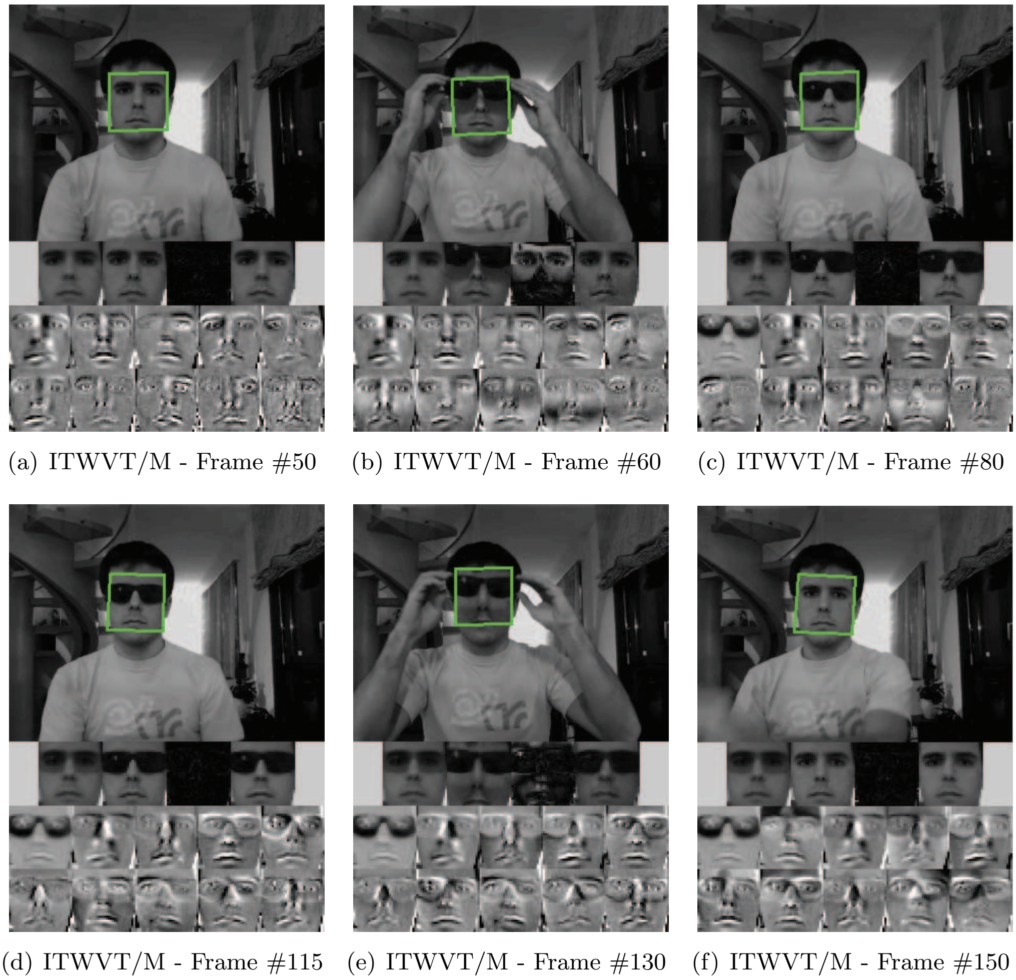
eigenvalue, and therefore the most important one for computing particle weights. The deviation is computed as the distance between the first “correct” eigenvector and the first eigenvectors given by each algorithm. In Figure 3.13, a plot of these distances is given, showing that ITWVT/R and ITWVT/M keep the eigenvectors closer to those before the occlusions start. As commented before, ITWVT/M produces smaller sample weights (see Figure 3.14), which makes the distances slightly smaller than with ITWVT/R.

Finally, to show the polyvalence of the algorithms presented here, we have performed several experiments in two other tracking applications: pedestrian tracking and vehicle tracking. The videos do not present particular difficulties in terms of partial occlusions, which makes that similar performances are obtained with all the algorithms (ITWVT and ITWVTSP). Here we show the results with ITWVTSP/R-iso.

In Figure 3.15, the tracking of a subject in sequence S1-T1-C of Camera 3 of the PETS2006 Dataset[3] is shown. Given the variability on the appearance of a pedestrian, mainly due to the legs, we use an “iso” spatial weighting strategy but with the Gaussian shape displaced toward the upper part of the patch. This gives more importance to the body of the pedestrian than to his legs. The maximum spatial weight used is ${}^{s}\omega_{\max} = 3.2$ and the noise threshold $\varepsilon = 0.12$.

In Figure 3.16 and Figure 3.17, a vehicle tracking is performed. In the first

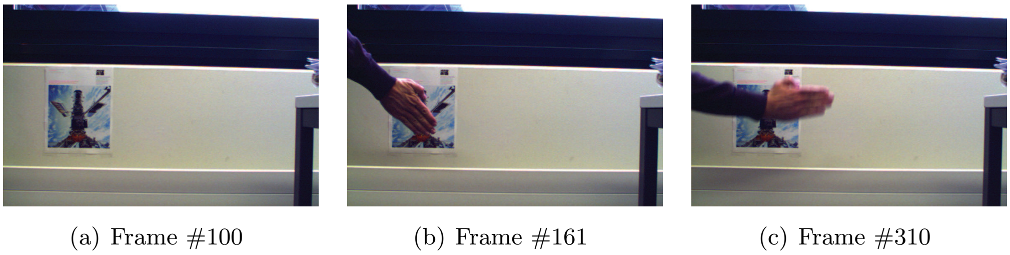
[3] Available at http://www.cvg.reading.ac.uk/PETS2006/data.html (last visited in june 2011)



(a) ITWVT/M - Frame #50 (b) ITWVT/M - Frame #60 (c) ITWVT/M - Frame #80

(d) ITWVT/M - Frame #115 (e) ITWVT/M - Frame #130 (f) ITWVT/M - Frame #150

Figure 3.11: Results obtained with the ITWVT/M algorithm. The tracking continues correctly after taking the sunglasses off.



(a) Frame #100 (b) Frame #161 (c) Frame #310

Figure 3.12: Several frames of the Poster sequence. The total sequence is composed of 585 frames.

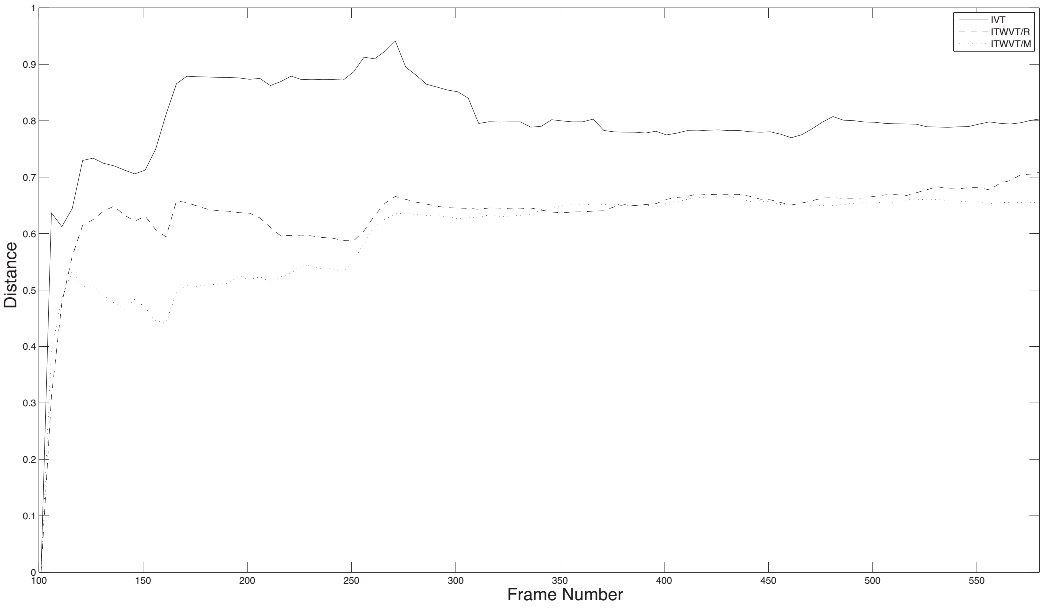


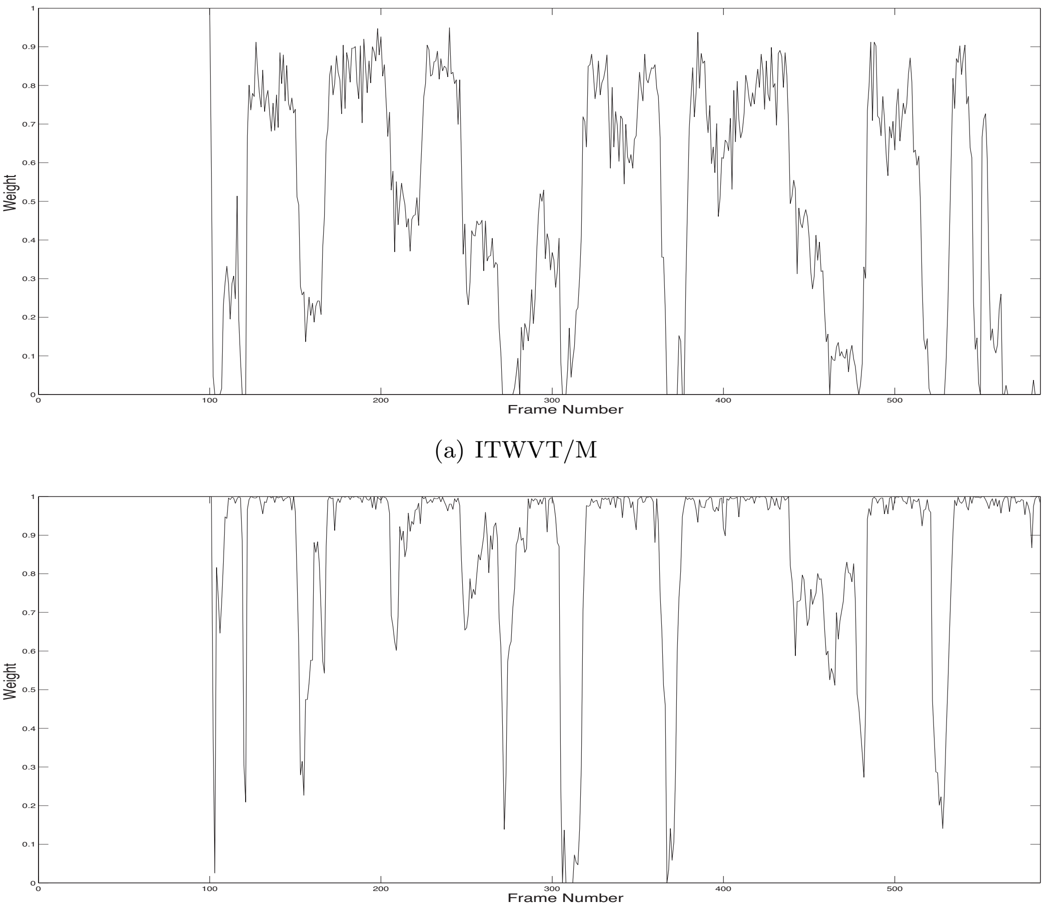
Figure 3.13: Distances between the first eigenvector at frame 100 and the first eigenvector computed using IVT (solid line), ITWVT/R (dashed line) and ITWVT/M (dotted line) at subsequent frames.

sequence, the tracked vehicle experiences extreme and sudden changes in its illumination, which can be observed in the temporal weights going to zero. In the second sequence, which runs at night, the illumination is considerably bad during the whole sequence, but without any significant variation of the conditions. This can also be observed in the weights, which are around the same values during the whole sequence.

3.6 Conclusions and Future Work

In this Chapter we have introduced an incremental PCA algorithm with weighted samples (ITWPCA). This algorithm can be used in any application requiring an incremental computation of a PCA, due to either computational requirements or the lack of the whole dataset at the beginning. The capacity of this algorithm for weighting the contribution of samples can be used for minimising the impact of outliers in the computed PCA.

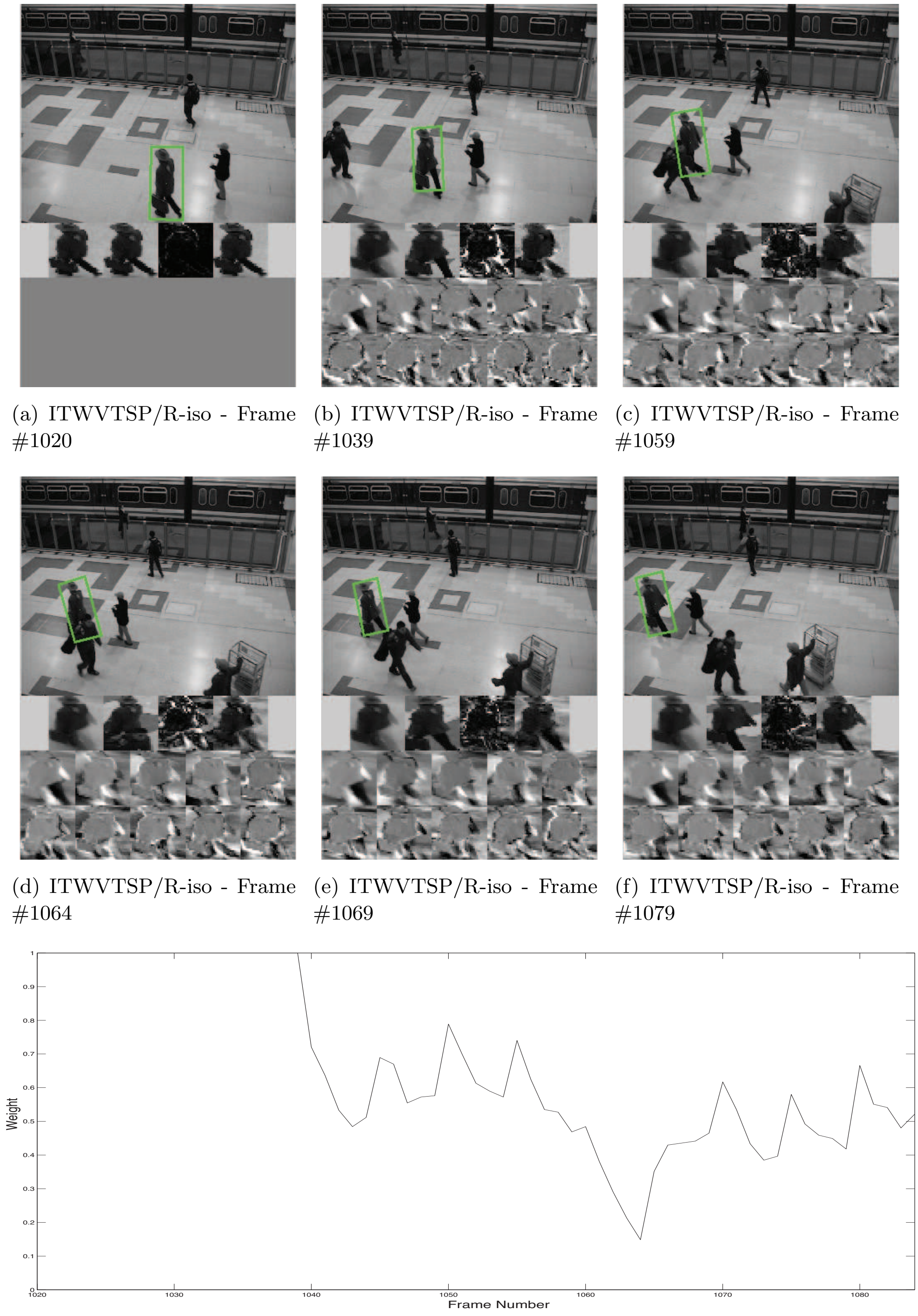
The preprocessing strategy used for considering temporal weights cannot be used for considering spatial weights, as has been shown. However, in the context of VT, a strategy for considering spatial weights has been introduced. Both types of weights, temporal and spatial, have been combined in a VT algorithm, based



(a) ITWVT/M

(b) ITWVT/R

Figure 3.14: Weights applied to the samples of the poster in the Poster sequence using ITWVT/M and ITWVT/R.



(a) ITWVTSP/R-iso - Frame #1020

(b) ITWVTSP/R-iso - Frame #1039

(c) ITWVTSP/R-iso - Frame #1059

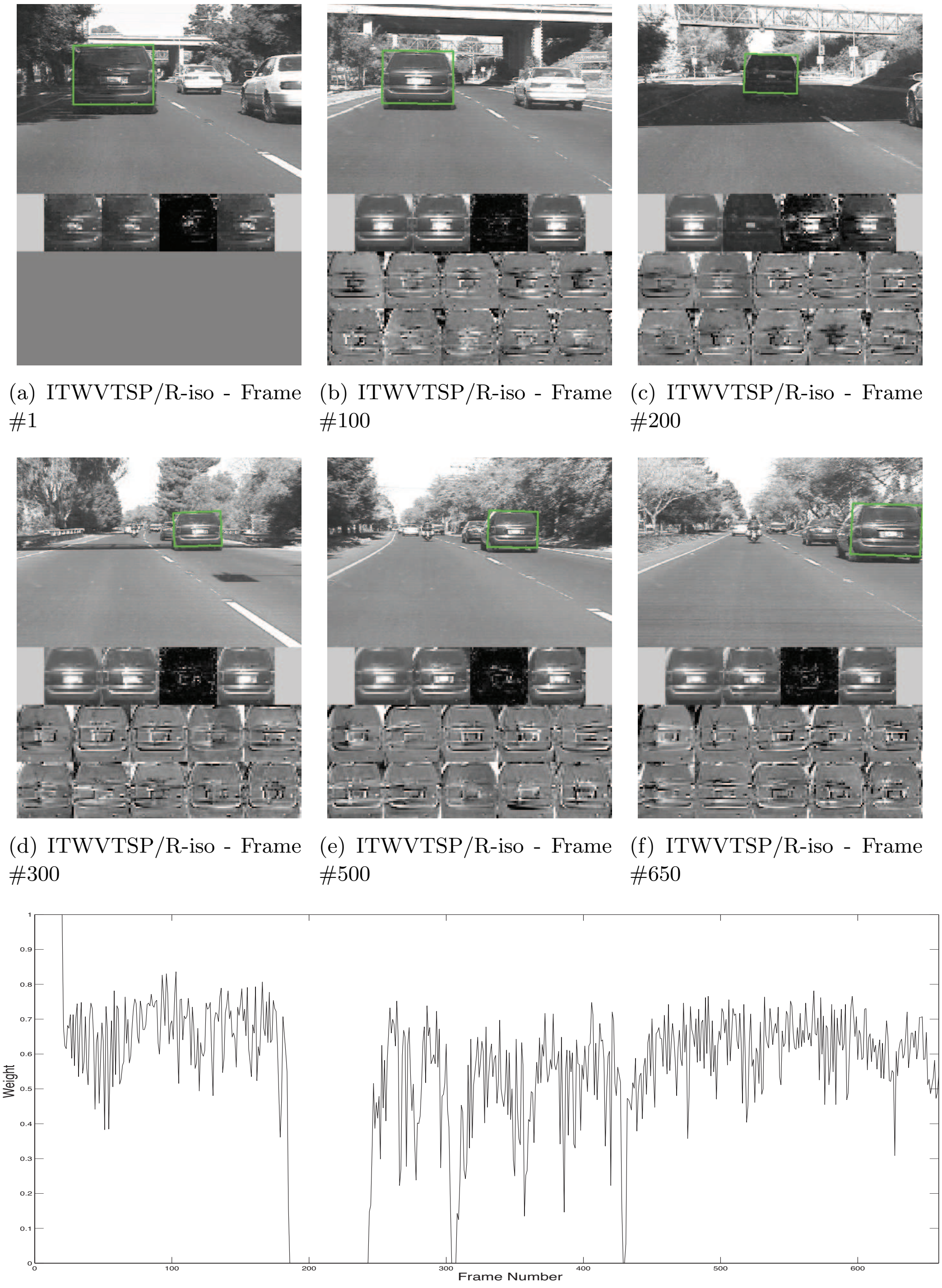
(d) ITWVTSP/R-iso - Frame #1064

(e) ITWVTSP/R-iso - Frame #1069

(f) ITWVTSP/R-iso - Frame #1079

(g) Weights applied to each frame

Figure 3.15: Example of pedestrian tracking using ITWVTSP/R-iso ($\varepsilon = 0.12$ and ${}^{s}\omega_{\max} = 3.2$) on the sequence S1-T1-C Camera 3 of the PETS2006 Dataset.

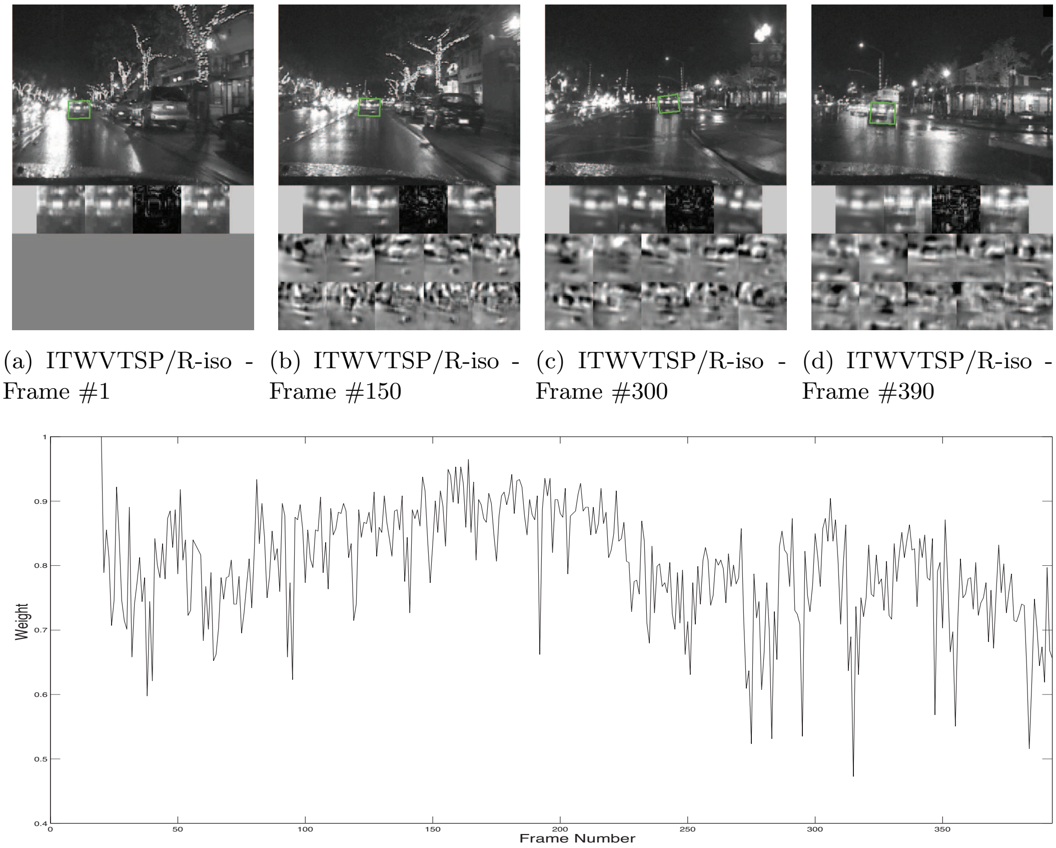


(a) ITWVTSP/R-iso - Frame #1 (b) ITWVTSP/R-iso - Frame #100 (c) ITWVTSP/R-iso - Frame #200

(d) ITWVTSP/R-iso - Frame #300 (e) ITWVTSP/R-iso - Frame #500 (f) ITWVTSP/R-iso - Frame #650

(g) Weights applied to each frame

Figure 3.16: Example of vehicle tracking ITWVTSP/R-iso ($\varepsilon = 0.07$ and ${}^{s}\omega_{\max} = 2.0$).



(a) ITWVTSP/R-iso - Frame #1

(b) ITWVTSP/R-iso - Frame #150

(c) ITWVTSP/R-iso - Frame #300

(d) ITWVTSP/R-iso - Frame #390

(e) Weights applied to each frame

Figure 3.17: Example of night vehicle tracking ITWVTSP/R-iso ($\varepsilon = 0.12$ and ${}^{s}\omega_{\max} = 2.0$).

on a particle filtering approach, producing the Incremental Temporally Weighted Visual Tracking with Spatial Penalty (ITWVTSP) algorithm and its reduced version (spatial weights fixed to one) the Incremental Temporally Weighted Visual Tracking (ITWVT) algorithm. These VT algorithms track targets while building on-line a robust appearance model of the object of interest. In addition, the ITWVTSP algorithm has the capacity of increasing accuracy in important regions of the object of interest.

Several alternatives for the computation of temporal weights and spatial penalty have been introduced, producing a family of VT algorithms. All the alternatives have been tested on challenging video sequences, showing their good performance compared to state-of-the-art techniques, and their polyvalence with respect to the scenario of application. Indeed, the algorithms have been applied to face tracking, pedestrian tracking, vehicle tracking and on the tracking of a rigid and static object (the poster).

The algorithms introduced in this Chapter are based on two weighting strategies, the temporal weighting of samples and the spatial penalty of hypothesis. With respect to the temporal weighting, a more in-deep interaction between the particle filter and the weighting strategy arises as an interesting future line of research to be explored. Indeed, the weights of the particles seems to be a good source of information for modulating the contribution of samples to the PCA. In Chapter 4 we will explore this in the context of total occlusion detection, although their use for defining the tracked sample confidence needs to be investigated.

With respect to spatial weights, in the Rockstar sequence we have seen that changes in the appearance of the tracked object in spatially important regions can decrease the performance of ITWVTSP compared to ITWVT. This suggest the study of dynamical spatial penalty strategies. Indeed, reconstruction error gives valuable spatial information about changes of the object appearance. This information could be used for adapting dynamically the values of the spatial weights.

In the next chapter, we study a solution to handle total occlusions of the tracked object by using a behavioural model of motion. This solution uses the algorithms developed here for avoiding model drift, and some measures of the PF for the total occlusion detection.

Chapter 4

Behavioural Information for Total Occlusion Handling in Visual Tracking: Application to Pedestrian Tracking

Contents

4.1 Introduction

A total occlusion of the object of interest is a hard problem in tracking that is often not handled by tracking algorithms. In the case of partial occlusions, the capacity of continuing tracking is usually left to the robustness of the tracking method, without performing any special action. In Chapter 3, we have developed a tracking algorithm that considers the quality of tracked samples for updating the object representation model. This strategy allows to decrease the importance of samples affected by, for instance, partial occlusions, which reduces the risk of model drift. Nevertheless, this strategy is not appropriate for total occlusions. Indeed, during a total occlusion, the visual information of the tracked object completely disappears. In such a case, the strategy described in Chapter 3 could stop the model update, but the output of the tracker would be unpredictable and nothing would guarantee a track recovery after the occlusion. Furthermore, temporal weights, as computed in Chapter 3, are not a good indicator of a total occlusion. A sudden change in the illumination, for instance, can decrease temporal weights but tracking using the visual model is still valid and no occlusion is present.

When designing a tracking algorithm capable of handling total occlusions, two strategies arise: make assumptions about the hidden movement or employ detection in the tracking. On the one hand, assumptions about the hidden motion usually consider a constant velocity of the target [Munder 2008, Pal 2008, Hu 2009, Ess 2009], although other strategies are also adopted in the literature. In [Papadourakis 2010], the principle of object permanence [Baillargeon 1985] is applied, and objects that disappear because of an occlusion are expected to appear again near the occluder. This kind of hypothesis is restrictive and does not take into account additional aspects that could influence the behaviour of the tracked object during the occlusion. On the other hand, tracking algorithms sometimes also perform detection. In this case, when the tracked object disappears due to an occlusion, the tracking stops and remains in detection mode until the object is detected again [Kalal 2010], or recovers hidden positions once the object is re-detected [Kelly 2009]. The main drawback of this strategy is clutter, which can generate false detections and consequently bad recovered trackings after occlusions.

When detection is used in tracking, the tracker interacts with an object detector and therefore occlusions are detected by the object detector itself. However, when assumptions about the hidden movement are done, a method for detecting occlusions is needed in order to decide when the tracker has to switch to "occluded mode". In [Pal 2008], Bhattacharyya distance is used to detect occlusions in face tracking. Indeed, the Bhattacharyya distance between the colour distribution of the tracked region and a Gaussian mixture model of colour components of old tracked region is computed. If this distance is bigger than 2.5 times the standard deviation of at least half of the components of the model, then the sample is labeled as occluded and the colour model is not updated. In [Hu 2009], a constant velocity model is supposed in case of occlusion in pedestrian tracking. Occlusions are considered only between tracked pedestrians and treated as an additional dimension of the state space that influences shape and appearance of pedestrians. The estimated occlusion state allows to know which pedestrian is occluding which one. This makes the observation likelihood function more robust, since only appearance information of observable pedestrians is employed. In [Yu 2010], the authors developed a VT algorithm where occlusion is taken into account when updating the object appearance model. This model is computed using a robust incremental PCA algorithm that computes principal components using only a subset of pixels (supposed not occluded) in each template. For recovering from an occlusion, the number of particles and the variance of the dynamical model are increased, which can be a problem in the presence of clutter.

Among tracking applications, pedestrian tracking is one of the most important. The spectrum of potential applications is wide, from video-surveillance [Leykin 2006] to ambient intelligence [Piotto 2009], passing through safety [Pai 2004] or urban planning [Kelly 2008]. Pedestrian tracking algorithms exploit the specificities of the tracked object but are faced to hard intrinsic and extrinsic difficulties. The main intrinsic difficulties are the highly articulated nature of the tracked object and the huge intra-class variability. The first, besides generating a wide set of feasible shapes, constantly generates partial occlusions. Indeed, body parts as arms and legs generate occlusions with other parts of the body and with nearby pedestrians. For this reason, human kinematics, from a point of view of an articulated object, is commonly considered in human tracking systems [Sundaresan 2009, del Rincón 2011]. The intra-class variability makes the design of a general appearance model for pedestrians a difficult task. Indeed, clothing can make the appearance of different pedestrians differ considerably.

Extrinsic difficulties (not directly related to pedestrians but to their context) are also important in pedestrian tracking. Indeed, common scenarios for pedestrian tracking are outside emplacements and big closed spaces. Both scenarios present illumination problems: outside emplacements because of big changes in illumination due to clouds or shadows, and big closed spaces because of a bad lighting. Bad lighting and illumination changes cause a reduction of the contrast between pedestrians and background, which makes VT harder. In addition, the clutter phenomenon is specially relevant in pedestrian tracking. Not only because of similar shapes present in the scene, but also because pedestrians are usually not alone. Indeed, the common situation is that more than one pedestrian appears at the same time in the scene. This results in a marked multi-modality of the probability distribution of the presence of pedestrians over the image, which is the main reason of popularity of PF approaches for localisation in pedestrian tracking. Actually, although techniques such as optimisation methods on graphs [Berclaz 2010, Alahi 2011], Kalman Filter [Alonso 2007] or Mean Shift [Wu 2006a] are used for pedestrian tracking, PF methods are dominant [Isard 2001, Wu 2006b, Munder 2008, Hu 2009].

In addition to all these problems related to pedestrians, the high mobility of them is another aspect that makes tracking difficult. This high mobility generates easily total occlusions in unconstrained scenarios, between pedestrians and objects or among pedestrians themselves. For this reason, the capacity of dealing with total occlusions is of special interest in pedestrian tracking applications. As commented above, some works that suppose a constant direction and speed can be found in the literature [Ess 2009, Breitenstein 2009]. However, pedestrians are objects with a particular behaviour that is rarely exploited by systems using real behavioural data. In [Venegas 2005], a reduced version of the pedestrian walking behavioural model described in Section 4.2 was used for pedestrian tracking. The tracking was performed in a Bayesian framework with the behavioural model as prior. This has two potential problems. On the one hand, if the model gives narrow candidate regions for position, the track can be lost if the behaviour of the pedestrian differs slightly from the model. On the other hand, if the model gives too wide regions, clutter can generate track errors. The tracking algorithm presented later tries to avoid these two situations by combining a behavioural and a visual model into a single VT framework.

The framework that we propose combines the two strategies described above for handling total occlusions: hidden motion assumptions and object detection. On one side, when a total occlusion occurs, the tracking stops using visual information and starts using a behavioural model of motion as likelihood function for assigning the weights of the particles. On the other side, while behavioural tracking is performed, the update of the visual model is stopped and the tracking switches to a sort of detection mode using this visual model. This detection mode allows to re-detect the tracked object after the occlusion, and then switch again to visual tracking. The occlusion detection and the pedestrian re-detection are performed by using statistics extracted from the particle filter used for tracking. This does not alter the computational complexity of the tracking algorithm, which is not the case in general when an object detector is used.

We apply this framework to pedestrian tracking using the Discrete Choice Pedestrian Model (DCPM), introduced in [Robin 2009], as behavioural model of motion. The information provided by this behavioural model depends on personal conditions of the tracked pedestrian, but also on conditions of surrounding pedestrians. This allows to capture complex behaviours, reproducing better than a simple constant velocity model the motion pattern of a pedestrian (see validation of the model in [Robin 2009]). Although we use the proposed framework for pedestrian tracking, it can be easily adapted to any other problem with an inherent behaviour of motion.

This Chapter is organised as follows. In Section 4.2 we make a short introduction to pedestrian behaviour modelling and describe the DCPM, which is the behavioural model used afterwards for tracking pedestrians. Then, in Section 4.3 the Model cOrruption and Total Occlusion Handling (MOTOH) framework is introduced, describing each one of its modules in detail. Afterwards, the details of the integration of the pedestrian behaviour model into the MOTOH framework are discussed in Section 4.3.2. Then, in Section 4.4 several tests are performed and finally, in Section 4.5, some conclusions and potential lines of future research are given. A journal paper on the works presented in this Chapter is being prepared and will be submitted soon.

4.2 Pedestrian Behaviour Modelling: the Discrete Choice Pedestrian Model

Pedestrian behaviour modelling is an important topic in a wide variety of fields, such as architecture, transportation, sociology, economics or computer vision. One of the main problems in pedestrian behaviour modelling is the difficulty of data acquisition, and only a few of the models presented in the literature have been calibrated on real datasets.

In general, pedestrian behaviour modelling methods can be classified into two main families: macroscopic models and microscopic models. On the one hand, macroscopic models treat groups of individuals as an entity and reproduce its evolution under certain conditions [Helbing 2007, Piccoli 2011]. On the other hand, microscopic models treat pedestrians individually. These techniques are receiving great attention nowadays and in our context, pedestrian tracking, these are the most appropriate. Examples of microscopic pedestrian models are the social force model [Helbing 1995], the multi-layer utility maximisation model [Hoogendoorn 2002] or the Discrete Choice Pedestrian Model (DCPM) [Robin 2009]. For a general literature review on pedestrian modelling, we refer the interested reader to [Bierlaire 2009].

We will concentrate on the DCPM introduced in [Robin 2009]. A reduced version of this model has been used in CV applications [Antonini 2004, Venegas 2005, Antonini 2005, Antonini 2006]. This model concentrates on the operational level of the hierarchical pedestrian decision scheme introduced by [Daamen 2004], where the author divide the decisions taken by a pedestrian in three levels:

- **Strategical:** in charge of the choice of destination and activity;
- **Tactical:** in charge of the planning of the order of activities, activity area choice and route choice;
- **Operational:** in charge of instantaneous decisions, such as steps and stops.

The DCPM considers that the strategic and tactical decisions have been exogenously made, i.e. the model considers that origin and destination are given, and models the short range behaviour in normal conditions (non-evacuation and non-panic situations).

The behavioural structure of the model is illustrated in Figure 4.1. There, it can be observed that the DCPM models two types of decisions: unconstrained decisions and constrained decisions. Unconstrained decisions are independent of the presence of other pedestrians, while constrained decisions depend on the presence and behaviour of nearby pedestrians. This means that the model captures individual behaviour as well as interactions among pedestrians.

The DCPM is a Discrete Choice Model (DCM). DCMs are behavioural models designed to forecast the behaviour of decision makers in a choice context, when a finite set of alternatives is available. They are based on four elements:

- a set of alternatives C_n available for decision maker n;

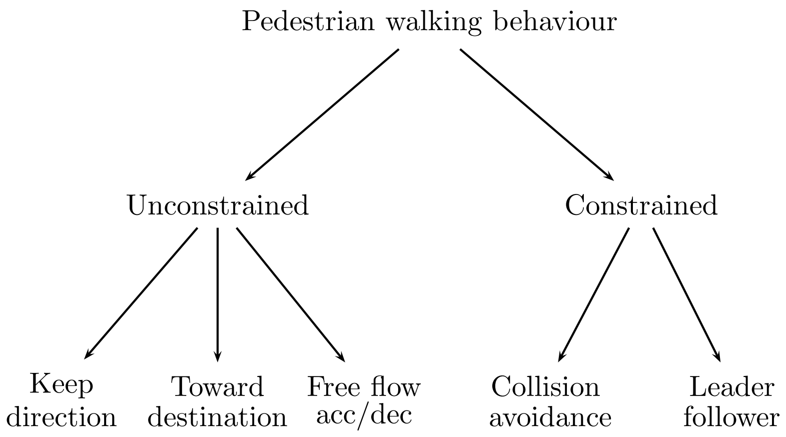


Figure 4.1: Conceptual framework for the DCPM from [Robin 2009].

- for each $i \in C_n$, a set of attributes X_{in} describing the alternative i;
- a set of socio-economic characteristics Z_n describing the decision maker;
- assumptions about the choice process and the modelling of uncertainty.

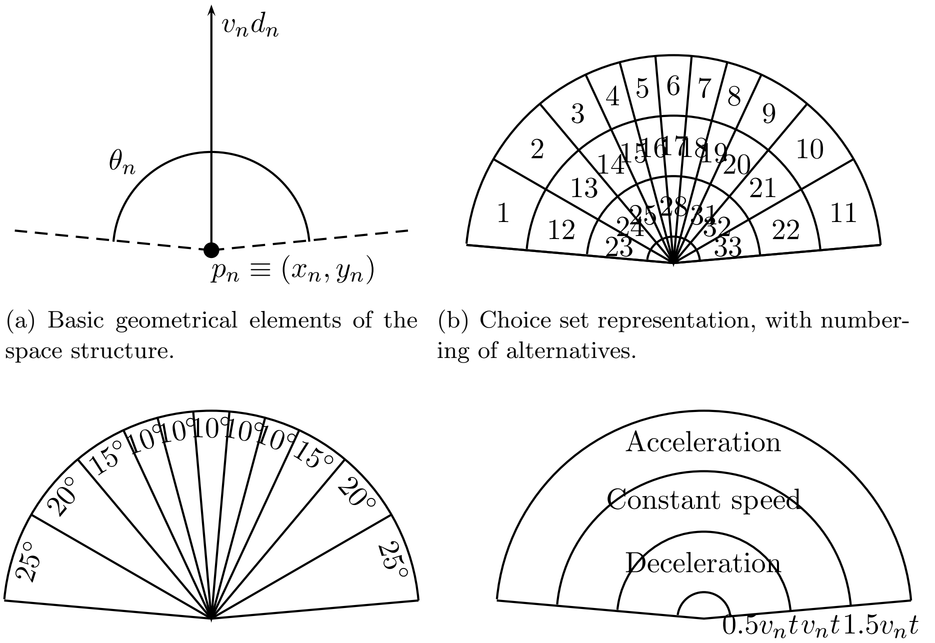
DCMs assume a rational decision maker, who performs choices by maximising the utilities perceived from the available alternatives [Ben-Akiva 1985]. The utility function perceived by an individual n for an alternative $i \in C_n$ is modelled as a random variable U_{in}:

$$U_{in} = V_{in} + \varepsilon_{in}, \tag{4.1}$$

where V_{in} denotes the deterministic term that captures the systematic behaviour, and ε_{in} is a random term that captures the uncertainty.

In the DCPM, the choice set consists of 33 cell regions situated in a circular sector of 170° in front of the pedestrian, whose centre is defined by the position of the pedestrian and the radius by 1.75 times its speed. Each alternative corresponds to a combination of speed regime v (deceleration, constant speed or acceleration) and a radial direction d (±72.5°, ±50°, ±32.5°, ±20°, ±10° or 0°). In Figure 4.2 the choice set and the elements that define it are shown.

The systematic utility perceived by individual n for the alternative identified by the



(a) Basic geometrical elements of the space structure.

(b) Choice set representation, with numbering of alternatives.

(c) Spatial discretisation of directions.

(d) Spatial discretisation of speed regimes.

Figure 4.2: Elements that define the choice set of alternatives for the next step in [Robin 2009].

speed regime v and direction d is

$$
\begin{array}{rll}
V_{vdn} = \beta_{\text{dir\_central}} \text{dir}_{dn} I_{\text{d,central}} & + & \multirow{3}{*}{} \\
\beta_{\text{dir\_side}} \text{dir}_{dn} I_{\text{d,side}} & + & \left.\right\} \textit{keep direction} \\
\beta_{\text{dir\_extreme}} \text{dir}_{dn} I_{\text{d,extreme}} & + & \\
\beta_{\text{ddist}} \text{ddist}_{vdn} & + & \left.\right\} \textit{toward destination} \\
\beta_{\text{ddir}} \text{ddir}_{dn} & + & \\
\beta_{\text{dec}} I_{\text{v,dec}} (v_n / v_{\max})^{\lambda_{\text{dec}}} & + & \\
\beta_{\text{accLS}} I_{\text{n,LS}} I_{\text{v,acc}} (v_n / v_{\text{maxLS}})^{\lambda_{\text{accLS}}} & + & \left.\right\} \textit{free flow acceleration} \\
\beta_{\text{accHS}} I_{\text{n,HS}} I_{\text{v,acc}} (v_n / v_{\max})^{\lambda_{\text{accHS}}} & + & \\
I_{\text{v,acc}} I^L_{\text{d,acc}} \alpha^L_{\text{acc}} D_L^{\rho^L_{\text{acc}}} \Delta v_L^{\gamma^L_{\text{acc}}} \Delta \theta_L^{\delta^L_{\text{acc}}} & + & \left.\right\} \textit{leader-follower} \\
I_{\text{v,dec}} I^L_{\text{d,dec}} \alpha^L_{\text{dec}} D_L^{\rho^L_{\text{dec}}} \Delta v_L^{\gamma^L_{\text{dec}}} \Delta \theta_L^{\delta^L_{\text{dec}}} & + & \\
I_{d,C} \alpha_C e^{\rho_C D_C} \Delta v_C^{\gamma_C} \Delta \theta_C^{\delta_C} & & \left.\right\} \textit{collision avoidance}
\end{array}
\tag{4.2}
$$

where all the α, β, γ, δ, λ and ρ parameters are estimated. This expression has five sub-blocks: *keep direction*, *toward destination*, *free flow acceleration*, *leader-follower* and *collision avoidance*. Each one of these sub-blocks models a different aspect of the walking behaviour of pedestrians.

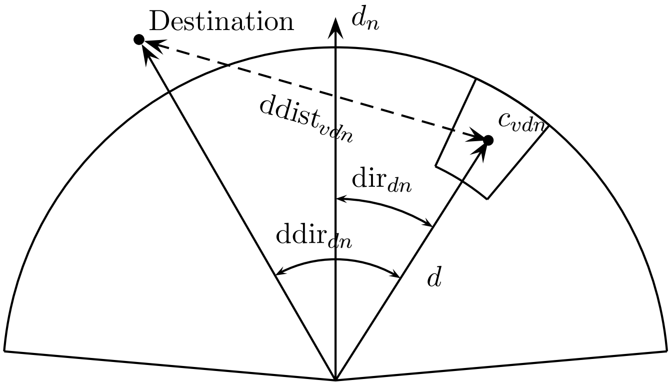


Figure 4.3: Elements capturing the *keep direction* and *toward destination* behaviours

Keep direction captures the tendency of people to avoid frequent variation of direction and plays a smoothing role in the model, avoiding drastic changes of direction. The non-linearity of this pattern is captured by defining three different groups of directions: central (cones containing cells 5, 6 and 7 in Figure 4.2(b)), side (cones containing cells 3, 4, 8 and 9 in Figure 4.2(b)) and extreme (cones containing cells 1, 2, 10 and 11 in Figure 4.2(b)). Each group has its own term in the utility function

$$\beta_{\mathrm{dir\_central}}\mathrm{dir}_{dn}I_{\mathrm{d,central}} + \beta_{\mathrm{dir\_side}}\mathrm{dir}_{dn}I_{\mathrm{d,side}} + \beta_{\mathrm{dir\_extreme}}\mathrm{dir}_{dn}I_{\mathrm{d,extreme}} \qquad (4.3)$$

where the variable dir_{dn} is defined as the angle in degrees between the direction d and the direction d_n corresponding to the current direction, as shown in Figure 4.3. Note that the indicators $I_{\mathrm{d,central}}$, $I_{\mathrm{d,side}}$ and $I_{\mathrm{d,extreme}}$, that are equal to one if the alternative corresponds to a cell in their cones and zero otherwise, guarantee that only one of the three terms is nonzero for any given alternative.

Toward destination models the tendency of individuals to choose, for the next step, a spatial location that minimises angular displacement and distance to destination. It is captured by

$$\beta_{\mathrm{ddist}}\mathrm{ddist}_{vdn} + \beta_{\mathrm{ddir}}\mathrm{ddir}_{dn} \qquad (4.4)$$

where the variable ddist_{vdn} is defined as the distance (in meters) between the destination and the center of the alternative C_{vdn}, while ddir_{dn} is defined as the angle in degrees between the destination and the alternative's direction d, as shown in Figure 4.3.

Free flow acceleration models the acceleration of individuals in free flow conditions. It is captured by three terms, depending on the speed regime of the alternative (see

Figure 4.2(d)):

$$\beta_{\text{dec}} I_{\text{v,dec}} (v_n/v_{\max})^{\lambda_{\text{dec}}} + \beta_{\text{accLS}} I_{\text{n,LS}} I_{\text{v,acc}} (v_n/v_{\text{maxLS}})^{\lambda_{\text{accLS}}} + \beta_{\text{accHS}} I_{\text{n,HS}} I_{\text{v,acc}} (v_n/v_{\max})^{\lambda_{\text{accHS}}}, \tag{4.5}$$

where $I_{\text{v,dec}}$ is one if v corresponds to a deceleration, zero otherwise; $I_{\text{v,acc}}$ is one if v corresponds to an acceleration, zero otherwise; $I_{\text{n,LS}}$ is one if the individual's current speed is less than or equal to 1.39, zero otherwise; $I_{\text{n,HS}} = 1 - I_{\text{n,LS}}$; the reference speed for low speeds is $v_{\text{maxLS}} = 1.39$; and the reference speed is selected to be the maximum speed observed, $v_{\max} = 4.84$ (m/s).

The *leader-follower* block captures the influence on an individual of other pedestrians walking in front of him [Li 2001]. A possible leader can be identified among a set of potential leaders for each one of the 11 radial cones of Figure 4.2(c). A potential leader is an individual who is inside a certain region of interest, not far from the decision maker and with a moving direction close enough to the direction of its corresponding radial cone. An individual k is defined as a potential leader based on the following indicator function (see Figure 4.4 for graphical details):

$$I_L^k = \begin{cases} 1, & \text{if } d_l \leq d_k \leq d_r \ \ \text{(is in the cone)}, \\ & \text{and } 0 < D_k \leq D_{th} \ \ \text{(not too far)}, \\ & \text{and } 0 < |\Delta\theta_k| \leq \Delta\theta_{th} \ \ \text{(walking in almost the same direction)}, \\ 0, & \text{otherwise}, \end{cases}$$

where d_l and d_r represent the bounding left and right directions of the cone in the choice set (defining the region of interest) while d_k is the direction identifying the position of pedestrian k. D_k is the distance between pedestrian k and the decision maker, $\Delta\theta_k = \theta_k - \theta_d$ is the difference between the movement direction of pedestrian k (θ_k) and the angle characterising direction d, i.e. the direction identifying the radial cone where individual k lies (θ_d). The two thresholds D_{th} and $\Delta\theta_{th}$ are fixed at the values $D_{th} = 5D_{max}$, where D_{max} is the radius of the choice set, and $\Delta\theta_{th} = 10$ degrees. This seems to be reasonable and well adapted to pedestrian environment perception.

Among the set of potential leaders for each radial direction, the one at the minimum distance $D_L = \min_{k \in K}(D_k)$ is chosen as Leader and induces an attractive interaction on the decision maker, captured by

$$I_{\text{v,acc}} I^L_{\text{d,acc}} \alpha^L_{\text{acc}} D_L^{\rho^L_{\text{acc}}} \Delta v_L^{\gamma^L_{\text{acc}}} \Delta\theta_L^{\delta^L_{\text{acc}}} + I_{\text{v,dec}} I^L_{\text{d,dec}} \alpha^L_{\text{dec}} D_L^{\rho^L_{\text{dec}}} \Delta v_L^{\gamma^L_{\text{dec}}} \Delta\theta_L^{\delta^L_{\text{dec}}}, \tag{4.6}$$

where $I^L_{\text{d,acc}}$ is one if the leader in the cone d has been identified with a speed larger than v_n, zero otherwise. Similarly, $I^L_{\text{d,dec}} = 1 - I^L_{\text{d,acc}}$ is one if the leader in cone d has been identified with a speed lower than v_n, zero otherwise. The indicator functions $I_{v,acc}$ and $I_{v,dec}$ discriminate between accelerated and decelerated alternatives, as with the free flow acceleration model. Finally, $\Delta v_L = |v_L - v_n|$, where v_L and v_n are the leader's speed module and the decision maker's speed

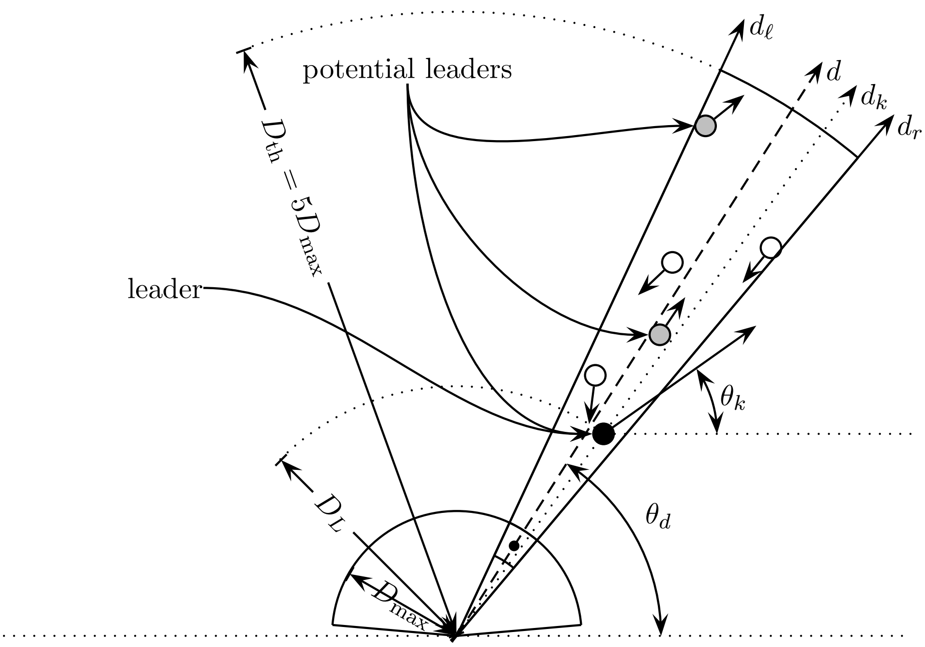


Figure 4.4: Leader and potential leaders in a given cone.

module, respectively; and $\Delta\theta_L = \theta_L - \theta_d$, where θ_L represents the leader's movement direction and θ_d is the angle characterising direction d, as shown in Figure 4.4. Note that in the final specification, the parameter δ^L_{dec} appeared not to be significantly different from 0 and therefore was removed from the specification.

Finally, *collision avoidance* models the influence of potential collision on the trajectory of the decision maker [Collett 1981]. Similarly to *leader-follower*, for each direction in the choice set, a collider is identified among a set of potential colliders not far from the decision maker and walking in the opposite direction, i.e. an individual k is defined as a potential collider based on the following indicator function:

$$I_C^k = \begin{cases} 1, & \text{if } d_l \leq d_k \leq d_r \ \text{ (is in the cone),} \\ & \text{and } 0 < D_k \leq D'_{th} \ \text{ (not too far),} \\ & \text{and } \frac{\pi}{2} \leq |\Delta\theta_k| \leq \pi \ \text{ (walking in the other direction),} \\ 0, & \text{otherwise,} \end{cases}$$

where d_l, d_r and d_k are the same as those defined for the *leader-follower* model; D'_k is the distance between individual k and the center of the alternative; and $\Delta\theta_k = \theta_k - \theta_{d_n}$ is the difference between the movement direction of pedestrian k, θ_k, and the movement direction of the decision maker, θ_{d_n}. The value of the distance threshold is now fixed to $D'_{th} = 10D_{max}$, which is larger than the value

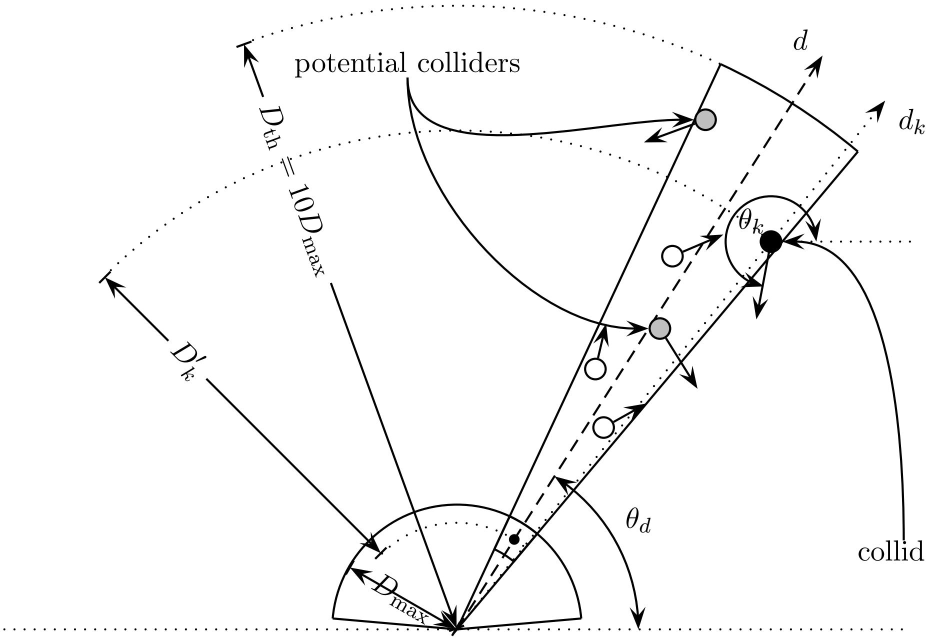


Figure 4.5: Collider and potential colliders in a given cone

used for *leader-follower* assuming that the collision avoidance behaviour has a longer range of action, happening also at a lower level of density. In Figure 4.5, graphical details are shown.

Among the set of K_d potential colliders for direction d, a collider is chosen in each cone as that individual having $\Delta\theta_C = \max_{k\in K_d} |\Delta\theta_k|$. The indicator $I_{d,C} = 1$ if a collider has been identified, and 0 otherwise. Finally, the collision avoidance term is included in the utility functions of all the alternatives as

$$I_{d,C}\alpha_C e^{\rho_C D_C} \Delta v_C^{\gamma_C} \Delta\theta_C^{\delta_C}. \tag{4.7}$$

Note that in the final specification, the parameters γ_C and δ_C were not significantly different from 0. Therefore the collision avoidance term depends only on the presence or not of a collider and on the cone where this collider is placed. Since the estimated value of $\alpha_C < 0$, this expression reduces the utility, and therefore the probability, of the alternatives in cones containing colliders.

For a detailed description of the pedestrian behaviour motivations of using these expressions for the utilities, we refer the reader to [Robin 2009].

The random term in Equation (4.1) is considered with a correlation structure depending on the speed and direction, producing a Cross Nested Logit (CNL) model

[Bierlaire 2006]. Five nests were identified: accelerated, constant speed, decelerated, central and not central. The probability of choosing alternative i within the choice set C is

$$P(i|C) = \sum_{m=1}^{M} \frac{(\sum_{j\in C} \alpha_{jm}^{\mu_m/\mu} e^{\mu_m V_j})^{\frac{\mu}{\mu_m}}}{\sum_{n=1}^{M}(\sum_{j\in C} \alpha_{jn}^{\mu_n/\mu} e^{\mu_n V_j})^{\frac{\mu}{\mu_n}}} \frac{\alpha_{im}^{\mu_m/\mu} e^{\mu_m V_i}}{\sum_{j\in C} \alpha_{jm}^{\mu_m/\mu} e^{\mu_m V_j}}, \tag{4.8}$$

where M is the number of nests, α_{jm} is the degree of membership of alternative j to nest m and $0 < \mu \leq \mu_m$ $\forall m$, being μ the scale parameter and μ_m the nest parameter. In the DCPM, $M = 5$, $\alpha_{jm} = 0.5$ $\forall j, m$ and $\mu = 1$, all the other parameters are estimated using behavioural data.

Model parameters were estimated using real pedestrian trajectories, manually tracked from video sequences collected in Sendai, Japan [Teknomo 2000, Teknomo 2002]. The dataset consists of 190 pedestrian trajectories with 2 observations per second, which produce a total of 10200 observation (9281 after data cleaning). The values of parameters can be found in Table 4.1.

The model was also validated on the same dataset (using a cross-validation procedure) and on a dataset collected at Delft University [Daamen 2003b, Daamen 2003a, Daamen 2004]. The validation allowed to show the robustness and quality of the specification, as well as its superiority with respect to a constant speed model.

4.3 The MOTOH Visual Tracker Framework

In the previous Chapter, an approach for avoiding model drift has been introduced. This approach is useful in situations where the appearance of the tracked object changes momentarily due to illumination changes, partial occlusions or temporal deformations, for instance. However, if a total occlusion of the tracked object occurs, this approach cannot handle the situation, since it requires the tracked object to be at least partially visible. For dealing with this type of situations, here we introduce the Model cOrruption and Total Occlusion Handling (MOTOH) visual tracking framework, which combines the spatial and temporal weighting strategies introduced in Chapter 3, a total occlusion detector, based on the PF used for tracking, and a behavioural model of motion for continuing the tracking when the object is occluded.

The common approach, when a behavioural motion model is available, is to use it as a prior for state evolution. This presents two potential problems. On the one hand, if the behavioural model over-fits the behaviour of the tracked object, a slightly abnormal behaviour of this object can potentially cause a loss of track. On the other hand, if the model under-fits the behaviour, propagated particles can

Variable name	**Coefficient estimate**	***t*-test against 0**	**Variable name**	**Coefficient estimate**	***t*-test against 0**	***t*-test against 1**
β_{ddir}	-0.0793	-24.14	ρ^L_{acc}	-0.465	-1.78	
β_{ddist}	-1.52	-11.63	γ^L_{acc}	0.552	1.98	
$\beta_{\mathrm{dir\_extreme}}$	-0.0343	-9.71	α^L_{dec}	3.78	5.41	
$\beta_{\mathrm{dir\_side}}$	-0.0553	-22.71	ρ^L_{dec}	-0.654	-6.70	
$\beta_{\mathrm{dir\_central}}$	-0.0320	-13.90	γ^L_{dec}	0.658	5.48	
β_{accLS}	-4.94	-25.20	δ^L_{acc}	-0.179	-2.22	
β_{accHS}	-7.41	-5.10	α_C	-0.00730	-10.84	
β_{dec}	-0.0645	-2.46	ρ_C	-0.212	-8.38	
λ_{accLS}	4.37	20.06	μ_{acc}	1.66	9.97	3.95
λ_{accHS}	0.354	2.02	μ_{const}	1.45	16.99	5.25
λ_{dec}	-2.40	-8.50	$\mu_{central}$	5.76	2.84	2.34
α^L_{acc}	0.735	1.87	$\mu_{not\_central}$	1.82	13.12	5.91
Sample size = 9281			Init log-likelihood = -32451			
Nbr of estimated parameters = 24			Final log-likelihood = -13944.74			
$\bar{\rho}^2 = 0.570$			Likelihood ratio test = 37013			

Table 4.1: **CNL** estimation results extracted from [Robin 2009]

be spread over a wide region of the state space, which can generate mistakes in cluttered situations. Furthermore, under the hypothesis of smoothness in position, shape and appearance change, a Gaussian prior is usually well adapted and the added value of using the behavioural model would be low. For these reasons, we propose here to use the behavioural model directly in the likelihood $p(z_k|x_k)$, but only when the visual information on the scene does not allow to continue the tracking of the object of interest, i.e. when a total occlusion happens. This switch between visual and behavioural tracking requires a total occlusion detector, which is performed by measuring some parameters of the PF in charge of the tracking. The objective of performing occlusion detection by measuring some parameters of the PF is to keep the computational complexity of the algorithm as similar to ITWVTSP as possible.

In Figure 4.6, a schematic view of the MOTOH framework is shown. The input video is processed by the feature extraction block. As in Chapter 3, we use grayscale level pixels as features, although the use of another kind of features could be easily implemented. The object representation feeds the particle filter and is computed by the ITWPCA algorithm applied to rectangular templates. The particle filter considers spatial weights and therefore, up to here, the diagram represents the ITWVTSP algorithm. Actually, occlusions are detected by detecting changes in the particles, as introduced later, and depending on this, the total occlusion detector block chooses between the visual likelihood and the behavioural likelihood. The occlusion detector gives as output the mean state according to particle weights, which is used by the object representation for updating the visual model and by the behavioural model as source of historical behavioural information.

In the following Section, the proposed total occlusion detector is introduced and in Section 4.3.2, the complete algorithm for using the MOTOH framework in pedestrian tracking with the DCPM as behavioural model is described in detail.

4.3.1 Detecting Total Occlusions

The design of a total occlusion detector is not straightforward. Indeed, nothing about the occluder object is known a priori, except that it causes the loss of the visual information of the tracked object. Actually, when a total occlusion happens, the visual model is measuring some sort of noise and this is the information that we use for detecting the occlusion.

When the tracked object is well visible and the tracking is performed correctly, the weights of the particles are high and the variance of the true weights low. Therefore, the normalisation factor and the estimated variance of the true weights are indicators, at a given frame k, of the quality of the observed information of the tracked object. However, scale information about these two measures

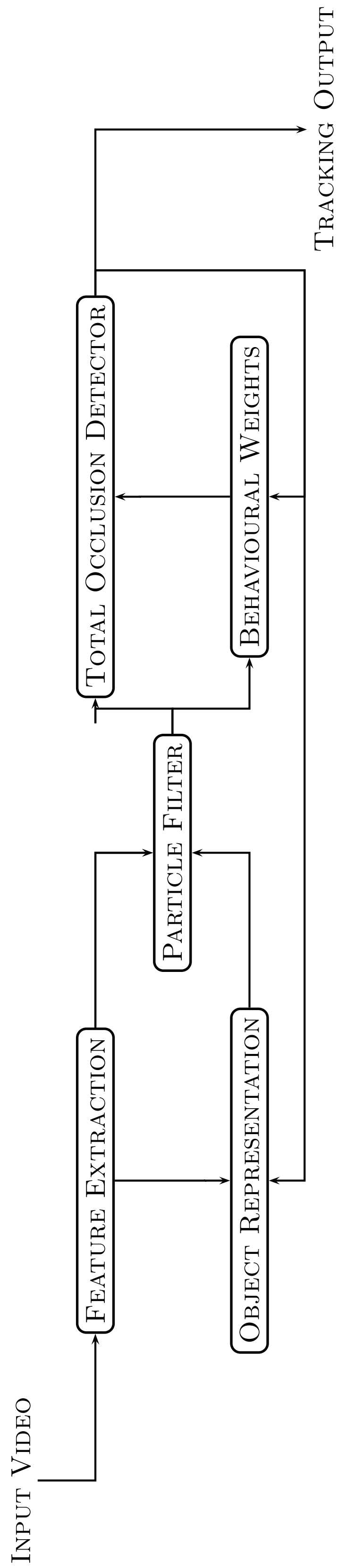


Figure 4.6: Schematic view of the MOTOH Framework

is not available, which makes hard thresholds too application-specific and even situation-specific. Indeed, even in the same application, the fluctuations of their values can be of several orders of magnitude without necessarily the presence of an occlusion, just a distancing of the object of interest from the visual model. This is the case, for instance, of pedestrian tracking, where the deformations of the walking pedestrian generate considerable changes on the values of these measures.

Let us precisely define these two measures. The normalisation factor $n_{w,k}$ at a given frame k is computed as

$$n_{w,k} = \sum_{i=1}^{N_s} \widetilde{w}_k^i, \tag{4.9}$$

where N_s is the number of particles and $\tilde{w}_k^i$ is the weight of particle i at time step k before normalisation, i.e. before transforming the set of particle weights into a discrete probability distribution function. The variance of the true weights at time step k, $v_{w,k}$, can be estimated by combining Equation (2.15) and Equation (2.16) as

$$v_{w,k} \simeq N_s \sum_{i=1}^{N_s} (w_k^i)^2 - 1, \tag{4.10}$$

where w_k^i is the normalised weight of particle i at time k.

Due to the fluctuations explained above, a dynamical procedure for deciding when the values of $n_{w,k}$ and $v_{w,k}$ correspond to an occlusion, is needed. Solutions based on the mean and the variance of the obtained values do not perform well, since a gradual variation of the values, due to a gradual occlusion for instance, can be undetected. The situation-specificity commented above makes techniques such as SVM, not well suited either.

Here we propose to keep an historical log of the obtained values and compare the new values with the median and the range of the log. Let us suppose that a set of L not occluded frames has been observed, obtaining $\{n_{w,\ell},\ \ell \in L\}$ and $\{v_{w,\ell},\ \ell \in L\}$. An indicator of occlusion at time step k, O_k, is computed as

$$O_k = \begin{cases} 1, & \text{if } \frac{\text{median}\{\frac{n_{w,\ell}}{v_{w,\ell}}, \ell \in L\}}{\frac{n_{w,k}}{v_{w,k}}} > \frac{\max\{\frac{n_{w,\ell}}{v_{w,\ell}}, \ell \in L\}}{\min\{\frac{n_{w,\ell}}{v_{w,\ell}}, \ell \in L\}} \\ 0, & \text{otherwise} \end{cases}, \tag{4.11}$$

where the value 1 indicates an occlusion at time k, and 0 no occlusion at time k. This occlusion indicator monitors the new values of the ratio between normalisation factor and variance of true weights. When the jump from the median value (which is a robust estimator of the central tendency) to the new obtained value is bigger than the maximum jump observed, the indicator considers that the new value is an outlier, i.e. an occlusion. This indicator has shown a good performance in the tests.

4.3.2 Pedestrian Tracking using the Discrete Choice Pedestrian Model in the MOTOH Framework

The MOTOH framework provides a generic environment for combining the tracking algorithms introduced in Chapter 3 with an occlusion handling algorithm. We apply this framework to pedestrian tracking, using the DCPM as behavioural model. In this Section, we describe the assumptions and conditions required by this behavioural model and we provide a detailed description of the global algorithm.

The DCPM (see Section 4.2 for details) needs position and speed of all the pedestrians in the scene for generating the choice set and computing utilities. Remember that this model captures interactions between pedestrians (leader-follower and collision avoidance terms of Equation (4.2)). Speed and position computations require the calibration of the camera, which limits the use of the presented algorithm to fixed cameras, in general. As the position of a pedestrian we consider the middle point of the base of the rectangular tracked region, which is supposed to be at floor, i.e $z = 0$. As the DCPM only depends on the dimensions of the state space related to the position, but not on scale, for instance, the original size s_o, size in the first tracked frame, is computed using the calibration information for every new tracked region. Then, if the tracking switches to behavioural tracking, the size of templates defined by particles is computed, and the difference with respect to the original size is used for penalising the weight of particles given by the DCPM. Given a particle i at time step k whose size in world coordinates is s^i_k, the weight given the DCPM is multiplied by a factor $f^i_{s,k}$ defined as

$$f^i_{s,k} = e^{-3|s_o - s^i_k|}. \tag{4.12}$$

This function was defined empirically, in order to keep the weight of particles with a variation of ±10cm in size almost unchanged ($e^{-0.3} \simeq 0.75$). The use of this penalising factor avoids an arbitrary change of size of the tracked region while using the behavioural tracking.

An important aspect when using a behavioural model is to respect the estimation conditions, i.e. for which situations it has been designed and under which conditions it has been calibrated. The DCPM has been designed and calibrated using data in normal conditions, i.e. non-evacuation and non-panic situations, and in general is valid only in this context. The data used for calibration was collected at 2 data points per second. The choice set is generated using points from half second before the current point and the prediction gives probabilities for positions one second after. These conditions have been respected when using the model in the MOTOH framework, which means that the algorithm needs to correctly track the first $1.5 \cdot \text{fps}$ frames of a pedestrian before being capable of using the pedestrian model. The frame rate of the videos is generally considerably higher than the frame rate of the data used in the model estimation (2 frames per second). In order to use this extra information and to avoid jumps in the choice set generation, the speed of pedestrians is computed using a linear regression on the trajectory points.

Finally, the destination is an exogenous variable in the DCPM and needs to be given for computing utilities. In the tests that we have done, a simple destination estimation is performed. Two potential destinations are predefined, $d_1 = (20, 0)$ and $d_2 = (-20, 0)$. If during the first $1.5 \cdot \text{fps}$ tracked frames, the x coordinate in world coordinates increases, d_1 is chosen as destination, if not d_2.

In Algorithm 6, the algorithm obtained by applying the MOTOH framework to pedestrian tracking, using the DCPM, is described in detail.

Algorithm 6 The MOTOH Pedestrian Tracker

```
At frame $k = 0$ compute pedestrian size in world coordinates, $s_o$, and store the value
for $k = 1, \ldots, 1.5 \cdot \text{fps}$ do
    Track pedestrian using the ITWVTSP algorithm (Algorithm 5)
    Store tracked position
    Compute $n_{w,k}$ and $v_{w,k}$ and store them
for $k > 1.5 \cdot \text{fps}$ do
    Draw particles according to the dynamical model (Equation (2.22)) and the weight distribution of particles.
    For each particle, compute its weight according to the observation model and spatial weights (Equation (3.10) and Equation (3.11)).
    Compute $n_{w,k}$ and $v_{w,k}$ and apply Equation (4.11) using the last $L$ stored values
    if $O_k = 1$ then
        For each particle, compute its weight according to the DCPM (Equation (4.8) multiplied by Equation (4.12)) using stored information about position and speed of all the pedestrians on the scene
        Compute mean particle and store its position
    else
        Compute mean particle and store its position, $n_{w,k}$ and $v_{w,k}$
```

4.4 Tests and Results

In Chapter 2 we have reviewed standard techniques for computing performance of visual tracking algorithms. These techniques have been employed in Chapter 3 for quantitatively showing the improvement obtained with the proposed algorithm. As also commented in Chapter 2, sometimes it is interesting to assess qualitatively the performance of an algorithm faced to a particular situation, especially if a ground truth cannot be easily defined or cannot be defined at all. The MOTOH Pedestrian Tracker, tested here, is under this situation. Indeed, the core tracking algorithm is the ITWVTSP algorithm, whose good performance has already been quantified in Chapter 3. Now we are interested in observing the capacity of the MOTOH Pedestrian Tracker to handle occluded situations, where a ground truth cannot be defined. We will apply the algorithm to several video sequences, without occlusions, with artificially generated occlusions and with real occlusions, and will observe its capacity to recover the track of the object after the occlusion. Obtained results are compared with those obtained using IVT and TLD, showing the higher robustness of the MOTOH Pedestrian Tracker, in non-occluded situations thanks to the ITWVTSP algorithm and in occluded situations thanks to the behavioural model.

The IVT and the MOTOH Pedestrian Tracker were used with the same parameters than in Chapter 3, in terms of number of particles (600 particles), maximum number of eigenvectors (16 eigenvectors), block size for updating the PCA (5 images), forgetting factor (0.97) and standard deviations of the dynamical model (Equation (2.22)): 9.0px for row and column displacements, 0.001 radians for rotation, 0.001 radians for skewness, 0.05 for scaling and 0.05 for aspect ratio. The size of eigenvectors was adapted to the shape of a pedestrian, using a template of 50×25 pixels. In the MOTOH Pedestrian Tracker, for avoiding instabilities due to high spatial weights or low temporal weights, conservative values were chosen for noise threshold $\varepsilon = 0.12$ and maximum spatial weight ${}^{s}\omega_{\max} = 2.0$ with an isotropic shape. Given that the upper part of a pedestrian is more stable, since the legs are constantly moving, the maximum spatial weights were not placed on the center of the template but on 1/3 of the height (see Figure 4.7). The size of the historical values of $n_{w,k}$ and $v_{w,k}$ was fixed to 15, which is a good compromise between the influence of recent values and values corresponding to samples already added to the model (during 15 samples, three updates of the PCA are performed).

In all the images, the big point in the middle of the basis of each tracked region represents the position considered for this pedestrian (only used in the MOTOH Pedestrian Tracker). The output of the MOTOH Pedestrian Tracker is coded in colours as follows:

- **Red:** The tracker is in the period where only ITWVTSP is used, first $1.5 \cdot \text{fps}$ frames.
- **Blue:** Enough positions have been collected and if an occlusion appears, the behavioural model could be used.
- **Green:** The occlusion detector has detected an occlusion and the tracking is being performed by the behavioural model. The black points represent the 33 alternatives of the DCPM (see Section 4.2).

The starting patch of every pedestrian has been introduced by hand and is the same for all the methods. A pedestrian is considered as going outside the image region if

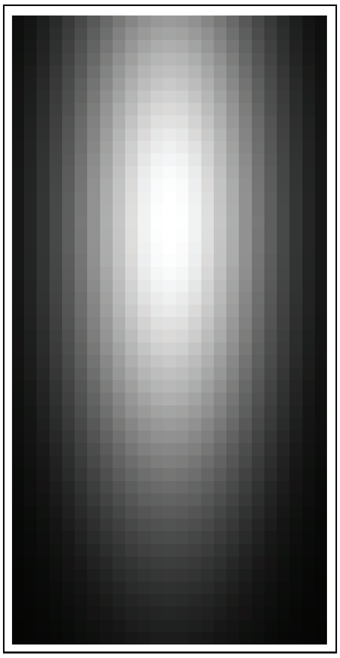


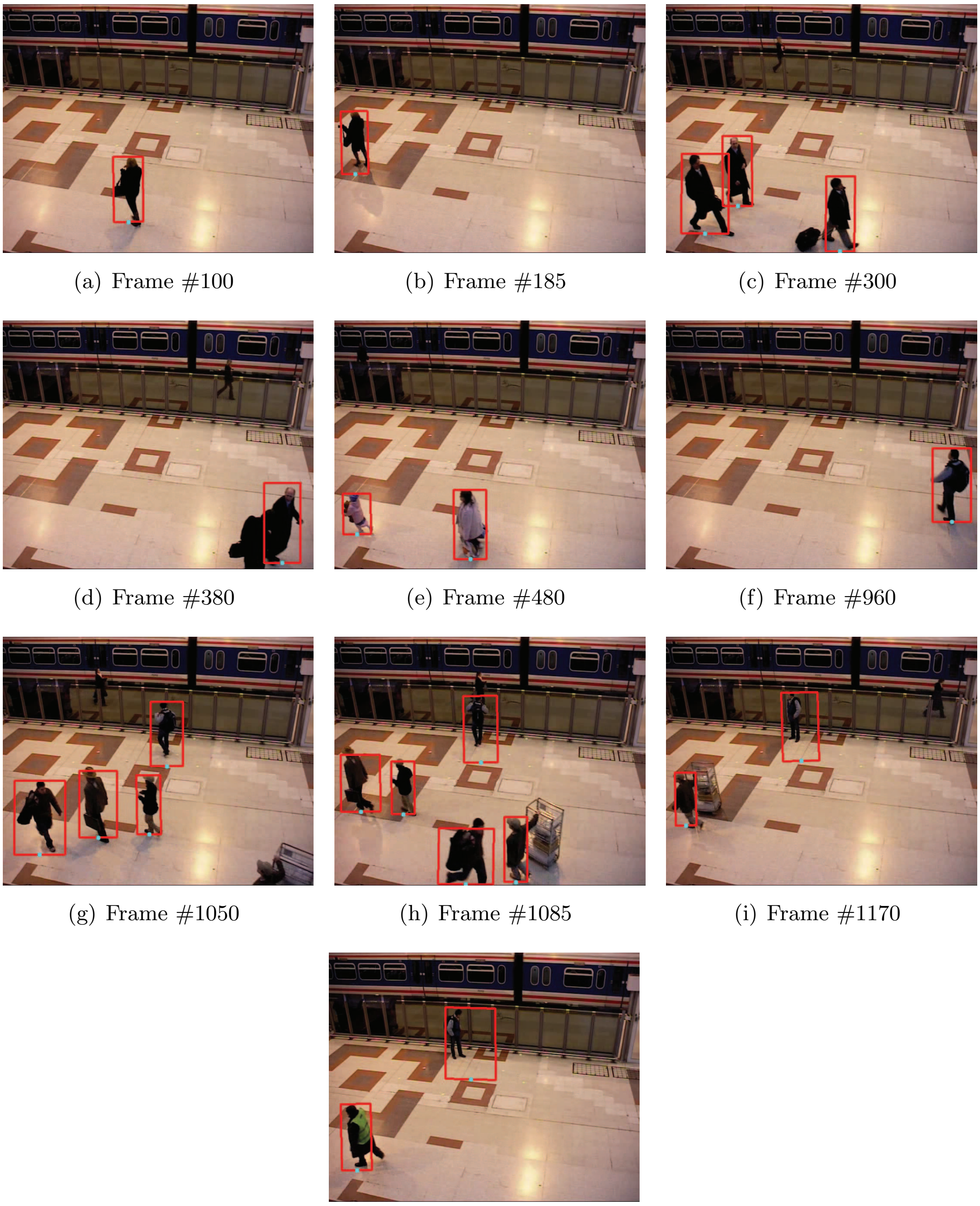
Figure 4.7: Spatial weights used in the experiments. Brighter regions correspond to high values (close to ${}^{s}\omega_{\max} = 2.0$) and darker regions to spatial weights close to 1.0.

one of the corners of the tracked region goes at least 2px outside of the image region.

First, we have applied the IVT and the MOTOH Pedestrian Tracker to the first 1700 frames of the sequence S1-T1-C of Camera 3 of the PETS2006 Dataset[1]. In this sequence there are no big difficulties, excepting a pedestrian that turns around himself several times, showing to the camera his back and his front and therefore changing considerably his appearance. This pedestrian is not well tracked with the IVT algorithm nor the MOTOH Pedestrian Tracker. The reasons however are slightly different. While in IVT the turns of the pedestrian causes a model drift, in the MOTOH Pedestrian Tracker these turns generate a false occlusion signal. The results using IVT are shown in Figure 4.8 and the results using the MOTOH Pedestrian Tracker in Figure 4.9.

An artificial occlusion has been generated in the same sequence from PETS 2006, simulating as if the first pedestrian passes behind a column. The occlusion has been done by simply copying another region of the image. The IVT algorithm fails to avoid the artificial column, as expected, while the MOTOH Pedestrian Tracker correctly detects an occlusion and switches to behavioural tracking. The end of the occlusion is also correctly detected, switching again to visual tracking. The

[1] Available at http://www.cvg.reading.ac.uk/PETS2006/data.html (last visited in june 2011)



(a) Frame #100

(b) Frame #185

(c) Frame #300

(d) Frame #380

(e) Frame #480

(f) Frame #960

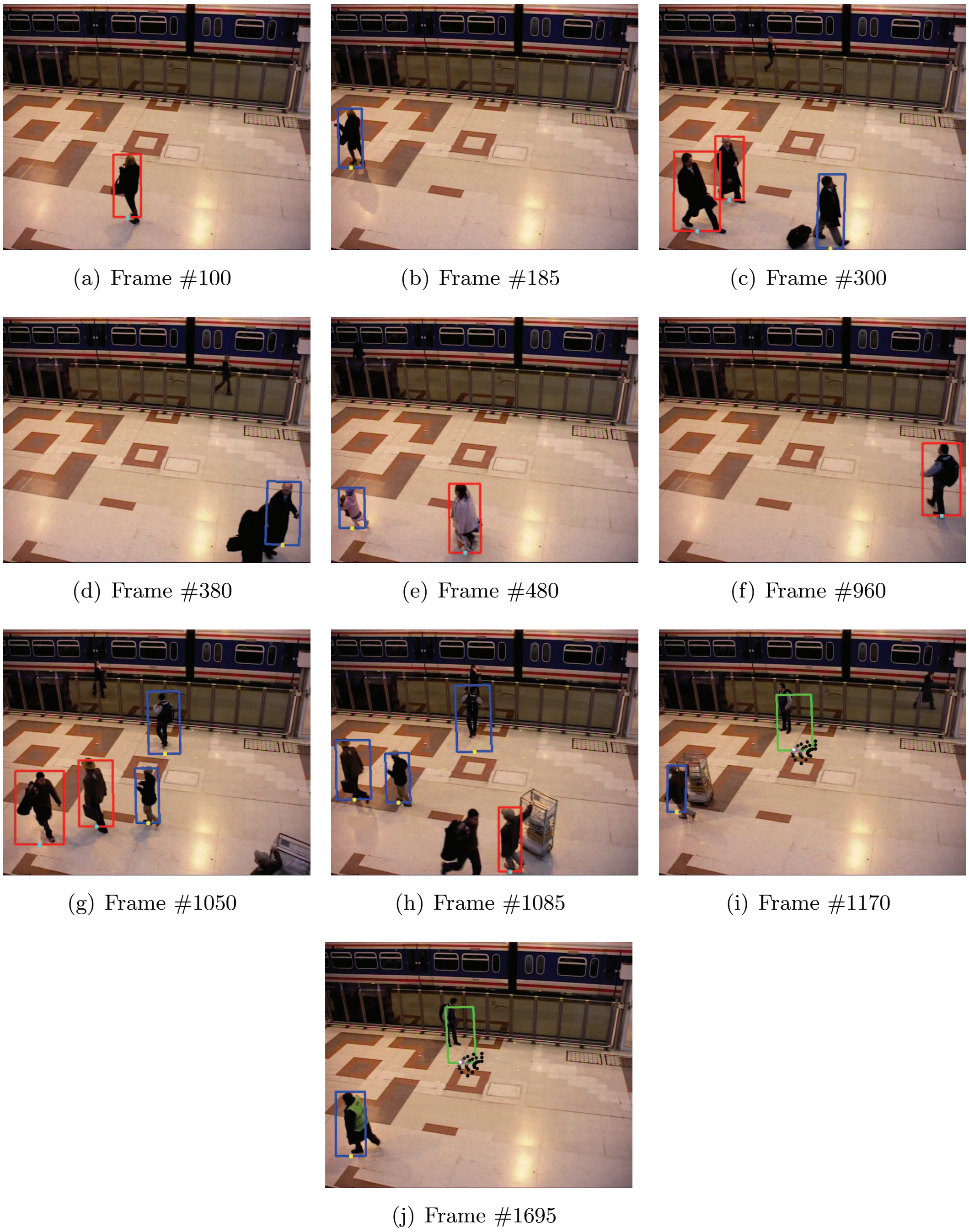
(g) Frame #1050

(h) Frame #1085

(i) Frame #1170

(j) Frame #1695

Figure 4.8: The IVT algorithm applied to the first 1700 frames of the sequence S1-T1-C of Camera 3 of the PETS2006 Dataset. The only pedestrian with problems for being well tracked is the one in the centre of Figure 4.8(j).



(a) Frame #100

(b) Frame #185

(c) Frame #300

(d) Frame #380

(e) Frame #480

(f) Frame #960

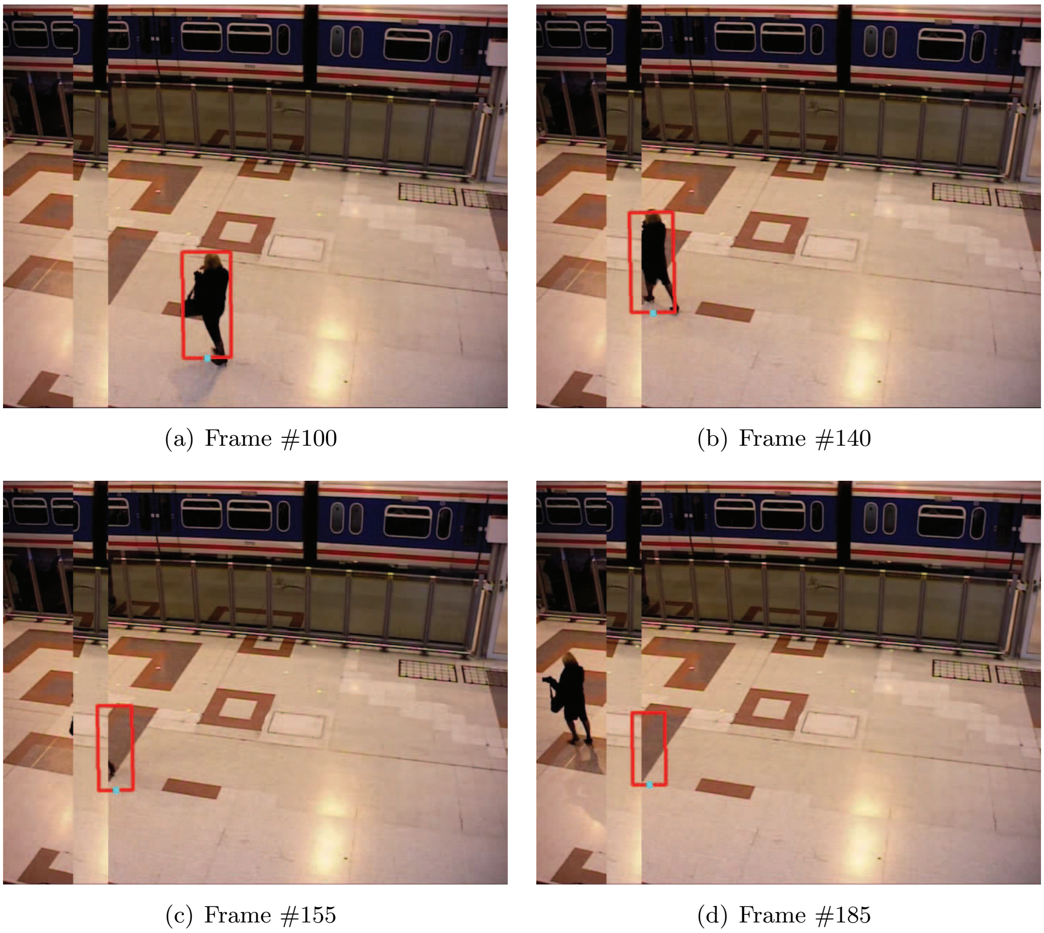
(g) Frame #1050

(h) Frame #1085

(i) Frame #1170

(j) Frame #1695

Figure 4.9: The MOTOH Pedestrian Tracker applied to the first 1700 frames of the sequence S1-T1-C of Camera 3 of the PETS2006 Dataset. The only pedestrian with problems for being well tracked is the one in the centre of Figure 4.9(j).



(a) Frame #100

(b) Frame #140

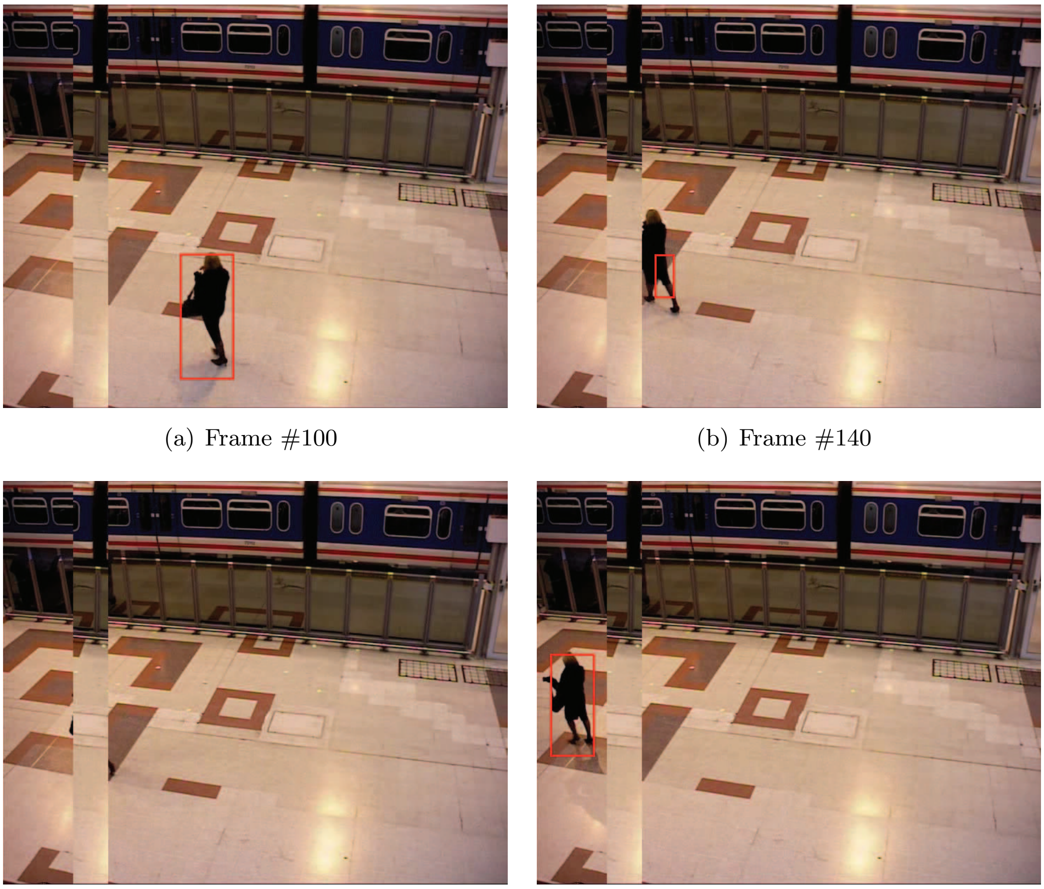
(c) Frame #155

(d) Frame #185

Figure 4.10: The IVT algorithm applied to the first pedestrian of the sequence S1-T1-C of Camera 3 of the PETS2006 Dataset. An occluded region has been artificially generated by copying another portion of the image on it.

TLD algorithm also avoids correctly the artificial occlusion generated. The results with IVT are shown in Figure 4.10, with TLD in Figure 4.11 and those with the MOTOH Pedestrian Tracker in Figure 4.12.

A video sequence with real occlusions, called the SV sequence, has been collected. In this sequence there are two columns that generate occlusions, but given the recording angle of the camera, multiple occlusions between pedestrians are also generated. Three pedestrians walk in the scene. The IVT algorithm fails in the tracking of all the pedestrians, one of them without even an occlusion. The TLD algorithm has serious problems with clutter in the scene, misleading several times the tracked regions, and only tracking approximately correctly one of the pedestrians. The MOTOH Pedestrian Tracker, however, correctly tracks all the pedestrians, only not correctly recovering the last occlusion of one of



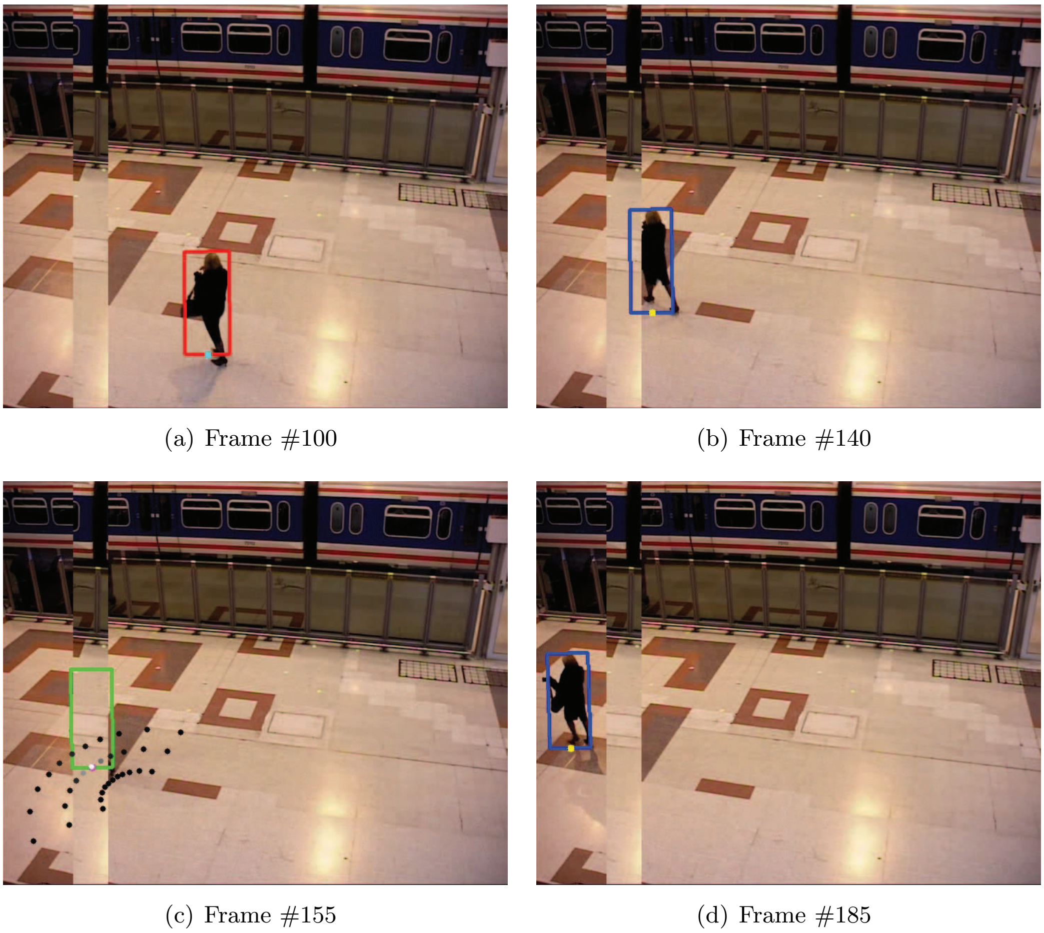
(a) Frame #100

(b) Frame #140

(c) Frame #155

(d) Frame #185

Figure 4.11: The TLD algorithm applied to the first pedestrian of the sequence S1-T1-C of Camera 3 of the PETS2006 Dataset. An occluded region has been artificially generated by copying another portion of the image on it.



(a) Frame #100

(b) Frame #140

(c) Frame #155

(d) Frame #185

Figure 4.12: The MOTOH Pedestrian Tracker applied to the first pedestrian of the sequence S1-T1-C of Camera 3 of the PETS2006 Dataset. An occluded region has been artificially generated by copying another portion of the image on it.

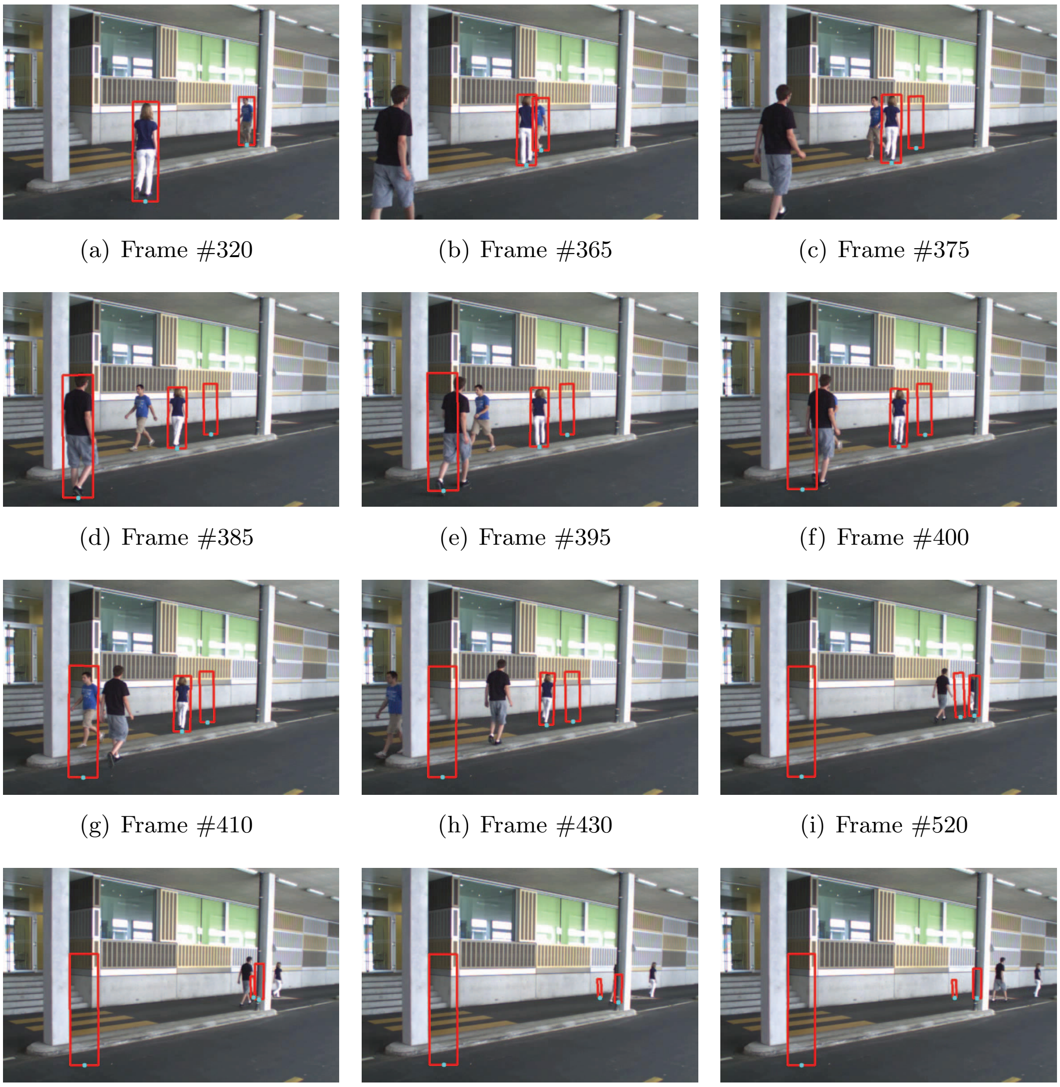
the pedestrians. Indeed, the MOTOH Pedestrian Tracker correctly tracks the pedestrian that is lost by IVT without any occlusion, showing the superiority of the ITWVTSP algorithm. Furthermore, from the five occlusions generated in this sequence, four are successfully handled. The other one is close (in time) to a previous occlusion, which probably avoided a proper update of the visual model with new information after the previous occlusion and generates the failure in the visual tracking recovery. The results are shown in Figure 4.13, Figure 4.14 and Figure 4.15 for IVT, TLD and the MOTOH Pedestrian Tracker, respectively.

Another video sequence, called the SV-CAR sequence, was also collected in the same place and from the same point than the SV sequence. In this sequence a single pedestrian crosses the scene. In addition to the occlusions generated by the columns, a vehicle crosses the scene, hiding momentarily the pedestrian. The IVT and the TLD algorithms loss the track due to the vehicle, while the MOTOH Pedestrian Tracker correctly detects the occlusion and switches to behavioural tracking while the pedestrian is hidden. After the pass of the vehicle, the occlusion detector correctly switches again to visual tracking. The occlusion generated by the column is then correctly detected too, although suddenly, the algorithm switches again to visual tracking, sticking the tracked region to the column, where it stays, losing the track of the pedestrian. The results are shown in Figure 4.16 for IVT, in Figure 4.17 for TLD and in Figure 4.18 for the MOTOH Pedestrian Tracker.

4.5 Conclusions and Future Work

In this Chapter we have introduced the Model cOrruption and Total Occlusion Handling (MOTOH) framework for visual tracking. This framework describes the combination of a visual model with a behavioural model for tracking. The visual model is computed on-line with the ITWVTSP algorithm (see Chapter 3). The behavioural model is used for tracking the target using behavioural information when there is an occlusion. The switch between the visual and the behavioural tracking is governed by a total occlusion detector that we have introduced. This detector computes some scores about the particle filter in charge of the tracking, for detecting a drop in the amount of visual information of the tracked object, i.e. an occlusion of this object. The occlusion detection algorithm introduced gets adapted dynamically to the weights of each object of interest, making the procedure application-independent.

We have applied the framework to pedestrian tracking by plugging the Discrete Choice Pedestrian Model (DCPM) in the “Behavioural Weights” block. Several tests in challenging sequences have been performed, showing the added value of combining visual and behavioural tracking and the capacity of the MOTOH Pedestrian Tracker to handle complex situations. However, when several occlusions



(a) Frame #320

(b) Frame #365

(c) Frame #375

(d) Frame #385

(e) Frame #395

(f) Frame #400

(g) Frame #410

(h) Frame #430

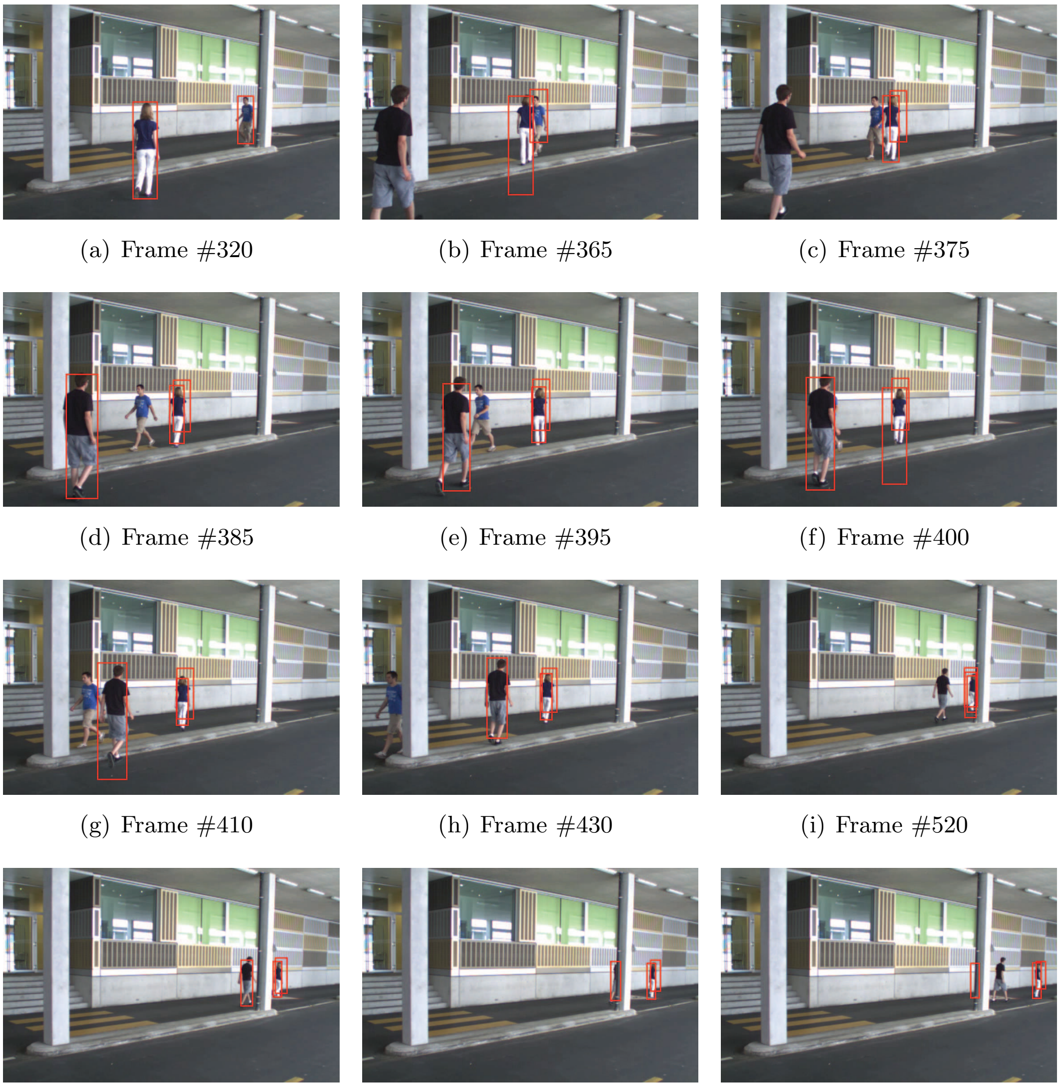
(i) Frame #520

(j) Frame #545

(k) Frame #560

(l) Frame #585

Figure 4.13: The IVT algorithm applied to the SV sequence.



(a) Frame #320

(b) Frame #365

(c) Frame #375

(d) Frame #385

(e) Frame #395

(f) Frame #400

(g) Frame #410

(h) Frame #430

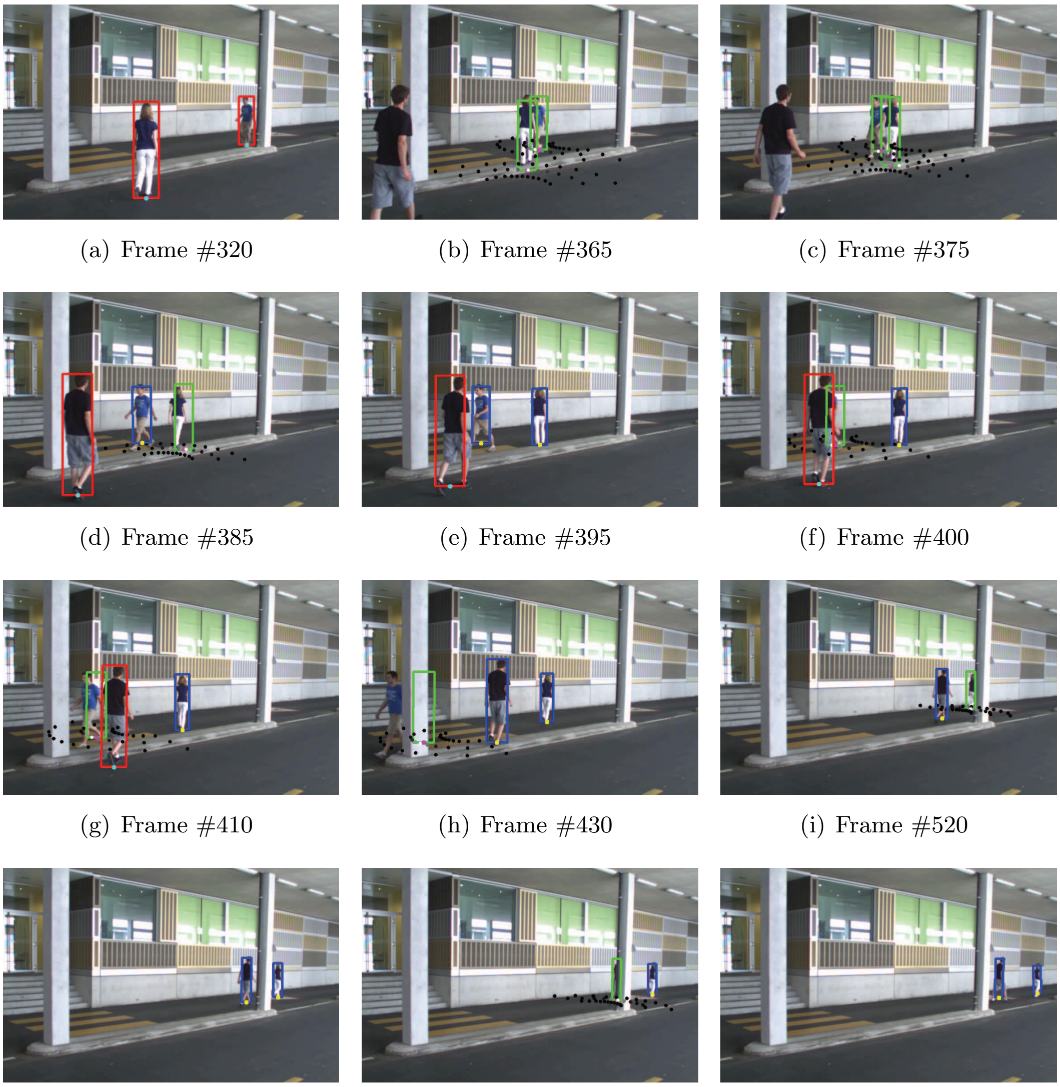
(i) Frame #520

(j) Frame #545

(k) Frame #560

(l) Frame #585

Figure 4.14: The TLD algorithm applied to the SV sequence.



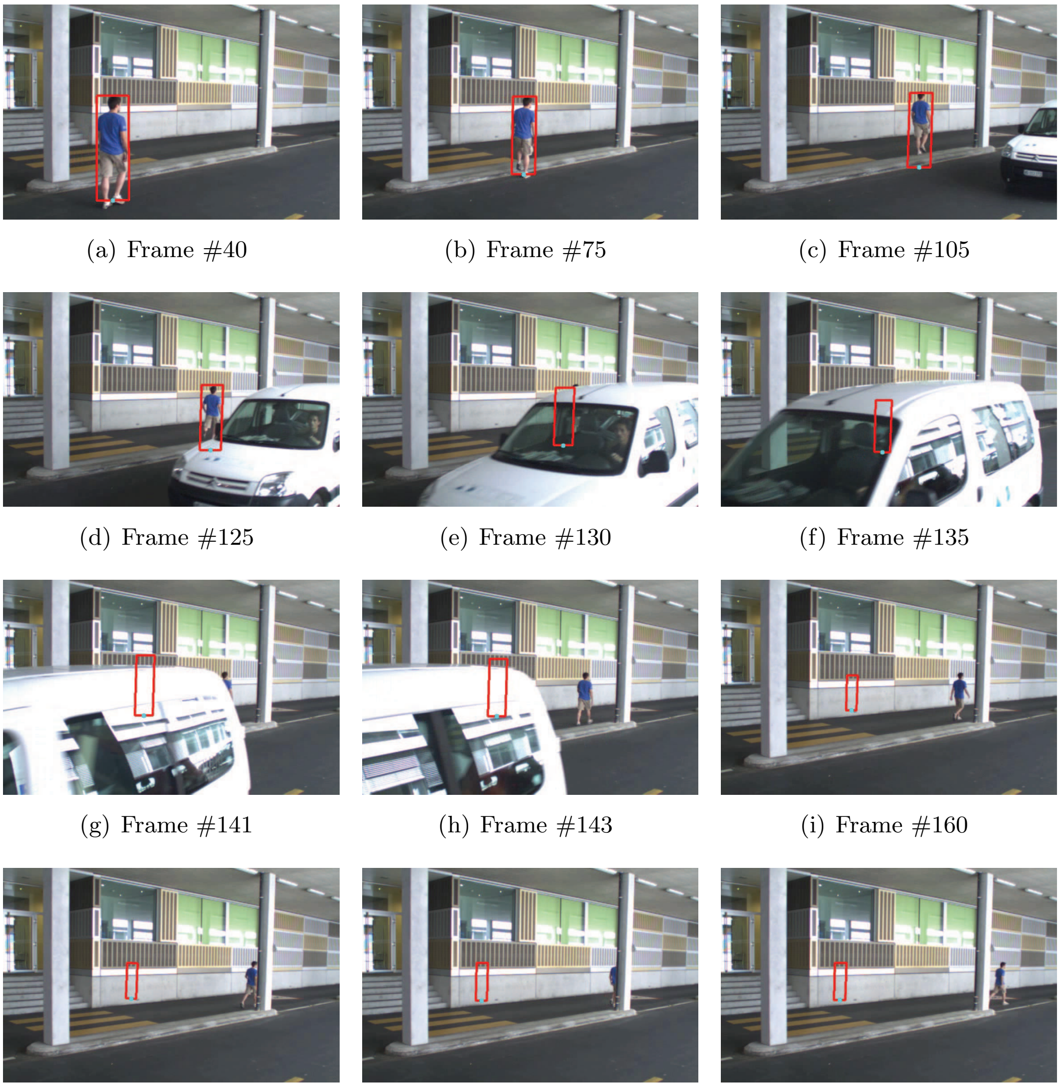
(a) Frame #320 (b) Frame #365 (c) Frame #375

(d) Frame #385 (e) Frame #395 (f) Frame #400

(g) Frame #410 (h) Frame #430 (i) Frame #520

(j) Frame #545 (k) Frame #560 (l) Frame #585

Figure 4.15: The MOTOH Pedestrian Tracker applied to the SV sequence.



(a) Frame #40

(b) Frame #75

(c) Frame #105

(d) Frame #125

(e) Frame #130

(f) Frame #135

(g) Frame #141

(h) Frame #143

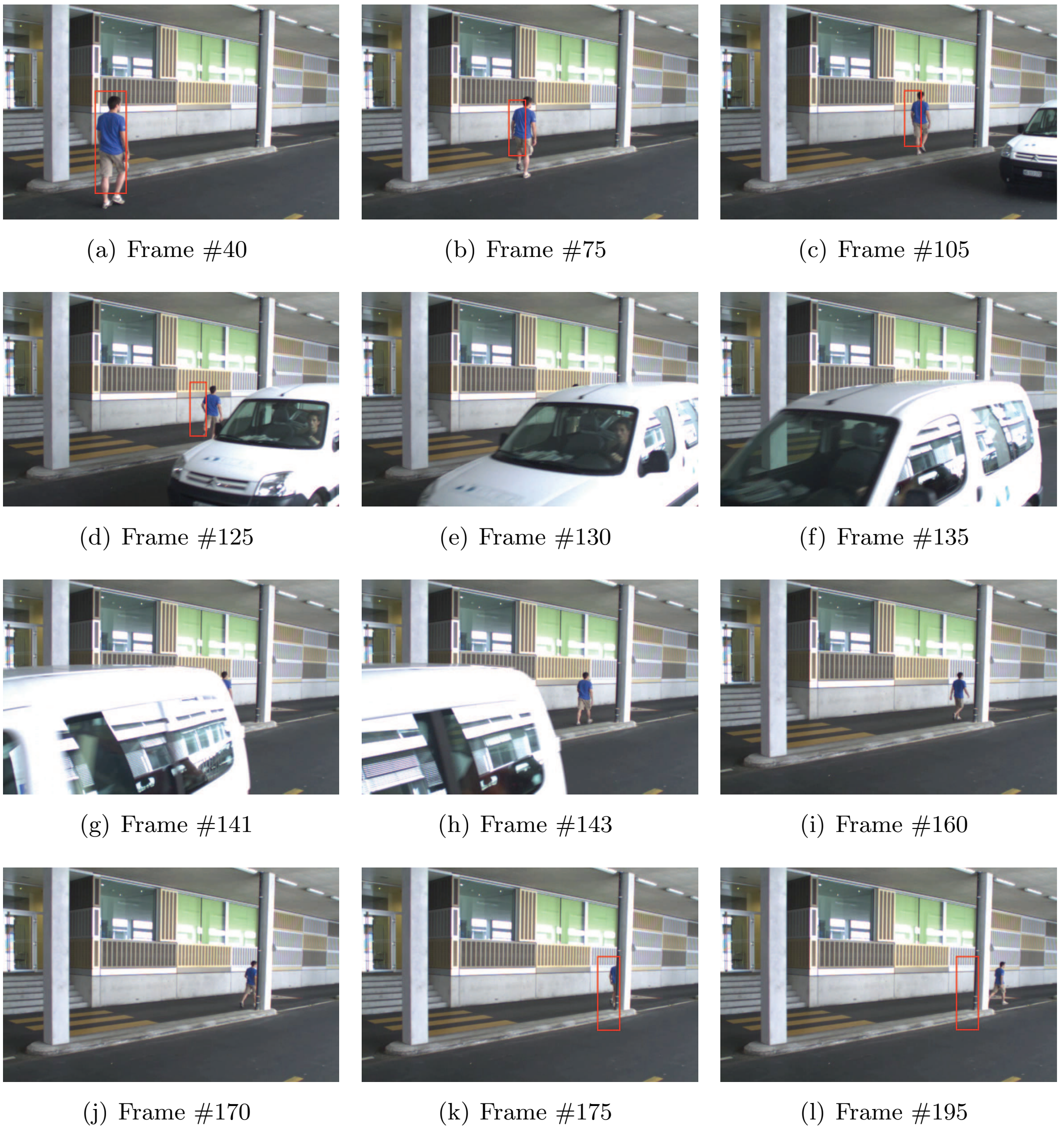
(i) Frame #160

(j) Frame #170

(k) Frame #175

(l) Frame #195

Figure 4.16: The IVT algorithm applied to the SV-CAR sequence.



(a) Frame #40

(b) Frame #75

(c) Frame #105

(d) Frame #125

(e) Frame #130

(f) Frame #135

(g) Frame #141

(h) Frame #143

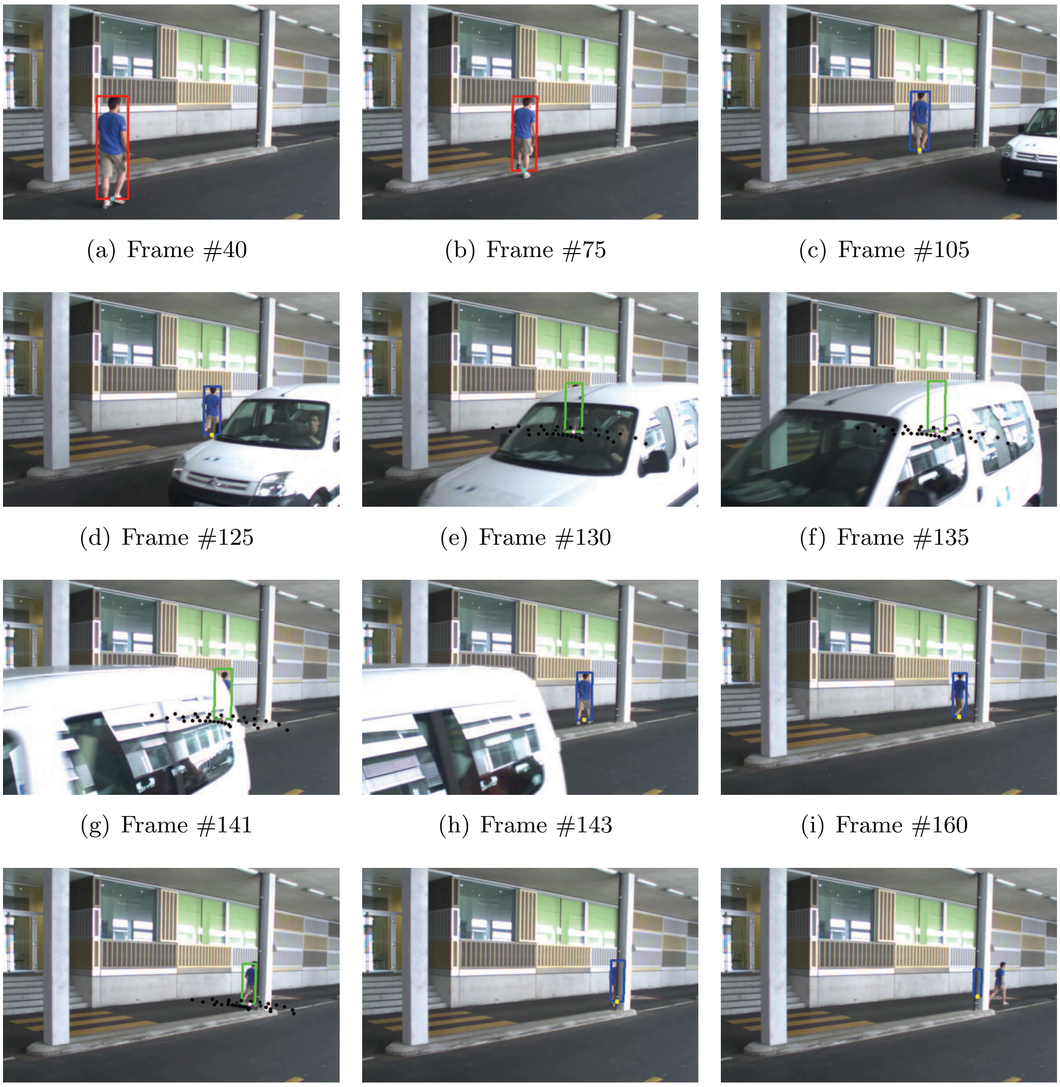
(i) Frame #160

(j) Frame #170

(k) Frame #175

(l) Frame #195

Figure 4.17: The TLD algorithm applied to the SV-CAR sequence.



(a) Frame #40

(b) Frame #75

(c) Frame #105

(d) Frame #125

(e) Frame #130

(f) Frame #135

(g) Frame #141

(h) Frame #143

(i) Frame #160

(j) Frame #170

(k) Frame #175

(l) Frame #195

Figure 4.18: The MOTOH Pedestrian Tracker applied to the SV-CAR sequence.

happen close in time, the system seems to have some troubles recovering from the occlusion. The reason is probably the batch process of samples in the ITWPCA algorithm. This batch process does not allow a fast adaptation of the visual model to the target conditions just after the occlusion. Strategies for adapting the batch size for fast adaptation of the visual model after an occlusion should be studied. These strategies could consider a change in the batch size, as well as changes in the forgetting factor and the temporal weights for a fast update of the visual model after occlusions.

As commented above, two potential destinations are considered and one of them is chosen according to the tracking during the first frames. The destination has an important impact in the DCPM and more complex destination estimation strategies should be developed. In particular, in the context of a long time monitoring system, origin and destination estimation could be done by tracking. These data could be used for generating dynamically origin/destination maps for updating destinations of new pedestrians on the scene. The impact on the tracking performance of this procedure should be quantified.

In this Chapter, we have applied the MOTOH Pedestrian Tracker on a mono-camera environment, but note that the algorithm could be used as a multi-camera tracking algorithm. Indeed, dead zones between cameras can be treated as occlusions, and the “transfer” of a pedestrian from a camera to another one be done by behaviourally tracking the pedestrian.

Here we close the part of this dissertation dedicated to visual tracking. The MOTOH framework includes procedures for avoiding model drift (introduced in Chapter 3) and for keeping tracking while the object is occluded (Chapter 4). This deals with hard problems due to exogenous sources of variability (see Chapter 1), producing a robust global tracking algorithm. In addition, we have applied the MOTOH framework to pedestrian tracking by using a pedestrian walking behaviour model calibrated and validated on real data. This combination of the human behaviour modelling field and the computer vision field is not very common.

In the next chapter we will study the problem of matching features between omnidirectional images or between omnidirectional and planar images. For that, a scale invariant feature transform on the sphere is developed and two local descriptors are introduced. Given the matching between omnidirectional and planar images, a mapping for segmenting regions in an omnidirectional image given the region in a planar image is also introduced. Let us note that, under the hypothesis of being placed on rigid objects, the features and descriptors that will be introduced in next Chapter are more robust than grayscale templates as used in Chapter 3 and Chapter 4. However, given that for the tracking we dealt with deformable objects, PCA performed on templates are more suitable.

Chapter 5

Scale Invariant Feature Transform on the Sphere: Matching Features in Omnidirectional Images

Contents

5.1 Introduction

The constant increase of computational power allows CV researchers to develop more complex algorithms that slowly but surely are approaching human vision performance. However, hardware restrictions imposed by visual sensors are sometimes hard to handle. A typical example is the narrow field of view provided by a regular projective camera, which can make difficult for instance to estimate motion on a scene. For this reason, research on new visual sensor geometries has been gaining attention during the last years. An example of these new visual sensor geometries are omnidirectional visual sensors.

Omnidirectional vision has become an important topic in computer vision. One of its main benefits is that one omnidirectional camera can cover 360° around it. As with conventional (planar) images, image matching is a main aspect of many computer vision problems involving omnidirectional images, although it has not been widely studied for this kind of cameras yet. Usually, techniques designed for planar images are applied on omnidirectional images, like for instance on panoramic images, i.e. omnidirectional images mapped on a cylinder, [Yuen 2005, Bur 2006], although this is not geometrically correct. Even if locally those algorithms are still valid, as soon as bigger regions of the image are considered, it is not the case anymore due to the deformation that the omnidirectional sensor introduces. Moreover, not only do omnidirectional and planar images coexist, but they are often used jointly, for instance in hybrid camera networks. This is a source of new problems, since matching between omnidirectional images is needed, but also between omnidirectional and planar ones.

A common way of tackling the matching problem between two given images is by using interest points. These are points in an image that fulfill some "interest" criterion. This criterion is usually defined in such a way that the obtained points have a well-defined position, contain as much local information on the surroundings as possible and are robust against changes in the image, such as noise, perspective transformations, illumination changes, etc. The location of these points is often used for extraction of local image descriptors. This is a transformation of the local image data into an element of the transformed space, usually a vector, where some characteristics are coded, as for example the shape, the orientation, the colour, the texture, etc. They can be used afterwards, for instance, in matching or image registration. One of the most well-known and employed interest points are Scale Invariant Feature Transform (SIFT) points. SIFT is also used in tracking, given its robustness to image changes and noise, especially on rigid objects. The drawback is the computational cost of the feature extraction, which in tracking forces to design strategies for computing SIFT only on small regions of the image [Zhou 2009].

The intuitive approach for defining and performing a scale invariant feature transform on omnidirectional images is to first map them on a panoramic image (i.e. in cylindrical coordinates) and then to apply the conventional SIFT algorithm. In fact, using the same reasoning, the classical SIFT has been applied to unwrapped omnidirectional images [Goedeme 2005, Tamimi 2006, Valgren 2007, Scaramuzza 2008]. The difficulties in this case come when there is information in the extremities of the omnidirectional image. Such images are obtained by spherical omnidirectional sensors.

On the one hand, images obtained by omnidirectional sensors contain significant deformations. Specific mapping, like panoramic or log-polar mappings, attempt to reduce somehow the distortions but do not succeed completely. A natural choice of a non-deformed domain for the full sphere of view, where there are no limitations on the zenithal range, is the sphere $S^2 \in \mathbb{R}^3$. On the other hand, the scale-invariant feature transform is based on distinctive invariant features from images for further matching. The features are invariant to image scale and rotation.

Considering the spherical geometry, which is the natural manifold for any omnidirectional image, we need to first recall the basic affine spherical transformations. In general, two types of transformations on the sphere are distinguished: motions (displacements) and dilations (scalings). Concerning the motions, there are three possible rotations. In particular, these are rotations by angles $\varphi \in [0, 2\pi), \psi \in [0, 2\pi)$ and $\theta \in [0, \pi]$. In other words, rotations by φ are those around the x_0-axis; rotations by θ are those around x_2- axis, and rotations by ψ, are rotations of the point on the sphere around itself (see Figure 5.1). The dilations affect the angle θ, and the motions the angles φ or ψ. Furthermore, a translation on the plane corresponds to a rotation on the sphere. Therefore, the notions of dilation and rotation on the plane cannot be simply translated to the sphere. That is why we cannot apply the standard SIFT paradigm to a complex data that is defined in spherical coordinates (as for instance, the omnidirectional image after it has been mapped onto the sphere).

Finally, from here it is clear that applying the standard SIFT on unwrapped omnidirectional images is locally valid, i.e. for big radius of curvature and, consequently, at small scales. However, if a global algorithm is wanted, SIFT must take place in spherical geometry. Having a SIFT algorithm that operates directly on the sphere is equivalent to preserve the spherical affine transformations and only in this conditions the transformations are geometrically correct.

In this Chapter, a SIFT algorithm in spherical coordinates and a new approach to match points between two spherical images or between spherical and planar images are proposed.

First, we develop an interest point extractor on the sphere based on the spherical scale-space representation and the SIFT algorithm. This algorithm processes omnidirectional images mapped on the sphere (see Figure 5.2(c)). The creation procedure of the spherical scale-space is speeded up by successive downsampling of the input image for each octave. This down-sampling generates an aliasing effect when the Spherical Fourier transform is applied at the corresponding level. For this reason, an anti-aliasing criterion is defined to decide whether an image is down-sampled or not.

Secondly, we propose two types of descriptors. The first is used for matching

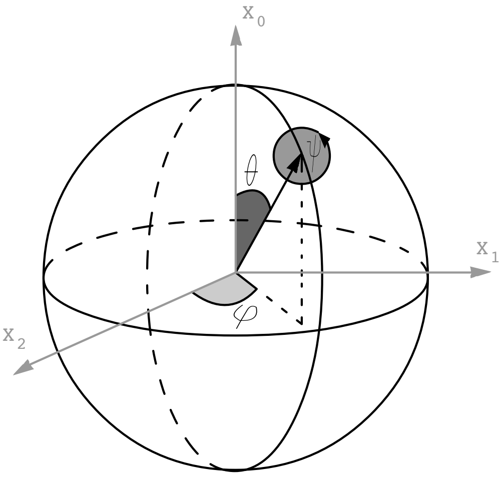


Figure 5.1: Rotations on the sphere

between two spherical images, the second for matching a spherical and a planar image. Both descriptors can be useful when working, for instance, with hybrid camera networks. In such a case, these descriptors can easily help registering data from all the components of the network.

Finally, we introduce a mapping between planar and spherical images. This mapping sends the contour or regions of an object in a planar image to a spherical one and vice versa. The parameters of this mapping are estimated by means of the obtained matched points, cleaning false detections with the Random Sample Consensus (RANSAC) algorithm [Fischler 1981]. The inputs of the estimation process are respectively the matched points from the omnidirectional and the planar images.

Several experiments are performed on real omnidirectional images to test the proposed algorithms. The code developed for these tests has been implemented in Matlab® and source code and images are freely available[1] under the GPL license. The source code requires the installation of the "Yet Another Wavelet Toolbox" (YAWTb)[2] for MatLab. This library provides an efficient way of computing the spherical harmonic transformations as well as a nice visualisation interface. Finally, for the RANSAC routines, we used the RANSAC Toolbox[3].

This Chapter is organised as follows. In Section 5.2, a review of the state-of-the-art in interest points and local descriptors in planar and omnidirectional images is shown. In Section 5.3, the mathematical aspects of the interest point extraction on the sphere are exposed. In Section 5.4, the proposed algorithm is described in detail, as well as the two proposed descriptors. Then, in Section 5.5, a "planar to

[1] http://transp-or2.epfl.ch/pagesPerso/javierFiles/software.php
[2] http://rhea.tele.ucl.ac.be/yawtb
[3] http://vision.ece.ucsb.edu/~zuliani/Code/Packages/RANSAC/

spherical" mapping is defined and a method for its estimation is given. In Section 5.6, several experimental results are presented. Finally, in Section 5.7, some conclusions and potential lines for future research are given. The works presented in this Chapter have been submitted to the International Journal of Computer Vision and are nowadays in the second revision round.

5.2 State-of-the-art

Interest points are widely used nowadays by computer vision algorithms. As commented before, two main aspects make these points useful:

- robustness against image changes and,
- richness of local information in terms of local image structure.

A wide variety of interest points has been defined to best combine these two aspects, like for example Harris-Stephens corners [Harris 1988], SUSAN corners [Smith 1997], salient regions [Kadir 2001], Maximally Stable Extremal Regions (MSER) [Matas 2002] or extrema of the Difference-of-Gaussians (DoG) [Lowe 2004]. An excellent survey on this kind of points can be found in [Tuytelaars 2007]. Due to their stability, these keypoints are often used for the computation of local descriptors, which are used afterwards for several tasks, such as tracking, object detection or region matching. A wide variety of local descriptors has also been proposed in the literature [Zabih 1994, Van Gool 1996, Baumberg 2000, Lowe 2004, Mikolajczyk 2005a]. An exhaustive comparison of local descriptors has been given in [Mikolajczyk 2005a].

Scale Invariant Feature Transform (SIFT), introduced in [Lowe 2004], is a well-known algorithm that successfully combines both notions. For interest points, it considers extrema of the Difference-of-Gaussians, and for local descriptors, a histogram of orientations. The SIFT algorithm detects points in a scale-invariant way, as extrema in the response of the convolution of the image with a DoG function

$$\psi(x, y, \sigma) = g(x, y, k\sigma) - g(x, y, \sigma), \tag{5.1}$$

where $g(x, y, \sigma)$ denotes a two-dimensional Gaussian kernel with standard deviation σ. This is based on the work of [Lindeberg 1998], and the convolution of an image with $\psi(x, y, \sigma)$ can be computed as the difference of consecutive images in the scale-space representation of the image, choosing properly the value of k. The scale-space representation $L(x, y, t) : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$ of an image $I(x, y)$ can be equivalently defined in two different ways. The first one is the evolution over time of the heat distribution $I(x, y)$ in an infinite homogeneous medium:

$$\partial_t L(x, y, t) = \frac{1}{2}\nabla^2 L(x, y, t), \tag{5.2}$$

where the initial condition is $L(x, y, 0) = I(x, y)$. The second one is the successive convolution of the image with a Gaussian kernel, $g(x, y, \sigma)$, of standard deviation $\sigma = \sqrt{t}$:

$$L(x, y, \sigma) = g(x, y, \sigma) * I(x, y). \tag{5.3}$$

This scale-space representation of an image is efficiently computed directly using the definition of the convolution, thanks to the separability of the Gaussian filter. The local data around each interest point is then used to compute SIFT descriptors. These local descriptors are invariant to rotation and scale changes. They consist of a three-dimensional histogram: two spatial dimensions and one dimension for orientations. The size of this region depends on the scale at which the point has been detected. Thanks to its simplicity, good results in terms of repeatability and accuracy on matching, it has been used to treat applications requiring tracking or matching of regions [Sirmacek 2009, Brox 2010].

Several variants of the SIFT algorithm have appeared, trying to improve the interest point extraction or the local descriptor. Among those trying to improve the interest point extraction, the most remarkable representative is probably the Speed-Up Robust Features (SURF) algorithm [Bay 2008]. For those trying to improve the local descriptor, a good representative is the Gradient Location and Orientation Histogram (GLOH) introduced in [Mikolajczyk 2005a].

All these algorithms and techniques have been developed to work with regular (planar) images or videos. Over the last years, though, omnidirectional imaging has become an important topic, due to both the availability of simple sensors (e.g. parabolic mirrors mounted on regular cameras) and the great advantages it provides (e.g. a 360 degrees view in one single image). This kind of sensors has a lot of applications, such as video surveillance [Boult 2001] or object tracking [Chen 2008], and their use has become common in robot navigation [Menegatti 2006] and in autonomous vehicles [Ehlgen 2008, Scaramuzza 2008].

Interest points and local descriptors-based techniques, such as SIFT, have been applied to omnidirectional images due to their good performance in planar images [Goedeme 2005, Tamimi 2006, Valgren 2007, Scaramuzza 2008]. Recently, several efforts have been made to develop algorithms specifically designed to treat these omnidirectional images [Bogdanova 2007, Hadj-Abdelkader 2008]. An important aid in this sense were the results of [Geyer 2001], where the authors showed that the most common catadioptric omnidirectional images (elliptic, parabolic and hyperbolic) can be bijectively mapped on the surface of a sphere. In particular, for the case of parabolic images, a parabolic projection is equivalent to the composition of normalisation to the unit sphere followed by stereographic projection (see Figure 5.2 for an example). Consequently, a whole family of omnidirectional images can be processed by algorithms treating spherical images. The mapping from the captured image to the sphere is the only adaptation needed for each element of the family. Based on this result, [Hansen 2007b, Hansen 2007a] developed a SIFT-like algorithm on the sphere to match points between wide-angle images. In this algorithm, the point extraction is computed on the back-projection of the spherical scale-space to the wide-angle image plane, and the descriptor is computed using a fixed size patch of 41 × 41 pixels around each extracted point at the corresponding scale. Also, in [Mauthner 2006] an interest region matching in omnidirectional images, which uses virtual camera planes, has been developed.

5.3 Spherical scale-space

5.3.1 Spherical Geometry

The 2-sphere ($S^2 \in \mathbb{R}^3$) is a compact manifold of constant positive curvature. In spherical coordinates, each point on the sphere is a three-dimensional vector

$$\omega = (x_0, x_1, x_2) \equiv (r\cos\theta, r\sin\theta\sin\varphi, r\sin\theta\cos\varphi),$$

with $r \in (0,\infty), \theta \in [0,\pi]$ and $\varphi \in (0,2\pi]$ as illustrated in Figure 5.3(a). Figure 5.3(b) also illustrates the so called stereographic projection from the South Pole, a projection that maps any point of the sphere onto a point of the tangent plane at the North Pole. If we take the sphere S^2 as the Riemannian sphere ($r = 1$) and the tangent plane as the complex plane $\mathbb{C}^2$, then the stereographic projection is a bijection given by

$$\Phi(\omega) = 2\tan\frac{\theta}{2}(\cos\varphi, \sin\varphi), \tag{5.4}$$

where $\omega \equiv (\theta,\varphi),\ \theta \in [0,\pi], \varphi \in [0,2\pi)$.

5.3.2 Fourier Transform on the Sphere

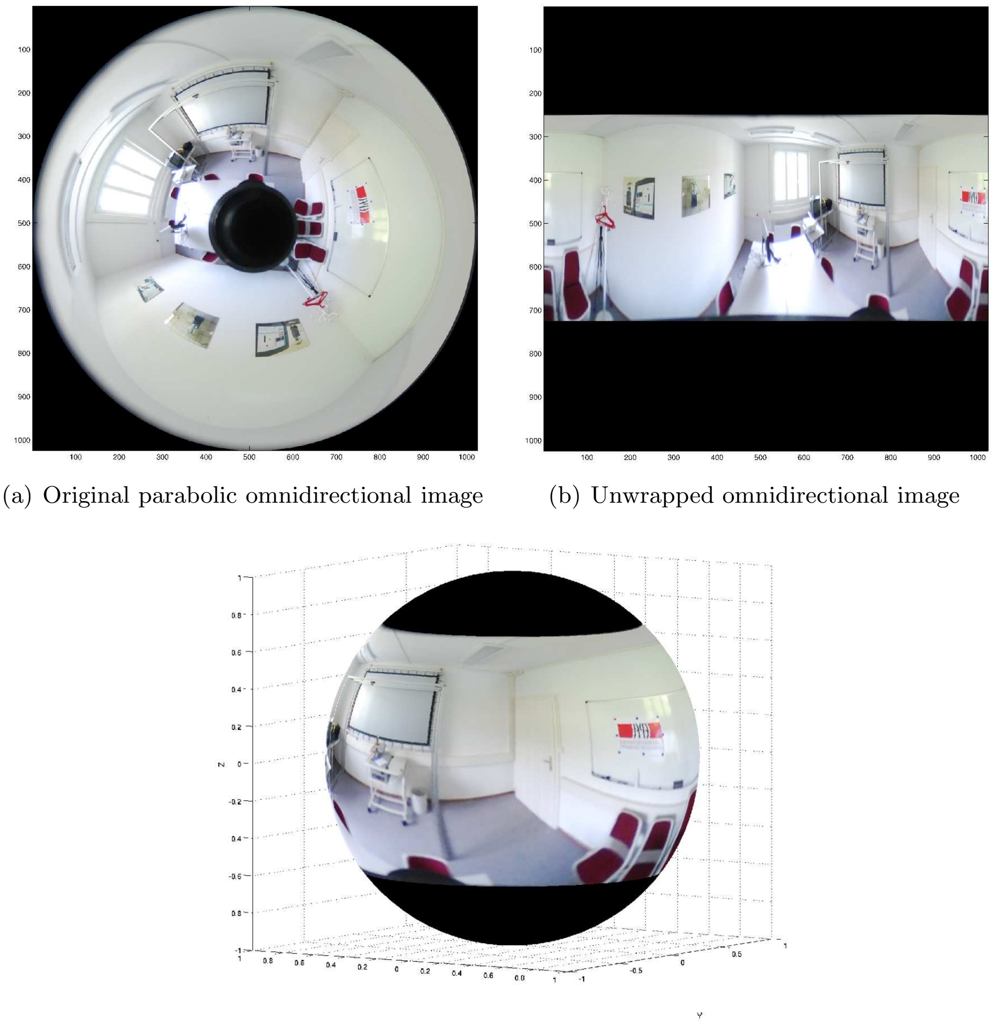
Let us consider two functions $f, h \in L^2(S^2)$ defined on the 2-sphere $S^2 \in \mathbb{R}^3$. Then, the convolution on the sphere reads

$$(f * h)(\omega) = \int_{r\in SO(3)} f(r\eta)h(r^{-1}\omega)dr, \tag{5.5}$$

where $\omega \equiv (\theta,\varphi) \in S^2$, $\theta \in [0,\pi]$, $\varphi \in [0,2\pi)$ (see Figure 5.3(a)), $SO(3)$ is the rotation group and η is the vector pointing to the north pole. Equation (5.5) is hard to compute, but as it was demonstrated by Driscoll and Healy in [Driscoll 1994], the convolution of two spherical functions $f, h \in L^2(S^2)$ can be calculated more efficiently as the point-wise product of their spherical Fourier transforms:

$$\widehat{(f * h)}(\ell, m) = 2\pi\sqrt{\frac{4\pi}{2\ell+1}}\widehat{f}(\ell,m)\widehat{h}(\ell,0), \tag{5.6}$$

where $\widehat{(\cdot)}$ is the spherical Fourier transform of the function.

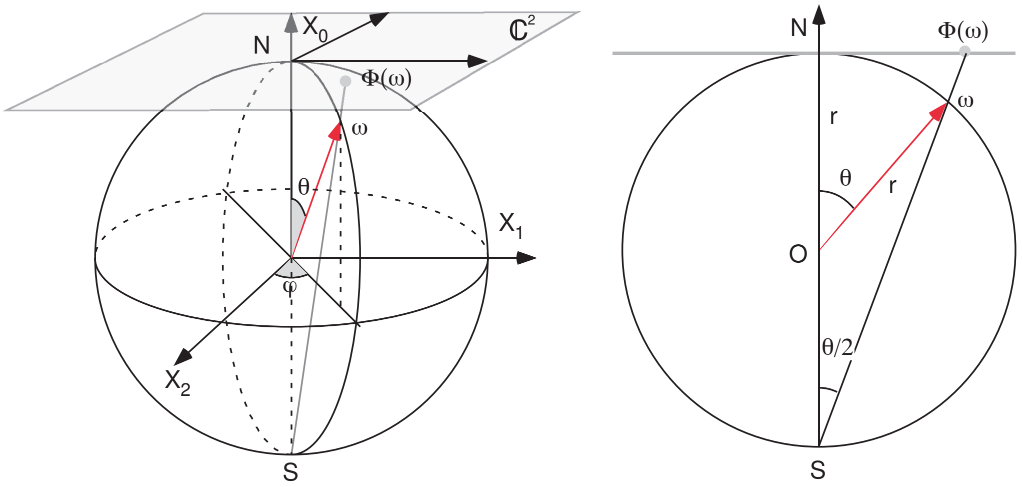


(a) Original parabolic omnidirectional image

(b) Unwrapped omnidirectional image

(c) Omnidirectional image mapped on the unit sphere

Figure 5.2: Example of mapping a parabolic omnidirectional image on the sphere. The unwrapped spherical image (Figure 5.2(b)) is often used for visualisation purposes.



(a) Spherical coordinates

(b) Cross-section of a stereographic projection through the South Pole

Figure 5.3: Spherical geometry

The spherical Fourier transform of a function $f \in L^2(S^2)$ is the set of coefficients of the expansion of this function in terms of spherical harmonics Y_ℓ^m, i.e. the coefficients $\widehat{f}(\ell, m)$ of the expression

$$f(\theta, \varphi) = \sum_{l \geq 0} \sum_{|m| \leq \ell} \widehat{f}(\ell, m) Y_\ell^m(\theta, \varphi), \tag{5.7}$$

where the function $f(\theta, \varphi)$ and the spherical harmonics $Y_\ell^m(\theta, \varphi)$ are expressed in spherical coordinates for the unit sphere ($0 \leq \theta \leq \pi$, $0 < \varphi \leq 2\pi$, $\rho = 1$, see Figure 5.3(a)). The spherical harmonics can be factorized as

$$Y_\ell^m(\theta, \varphi) = k_{\ell,m} P_\ell^m(\cos\theta) e^{im\varphi}, \tag{5.8}$$

where P_ℓ^m is an associated Legendre polynomial and $k_{\ell,m}$ is a normalisation constant that is

$$k_{\ell,m} = \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} \tag{5.9}$$

in case of orthonormal spherical harmonics (see [Barut 1986] for further details).

Then, the spherical Fourier transform of a function $f \in L^2(S^2)$ is calculated as the projection of this function on the orthonormal basis of the spherical harmonics

$$\begin{aligned} \widehat{f}(\ell, m) &= \langle f, Y_\ell^m \rangle = & (5.10) \\ &= k_{\ell,m} \int_{S^2} d\mu(\omega) \overline{Y_\ell^m(\omega)} f(\omega), & \\ &= k_{\ell,m} \int_0^{2\pi} \int_0^{\pi} \overline{Y_\ell^m(\theta, \varphi)} f(\theta, \varphi) \sin\theta d\theta d\varphi & (5.11) \end{aligned}$$

where $d\mu(\omega) = \sin\theta d\theta d\varphi$ is $SO(3)$ invariant measure on S^2. Using Equations (5.8) and (5.10), it is easy to see that the spherical Fourier transform is a regular Fourier transform in φ followed by a projection on the associated Legendre polynomial.

5.3.3 Spherical DoG as a scale-space

At this point, the only missing element to build the spherical scale-space representation of a spherical image is the function that plays the role of the Gaussian kernel in the planar case. Let us note that we need to pass through the spherical Fourier domain because convolution on the sphere in spatial domain (3D) is hard (almost impossible) to compute. See Equations (5.5) and (5.6) in Section 5.3.2 for details. This is not the case of the scale-space representation of a planar image, given the separability of the Gaussian filter and the simplicity of the planar (2D) convolution.

In [Bulow 2004], the author derives this function as a Green function of the heat equation (Equation (5.2)) over S^2, obtaining

$$g^{S^2}(\theta,\varphi,\sigma) = \sum_{\ell\in\mathbb{N}} \sqrt{\frac{2\ell+1}{4\pi}} Y_\ell^0(\theta,\varphi) e^{\frac{-\ell(\ell+1)\sigma^2}{2}}, \tag{5.12}$$

$$\widehat{g^{S^2}}(\ell,m,\sigma) = \sqrt{\frac{2\ell+1}{4\pi}} e^{\frac{-\ell(\ell+1)\sigma^2}{2}}, \tag{5.13}$$

where g^{S^2} denotes the spherical Gaussian function. Therefore, using Equation (5.6), the spherical Fourier transform of the scale-space representation of an omnidirectional image mapped on the sphere, $I(\theta,\varphi)$, is

$$\widehat{L}^{S^2}(\ell,m,\sigma) = \widehat{I}(\ell,m) e^{\frac{-\ell(\ell+1)\sigma^2}{2}} \tag{5.14}$$

for the set of considered scales (different values of σ), and its inverse spherical Fourier transform,

$$L^{S^2}(\theta,\varphi,\sigma) = I(\theta,\varphi) * g^{S^2}(\theta,\varphi,\sigma), \tag{5.15}$$

is the spherical scale-space representation of this image. Finally, the spherical DoG is computed as

$$\psi^{S^2}(\theta,\varphi,\sigma) = L^{S^2}(\theta,\varphi,k\sigma) - L^{S^2}(\theta,\varphi,\sigma). \tag{5.16}$$

Using these expressions, the algorithm for the extraction of interest points will be presented in the next section.

5.4 SIFT on the sphere

Let us define the SIFT algorithm in spherical coordinates. In this algorithm, the extraction of interest points and the local descriptor calculations are performed on the surface of the unit sphere. Here, we propose two types of descriptors: Local Spherical Descriptors (LSD) and Local Planar Descriptors (LPD). The first one is computed directly on the sphere and is intended to be matched with LSD of points extracted from different omnidirectional images. The second one is generated using a local planar approximation of the region around the extracted interest point, and can be matched with regular SIFT descriptors of points extracted from planar images. For the matching procedure we follow the method proposed in [Lowe 2004]. It consists of pairing the nearest points in terms of the distance between their descriptors, if and only if the ratio between this distance and the second smallest distance is lower than a fixed threshold $d \in [0, 1]$.

The workflow of the spherical SIFT algorithm is summarised in Algorithm 7. Each one of the steps is described in details in the following sections. Throughout this paper, a spherical image is considered defined in a (θ, φ)-grid where columns are points of constant longitude, $\varphi \in [0, 2\pi)$, and rows are points of constant latitude, $\theta \in [0, \pi]$.

Algorithm 7 Spherical SIFT algorithm

```
$I(\theta, \varphi) \longleftarrow$ omnidirectional input image mapped on $S^2$
Compute spherical scale-space representation of $I(\theta, \varphi)$
Compute spherical DoG
$E \longleftarrow$ Local extrema of spherical DoG
for each $E_i \in E$ do
    Compute LSD and/or LPD of $E_i$
```

5.4.1 Spherical scale-space and Difference-of-Gaussians

The spherical scale-space representation of a spherical image $I(\theta, \varphi)$ (ρ is fixed to 1) is computed using Equation (5.15) iteratively, i.e.

$$L^{S^2}(\theta, \varphi, \sigma_i) = L^{S^2}(\theta, \varphi, \sigma_{i-1}) * g^{S^2}(\theta, \varphi, \tilde{k}_i \sigma_0), \tag{5.17}$$

where σ_0 is the initial scale and $\tilde{k}_i$ is chosen in such a way that two neighbouring scales in the spherical scale-space representation are separated by a constant multiplicative factor $k = 2^{1/S}$ (in order to have a constant number S of images per octave). Therefore, $\sigma_i = k\sigma_{i-1} = k^i \sigma_0$ and using the semi-group property of the spherical scale-space representation, we have that $(k^i \sigma_0)^2 + (\tilde{k}_i \sigma_0)^2 = (k^{i+1}\sigma)^2$, and so $\tilde{k}_i = k^i \sqrt{k^2 - 1}$. These expressions are also valid in the planar case.

The spherical scale-space representation process is speeded up by downsampling the image by two, instead of increasing the scale σ, each time a complete octave of $\psi^{S^2}(\theta, \varphi, \sigma)$ is obtained. This is the common practice in the planar case too, but in the spherical case there is a peculiarity. In order to obtain $L^{S^2}(\theta, \varphi, \sigma)$, a spherical Fourier transform is computed and, therefore, aliasing has to be taken into account. This process of downsampling by 2 the images is especially sensitive to aliasing, since the bandwidth of the spherical Fourier transform is also divided by 2. For this reason, after the computation of each octave, the following condition is tested:

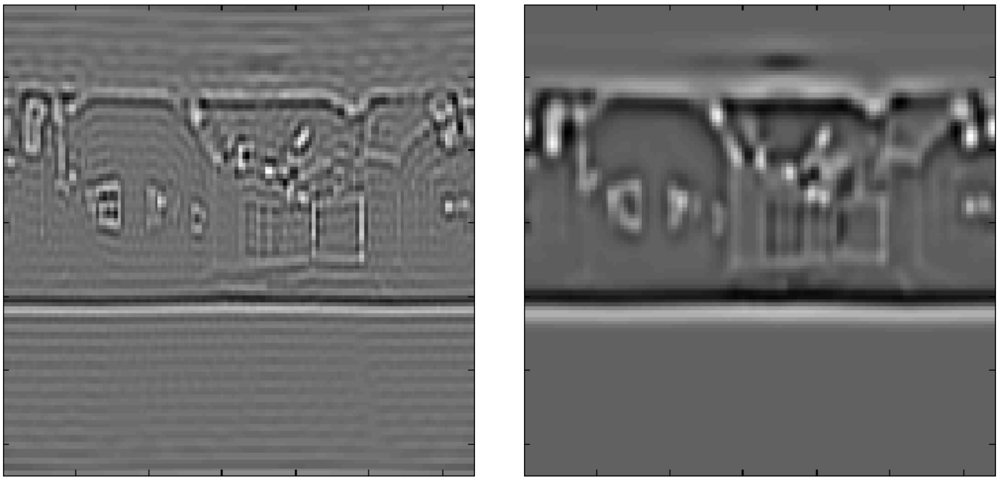
$$e^{\frac{-nH(nH+1)(\sigma_0/k)^2}{8}} \leq e^{-1}, \tag{5.18}$$

where nH is the new height of the image after reducing its size. This condition assures that the exponential part of Equation (5.13) remains small for the biggest value of ℓ. If Equation (5.18) is not fulfilled, instead of reducing the image size for the next octave, σ is increased and the image size is reduced after the convolution. Aliasing effects can still appear if they are present in the first computed spherical Fourier transform, or if σ increases considerably (Equation (5.18) not fulfilled even for the current H before downsampling). An example of the effect of applying this anti-aliasing criterion before downsampling an intermediate image in the computation of ψ^{S^2}, is shown in Figure 5.4.

The input images are supposed to have a nominal standard deviation σ_N of half pixel, which in our case means $\sigma_N = 0.5\pi/H$, where H is the height of the spherical image. To obtain the first image of the spherical scale space, $L^{S^2}(\theta, \varphi, \sigma_0/k)$, the input image is convolved with a spherical Gaussian filter with standard deviation $\sigma = \sqrt{(\sigma_0/k)^2 - \sigma_N^2}$. The computation of ψ^{S^2} is shown in Algorithm 8. Note that the size of the input image can be doubled before starting the process. Then, $\sigma_N = \pi/H$ and the first loop starts at $o = -1$.

Algorithm 8 Spherical scale-space and Difference-of-Gaussians computation

```
$S \longleftarrow$ number of stages per octave
$O \longleftarrow$ number of octaves
$n \longleftarrow 0$
for $o = 0$ to $O$ do
    Compute $L^{S^2}(\theta, \varphi, 2^o\sigma_0/k)$
    for $s = 0$ to $S + 1$ do
        Compute $L^{S^2}(\theta, \varphi, 2^o k^s \sigma_0)$
        Compute $\psi^{S^2}(\theta, \varphi, 2^o k^{s-1} \sigma_0)$
    if Equation (5.18) is satisfied then
        Down-sample by 2 the starting image of the current loop and use it for the next one
    else
        $n \longleftarrow n + 1$
        Double the $\sigma$'s of the current loop and use them in the next loop
        Each $L^{S^2}(\theta, \varphi, \sigma)$ in the next loop has to be down-sampled by $2^n$
```



(a) $\psi^{S^2}(\theta, \varphi, 2^3 k\sigma_0)$ (third stage of the fourth octave) downsampling the image without applying the anti-aliasing criterion

(b) $\psi^{S^2}(\theta, \varphi, 2^3 k\sigma_0)$ (third stage of the fourth octave) downsampling the image if the anti-aliasing criterion is fulfilled

Figure 5.4: Example of the effect of the anti-aliasing strategy for the ψ^{S^2} computation of the image in Figure 5.2(c). The image size is 1024×1024 and the spherical scale-space was generated using $\sigma_0 = 1.6\pi/1024$, $\sigma_N = 0.5\pi/1024$ and $S = 3$.

5.4.2 Extrema extraction

Interest points are local extrema of $\psi^{S^2}(\theta, \varphi, \sigma)$ (Equation (5.16)). A local extreme is a point on the spherical grid whose value is bigger (smaller) than its 8 neighbours, bigger (smaller) than its 9 neighbours in the scale above and bigger (smaller) than its 9 neighbours in the scale below. Note that, contrary to a planar image, an image on the sphere has no borders and then, points located at the last column (highest values of φ) are neighbours with points located at the first column (lowest values of φ) and vice versa. These simple comparisons give the extrema candidates, but principal curvature and contrast conditions are imposed on these points afterwards, in order to keep only the most stable ones.

For each detected local extreme of $\psi^{S^2}(\theta, \varphi, \sigma)$, $\omega_i \equiv (\theta_i, \varphi_i, \sigma_i)$, a quadratic function is fitted by using a Taylor expansion of Equation (5.16):

$$\begin{aligned} \psi^{S^2}(\theta, \varphi, \sigma) \;\; &\simeq \;\; \psi^{S^2}(\theta_i, \varphi_i, \sigma_i) + \\ &+ \left.\frac{\partial \psi^{S^2}}{\partial \Theta}\right|_{\omega_i}^{\top} \delta_{\omega_i} + \frac{1}{2}\delta_{\omega_i}^{\top} \left.\frac{\partial^2 \psi^{S^2}}{\partial \Theta^2}\right|_{\omega_i} \delta_{\omega_i}, \end{aligned} \tag{5.19}$$

where $\Theta \equiv (\theta, \varphi, \sigma)$ and $\delta_{\omega_i} = (\theta - \theta_i, \varphi - \varphi_i, \sigma - \sigma_i)^{\top}$. The derivatives are calculated as the central finite differences approximation of the derivatives of the image in that point, i.e. for a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ the central finite difference approximation of the derivative with respect to the jth variable, x^j is

$$\begin{aligned} \frac{\partial f}{\partial x^j} &= f_{x^j}(x^1, x^2, \dots, x^j, \dots, x^n) = \\ &= \frac{f(\dots, x^j + \Delta_{x^j}, \dots) - f(\dots, x^j - \Delta_{x^j}, \dots)}{2\Delta_{x^j}}. \end{aligned} \tag{5.20}$$

From now on, the notation f_x will be used to express the derivative (or the finite differences approximation) of f with respect to x.

Taking the derivative of Equation (5.19) with respect to δ_{ω_i}, the offset $\tilde{\delta}_{\omega_i}$ to the extreme of the fitted function is obtained

$$\tilde{\delta}_{\omega_i} = -\left(\frac{\partial^2 \psi^{S^2}}{\partial \Theta^2}\right)^{-1} \frac{\partial \psi^{S^2}}{\partial \Theta}. \tag{5.21}$$

If any of the components of vector $\tilde{\delta}_{\omega_i}$ is bigger than half the separation between two points in this dimension, the point ω_i is moved to its neighbour in this dimension and the process repeated. To avoid loops, if after moving the point ω_i 5 times a local extreme is still not found, the process is stopped. The movement in the σ dimension has not been taken into account, which is a common strategy in implementations of SIFT. This is due to the fact that a displacement in the σ direction changes completely the conditions where the quadratic function is fitted. At the end of this iterative process, a point $\tilde{\omega}_i$ is obtained.

Once $\tilde{\omega}_i$ has been obtained, the contrast at this point is computed, and if the condition

$$|\psi^{S^2}(\tilde{\omega}_i)| > \frac{0.02}{k^s 2^o} \tag{5.22}$$

is not satisfied, then $\tilde{\omega}_i$ is discarded. This condition is similar to the condition used in some implementations of SIFT. The objective is to be more strict when accepting a point at small scales, which are the points more affected by noise. At higher scales, the contrast constraint is relaxed.

Finally, the ratio r of principal curvatures is obtained at $\tilde{\omega}_i$ and the point is kept if and only if $r < 10$ (same value than in [Lowe 2004]). Principal curvatures of a surface at a given point p are the maximum and minimum curvatures of the resulting curves when intersecting the surface with all the planes containing the normal vector to the surface at p. This test eliminates points situated along edges, where one principal curvature is high but the other is low, which produces unstable points. In other words, if a point does not satisfy the following condition

$$\frac{\operatorname{trace}(H^{S^2})^2}{\det(H^{S^2})} < \frac{(r+1)^2}{r}, \tag{5.23}$$

where

$$H^{S^2} = \begin{pmatrix} \psi^{S^2}_{\theta\theta} & \psi^{S^2}_{\theta\varphi} \\ \psi^{S^2}_{\theta\varphi} & \psi^{S^2}_{\varphi\varphi} \end{pmatrix}, \tag{5.24}$$

then it is discarded. The condition expressed in Equation (5.23) can be easily obtained by applying that the trace of a matrix is equal to the sum of its eigenvalues and the determinant is equal to the product of eigenvalues. Note also that as we are working on the unit sphere in spherical coordinates, derivatives with respect to φ have a $1/\sin\theta$ coefficient. The full extrema extraction procedure is detailed in Algorithm 9.

Algorithm 9 Algorithm for the extraction of "good" local extrema

```
$E \longleftarrow \emptyset$ the set of local extrema
for $o = 0$ (or $o = -1$) to $O$ do
  for $s = 0$ to $S - 1$ do
    for each point $\omega_i \equiv (\theta_i, \varphi_i, \sigma_i)$ of $\psi^{S^2}(\theta, \varphi, 2^o k^s \sigma_0)$ do
      if $\omega_i$ is a local extreme then
        Compute $\tilde{\omega}_i$
        if Equations (5.22) and (5.23) are satisfied at $\tilde{\omega}_i$ then
          $E \longleftarrow \{E, \tilde{\omega}_i\}$
```

5.4.3 Local Spherical Descriptor (LSD)

In order to match points extracted from different omnidirectional images and obtained with the proposed algorithm, a Local Spherical Descriptor (LSD) is computed at each point. This descriptor is obtained using the spherical scale-space representation of the image (see Sections 5.3 and 5.4.1) and consists of a set of histograms of orientations in a region around the given point. The size of this region depends on the scale (σ) at which the point has been detected. Orientations are computed with respect to a principal orientation of the point, which makes the descriptor invariant to rotations around the axis that links the point with the centre of the sphere. The complete procedure is detailed below.

First, the orientation of a point in the spherical scale space representation has to be defined. Let us have a point $(\theta, \varphi) \in S^2$ at scale σ. Its orientation is defined as the angle of the gradient of L^{S^2} in that point, with the 0 degrees pointing to the south pole and the 90 degrees to bigger values of φ. These gradients are obtained using the central finite differences approximation of the derivatives as

$$\alpha(\theta, \varphi, \sigma) = \arctan\left(\frac{L_{\varphi}^{S^2}(\theta, \varphi, \sigma)}{L_{\theta}^{S^2}(\theta, \varphi, \sigma)}\right). \tag{5.25}$$

Then, for each considered extreme of the ψ^{S^2}, Equation (5.25) is used to compute the orientations of surrounding points on the spherical grid in a $3\sigma \times 3\sigma$ squared window centred at the extreme (where σ is the scale at which each extreme was located). To define this window, the distance between two points on the unit sphere,

$p_1 \equiv (\theta_1, \varphi_1)$ and $p_2 \equiv (\theta_2, \varphi_2)$, needs to be calculated. It can be obtained using the Vincenty's formula [Vincenty 1975]:

$$d(p_1, p_2) = \arctan\left(\frac{\sqrt{A^2 + B^2}}{C}\right), \tag{5.26}$$

where

$$\begin{aligned} A &= \sin\theta_1 \sin\Delta\varphi, && (5.27)\\ B &= \sin\theta_2 \cos\theta_1 - \cos\theta_2 \sin\theta_1 \cos\Delta\varphi, && (5.28)\\ C &= \cos\theta_2 \cos\theta_1 + \sin\theta_2 \sin\theta_1 \cos\Delta\varphi, && (5.29)\\ \Delta\varphi &= \varphi_1 - \varphi_2. && (5.30) \end{aligned}$$

For each window, a histogram of orientations is computed using the orientations of points of the spherical grid that are inside. The orientation value at each point defines the bin, and the value added to this corresponding bin is the norm of the gradient at that point,

$$m(\theta, \varphi, \sigma) = \sqrt{L_\varphi^{S^2}(\theta, \varphi, \sigma)^2 + L_\theta^{S^2}(\theta, \varphi, \sigma)^2}, \tag{5.31}$$

weighted by a Gaussian centred on the extreme and of standard deviation 1.5σ. For this histogram, 36 orientations are considered. Finally, once the histogram has been computed, the principal orientation is calculated as the axis of a parabola fitted around its maximum. If there are bins greater than 0.8 times the biggest one, they are also considered. This results in multiple principal orientations for the same point.

Then, LSD are computed taking their corresponding principal orientations as reference. This descriptor is a three-dimensional histogram of orientations (two spatial dimensions and one dimension for orientations) where all the orientations are considered with respect to the principal one. The produced histogram has $4^2 \times 8$ bins (4^2 bins for the spatial dimension and 8 bins for the orientations) and is computed considering the points of the spherical grid contained in a $6\sigma \times 6\sigma$ squared window centred at the extreme and rotated according to the principal orientation. Each bin value corresponds to the weighted sum of gradient magnitudes of points at the spatial and orientation defined by the bin. The weight value is defined by a Gaussian centred on the extreme and of standard deviation 1.5σ. The rotation of the window on the surface of the sphere can be computed using the Rodrigues' rotation formula [Rodrigues 1840] for the rotation of vectors, given by

$$v^{\text{Rot}} = v\cos\alpha + u \times v \sin\alpha + u \cdot v(1 - \cos\alpha)u, \tag{5.32}$$

where the vectors u, v and v^{Rot} are considered in Cartesian coordinates, and the vector v^{Rot} is the result of rotating α degrees the vector v around u.

In order to avoid boundary effects, the values of each gradient sample are distributed by trilinear interpolation into adjacent histogram bins. The resulting histogram is normalised, each bin thresholded to 0.2 and normalised again, in order to make it robust to contrast changes. The algorithm for computing Local Spherical Descriptors is summarised in Algorithm 10.

Algorithm 10 Algorithm for the computation of LSD

```
LSD $\longleftarrow \emptyset$ the set of local spherical descriptors
**for** each considered extreme of $\psi^{S^2}$, $(\theta_i, \varphi_i, \sigma_i)$ **do**
    Select a squared region of size $3\sigma_i \times 3\sigma_i$ centred at $(\theta_i, \varphi_i)$
    Compute orientations and gradient norms inside this region
    Compute histogram of orientations
    MAX $\longleftarrow$ maximum histogram value
    **for** each bin value $\geq 0.8$MAX **do**
        Fit a parabola around this bin
        $b \longleftarrow$ axis of the parabola
        Select a squared region of size $6\sigma_i \times 6\sigma_i$ centred at $(\theta_i, \varphi_i)$ and rotated $b$ degrees
        Compute orientations and gradient norms inside this region with respect to $b$
        LSD$_i \longleftarrow$ Compute 3-dimensional histogram
        LSD $\longleftarrow \{\text{LSD}, \text{LSD}_i\}$
```

5.4.4 Local Planar Descriptor (LPD)

Local Planar Descriptors (LPD) allow to match points extracted from a spherical image, using Algorithms 8 and 9, and SIFT descriptors of points extracted from planar images. This is of great importance, considering that a preexisting database of SIFT descriptors computed on planar images could be used to detect objects on the omnidirectional image.

The LPD is a regular SIFT descriptor computed on a planar approximation of the region around each interest point $\omega_i \equiv (\theta_i, \varphi_i, \sigma_i)$. We consider $p_i \equiv (\theta_i, \varphi_i)$ to be the centre of this planar approximation, which is the stereographic projection on the tangent plane of the sphere at p_i through its antipodal point. This projection of $L^{S^2}(\theta, \varphi, \sigma_i)$ around p_i can be seen as a local approximation of $L(x, y, \sigma)$. In other words, for a point $p_i \equiv (\theta_i, \varphi_i)$, extracted from the spherical image at scale σ_i, a squared window centred at ω_i on $L^{S^2}(\theta, \varphi, \sigma_i)$ and of size equal to the minimum between $12\sigma_i$ and π, is stereographically projected from $(\theta_i + \pi/2, \varphi + \pi)$ to the plane tangent at p_i. The projected points are linearly interpolated in order to obtain a planar image whose cartesian range is $[-2\tan\frac{6\sigma_i}{2}, 2\tan\frac{6\sigma_i}{2}] \times [-2\tan\frac{6\sigma_i}{2}, 2\tan\frac{6\sigma_i}{2}]$ and with a pixel spacing of $2\tan\frac{\pi}{2H}$. H is the height of $L^{S^2}(\theta, \varphi, \sigma_i)$. The equivalent

σ_i in the obtained planar image is given below:

$$\sigma_i^{\text{pl}} = \frac{\tan\frac{\sigma_i}{2}}{\tan\frac{\pi}{2H}}. \tag{5.33}$$

The outline of the Local Planar Descriptors computation is given in Algorithm 11.

Algorithm 11 Algorithm for the computation of LPD

```
LPD $\longleftarrow \emptyset$ the set of local planar descriptors
for each considered extreme of $\psi^{S^2}$, $(\theta_i, \varphi_i, \sigma_i)$ do
    $L(x, y, \sigma_i^{\text{pl}}) \longleftarrow$ stereographic projection of $L(\theta, \varphi, \sigma_i)$ from $(\theta_i + \frac{\pi}{2}, \varphi_i + \pi)$ to the tangent plane at $(\theta_i, \varphi_i)$
    $\text{LPD}_i \longleftarrow$ SIFT descriptor of $L(x, y, \sigma_i^{\text{pl}})$ at $(x, y) = (0, 0)$
    LPD $\longleftarrow \{\text{LPD}, \text{LPD}_i\}$
```

5.5 Planar to spherical mapping

As mentioned before, LPD can be matched with regular planar SIFT descriptors extracted from planar images. In addition to this new kind of matching, we propose a method to estimate the function that transfers points from an object in a planar image to their corresponding points in a spherical image. We suppose that the object is rigid and planar, because only its projection on a planar image is known. The transfer function, together with the planar to spherical matching, can segment objects in omnidirectional images given their segmentation in a planar image or vice-versa.

Let us consider two matched points, the first $p_i^{\text{pl}} \equiv (x_{2i}^{\text{pl}}, x_{1i}^{\text{pl}})$ in a planar image and the second $p_j^{S^2} \equiv (x_{2j}^{S^2}, x_{1j}^{S^2}, x_{0j}^{S^2})$ in a spherical image, both in cartesian coordinates. The idea is to find a linear transformation H that sends the point in the planar image p_i^{pl}, to a point q_{ij} in three-dimensional space, with projection to the unit sphere $p_j^{S^2}$ (see Figure 5.5). The linearity of H is given by the rigidity assumption. Let us note that only the transformation from p_i^{pl} to q_{ij} is linear, and this is not true for the total mapping from p_i^{pl} to $p_j^{S^2}$. In other words, we look for a 3×3 matrix H that satisfies

$$p_j^{S^2} = \frac{q_{ij}}{||q_{ij}||} = \frac{H\tilde{p}_i^{\text{pl}}}{||H\tilde{p}_i^{\text{pl}}||}, \tag{5.34}$$

where $||\cdot||$ denotes the 2-norm and $\tilde{p}^{\text{pl}_i}$ is an embedding of p_i^{pl} in $\mathbb{R}^3$ (more details follow). For estimating H, the planar image is placed tangentially to the sphere where the omnidirectional image is mapped. The central point of the planar image is the contact point with the sphere. In this way, a point $p_i^{\text{pl}} \equiv (x_{2i}^{\text{pl}}, x_{1i}^{\text{pl}})$ of the planar image is embedded in $\mathbb{R}^3$ as $\tilde{p}_i^{\text{pl}} \equiv (x_{2i}^{\text{pl}}, x_{1i}^{\text{pl}}, 1)$. Then, the fact that $p_j^{S^2}$ and $H\tilde{p}_i^{\text{pl}}$ must be collinear is exploited forcing their vectorial product to be zero, i.e.

$p_j^{S^2} \times H\tilde{p}_i^{\mathrm{pl}} = 0$. The latter condition generates three equations, one for each of the components of the resulting vector of the cross product:

$$\begin{aligned}
- &\; x_{0j}^{S^2} x_{2i}^{\mathrm{pl}} h_{21} - x_{0j}^{S^2} x_{1i}^{\mathrm{pl}} h_{22} - x_{0j}^{S^2} h_{23} \\
+ &\; x_{1j}^{S^2} x_{2i}^{\mathrm{pl}} h_{31} + x_{1j}^{S^2} x_{1i}^{\mathrm{pl}} h_{32} + x_{1j}^{S^2} h_{33} &= 0, \qquad (5.35)\\
&\; x_{0j}^{S^2} x_{2i}^{\mathrm{pl}} h_{11} + x_{0j}^{S^2} x_{1i}^{\mathrm{pl}} h_{12} + x_{0j}^{S^2} h_{13} \\
- &\; x_{2j}^{S^2} x_{2i}^{\mathrm{pl}} h_{31} - x_{2j}^{S^2} x_{1i}^{\mathrm{pl}} h_{32} - x_{2j}^{S^2} h_{33} &= 0, \qquad (5.36)\\
- &\; x_{1j}^{S^2} x_{2i}^{\mathrm{pl}} h_{11} - x_{1j}^{S^2} x_{1i}^{\mathrm{pl}} h_{12} - x_{1j}^{S^2} h_{13} \\
+ &\; x_{2j}^{S^2} x_{2i}^{\mathrm{pl}} h_{21} + x_{2j}^{S^2} x_{1i}^{\mathrm{pl}} h_{22} + x_{2j}^{S^2} h_{23} &= 0, \qquad (5.37)
\end{aligned}$$

where the elements of the matrix H are distributed as

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \qquad (5.38)$$

Consequently, if Equations (5.35), (5.36) and (5.37) are expressed in terms of h_{lm} and all the resulting equations for each pair of matched points are put together, a system of equations of the form $A\mathbf{h} = \mathbf{0}$ is obtained, where

$$\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^\top,$$

A is a $3N \times 9$ matrix and N is the number of points matched between the planar and spherical images. If the restriction $\|\mathbf{h}\| = 1$ is considered, $\mathbf{h}$ can be computed as the eigenvector of matrix A corresponding to the smallest eigenvalue. This eigenvector is the least squares estimator of the solution. Note that although each pair of matched points generates three equations, only two of them are linearly independent. This means that at least four non-collinear pairs of matched points are required for estimating the coefficients of the matrix H.

The estimation of this matrix results in a mapping $h : \mathbb{R}^2 \longrightarrow S^2$ that sends points in the planar image to points in the spherical one as follows:

$$h(x, y) = \frac{H \begin{pmatrix} x_2 \\ x_1 \\ 1 \end{pmatrix}}{\left\| H \begin{pmatrix} x_2 \\ x_1 \\ 1 \end{pmatrix} \right\|}. \qquad (5.39)$$

Mapping points of the spherical image onto points of the planar image, can also be done using the transformation H^{-1} and normalising the resulting point by its third component. In this way, a point of the form $(x_2, x_1, 1)$ is obtained.

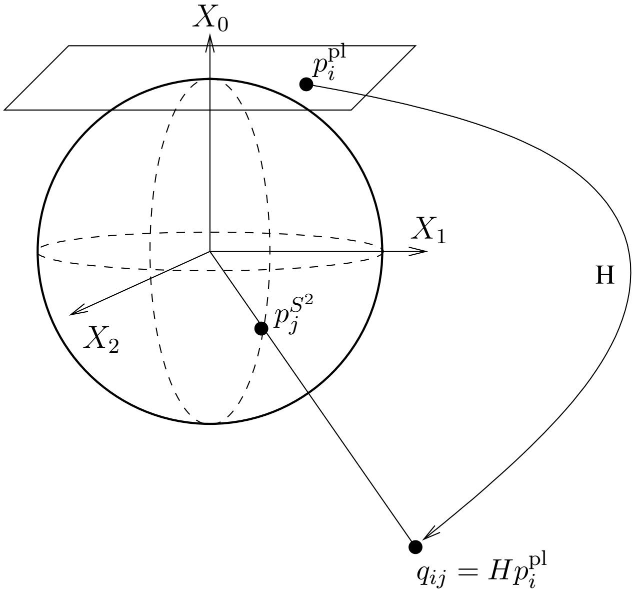


Figure 5.5: Graphical sketch of the mapping

The estimation of H using all the matched points would give bad results due to false matchings. For avoiding this, the chosen set of matched points for the estimation of H is selected using RANSAC. This procedure also softens the planar assumption. Indeed, RANSAC treats non-coplanar points as outliers, since they do not fit correctly the model generated by H. The outline of the process for computing H is specified in Algorithm 12.

Algorithm 12 Algorithm for the estimation of H

```
1: LPD ⟵ set of local planar descriptors of the spherical image
2: SIFTDesc ⟵ set of SIFT descriptors of the planar image
3: $M$ ⟵ matching points between SIFTDesc and LPD
4: $H$ ⟵ eigenvector with minimum eigenvalue of the matrix defined using Equations (5.35), (5.36) and (5.37) (use RANSAC to clean $M$ of false matchings).
```

5.6 Experimental results

In our experimental results we use two types of omnidirectional images: parabolic and spherical. Parabolic omnidirectional images are obtained by a catadioptric omnidirectional sensor: a parabolic mirror Kaidan EyeSee 360 deg[4] in combination with a Nikon D40X camera. In order to apply our algorithm on this kind of images, we first need to map them on the sphere. After this mapping, the images

[4] http://www.kaidan.com

cover a band of about 100 deg on the sphere. Spherical images are obtained with a Ladybug2 device[5] and they cover 75% of the sphere. It is important to note that the Ladybug2 outputs the images directly in spherical coordinates and thus no mapping on the sphere is needed for them. Both types of images have the same resolution, 1024 × 1024.

In all tests, two interest points p_1 and p_2 that define two interest regions in two different images, I_1 and I_2 respectively, are considered as the same point if after transferring p_2 to I_1, the overlap error computed using the intersection over union criterion [Mikolajczyk 2005b] is smaller than 0.5. The size of the interest regions considered to compute the overlap error is fixed by the scale at which each point is detected (see Section 5.4.3 for details).

This section is organised as follows. In Section 5.6.1, a comparison between standard and spherical SIFT has been done. An example where standard SIFT applied on omnidirectional images fails is given. In Section 5.6.2, the optimal parameters of the algorithm for omnidirectional images are deduced from several performed tests, and then some examples are shown. Finally, in Section 5.6.3, a matching test between an object on a planar image and several omnidirectional images containing the object is performed. In addition, the estimation of the planar to spherical mapping using matchings between planar and omnidirectional images is illustrated.

5.6.1 "Planar vs Spherical" scale invariant feature transform

As commented in Section 5.1, at small scales and for points far from the poles, the standard SIFT algorithm can perform acceptably well. But it is important to be aware of this limitation, since as soon as this two hypothesis are not fulfilled, the standard SIFT fails and the extra computation cost of considering the geometry of the sensor needs to be paid. However, this extra cost provides us more precision and invariance to the deformations that the spherical geometry introduces.

We have compared the points extracted on a sequence of omnidirectional images by a standard implementation of SIFT and by the proposed algorithm. In this experiment we apply the standard SIFT as developed and implemented in [Vedaldi 2008] on a sequence of three real spherical images. These images were obtained in an office while the sensor points down a table where red objects are placed. One of these objects is moved throughout the sequence and thus it is perceived on the South Pole of the sphere, i.e. just right below the sensor. In the unwrapped version of the spherical image, in (θ, φ) coordinates, it is completely deformed (top image of Figure 5.6). Then, we apply Algorithm 7 for computing SIFT in spherical coordinates on the same images. For obtaining comparable

[5] http://www.ptgrey.com/products/ladybug2/

results, both algorithms are run with the same parameters, those proposed by Lowe in [Lowe 2004], $S = 3$ and $\sigma_0 = 1.6$. The octave -1 is not computed in any of the two cases.

After applying the standard SIFT algorithm, in Figure 5.6(a), we can observe that in the first image (bottom), the red object has a point in the centre and points in the four corners. In the second image, the points in one of the corners and in the centre have disappeared. Finally, in the third image, only one point is extracted on the object. Nevertheless, in Figure 5.6(b), where the spherical SIFT is applied, in the first image (bottom), the red object has a point in the centre and points in three of its corners. In the second image, the object still has a point in the centre as well as in the three corners. And finally, in the third image, the red object has lost the point in the centre but still has the points in the three corners. In other words, the spherical SIFT detects the object even when it has been completely deformed by the sensor, while the standard SIFT fails.

5.6.2 "Omni vs Omni" repeatability and matching

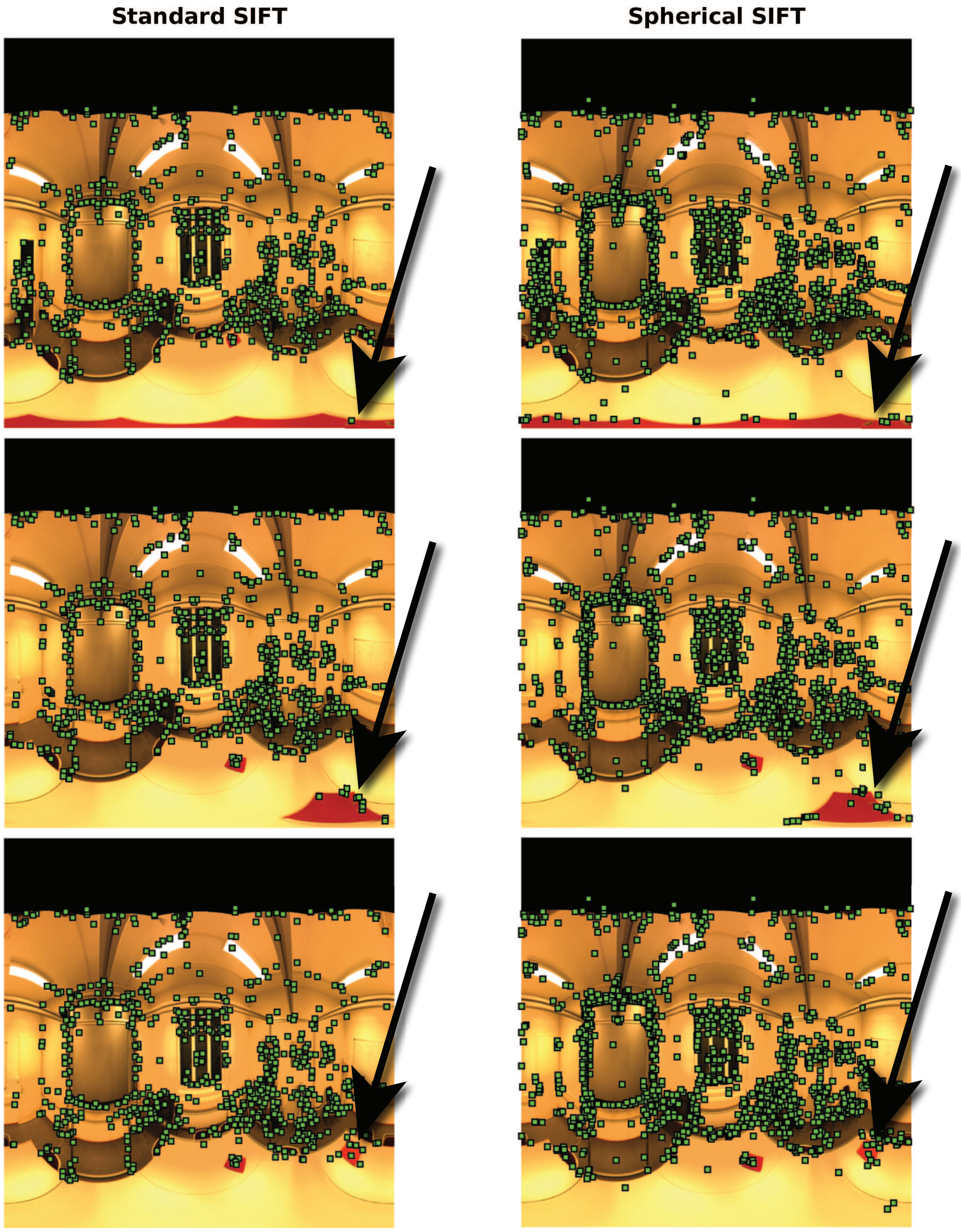
First of all, in order to test LSD matching, some parameters of the algorithm need to be fixed, essentially S and σ_0. In order to choose the values of S and σ_0 that maximise repeatability, repeatability tests have been performed on 28 real omnidirectional images. These images were taken in three different days and in two different locations, producing images under very different conditions. Some examples are shown in Figures 5.7 and 5.8.

For the purpose of testing, these images have been corrupted with zero mean additive Gaussian noise with standard deviation 0.05 (pixel values are in the range $[0, 1]$) and rotated on the sphere a random angle ψ around X_2 (see Figure 5.1). This produces hard deformations as it can be observed for instance in Figure 5.13(a)-right. Then, the repeatability score for a given pair of images, i and j, is computed as

$$r_{ij} = \frac{nR_{ij}}{\min\left(n_i, n_j\right)}, \tag{5.40}$$

where n_i and n_j are the number of extracted points from images i and j, respectively, and nR_{ij} is the number of repeated points, i.e. points defining regions with an overlap error lower than 0.5 [Mikolajczyk 2005b].

The results of repeatability tests are given in Figure 5.9. As expected, the repeatability increases for higher values of σ_0. Although a higher σ_0 also means that the extrema of the DoG at lower scales are lost. Consequently, a compromise has to be found between the smallest scale of the extrema detected and the tolerated amount of "noise" (not repeatable points) between all the extracted points. On the other hand, higher values of S imply a greater number of stages per octave,



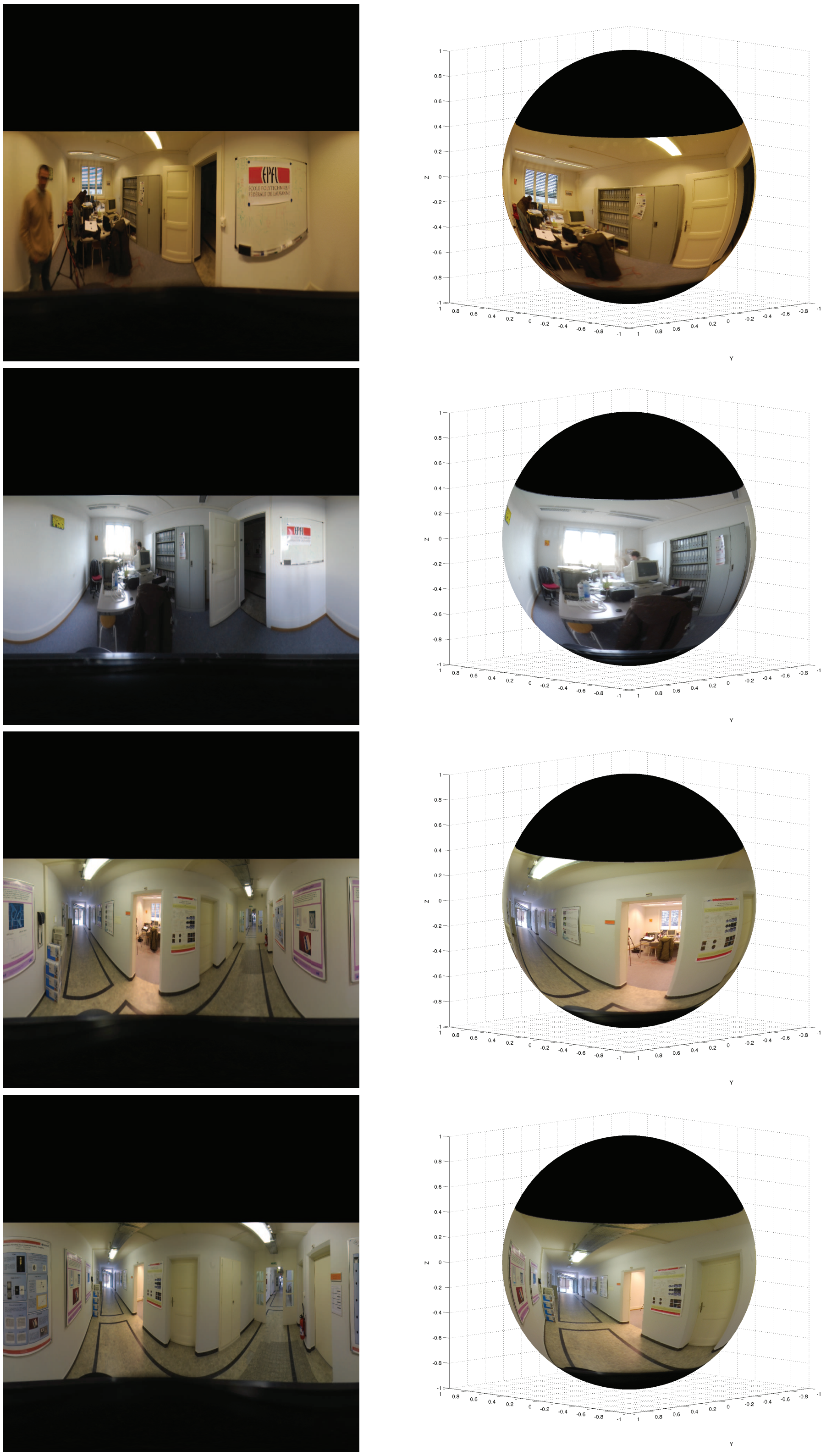
Standard SIFT

Spherical SIFT

(a) Location of the extracted points by the standard SIFT algorithm applied to a sequence of 3 omnidirectional images. From bottom to up: 808 points, 846 points and 855 points.

(b) Location of the extracted points by the proposed scale invariant feature transform on the sphere applied to a sequence of 3 omnidirectional images. From bottom to up: 1373 points, 1413 points and 1489 points.

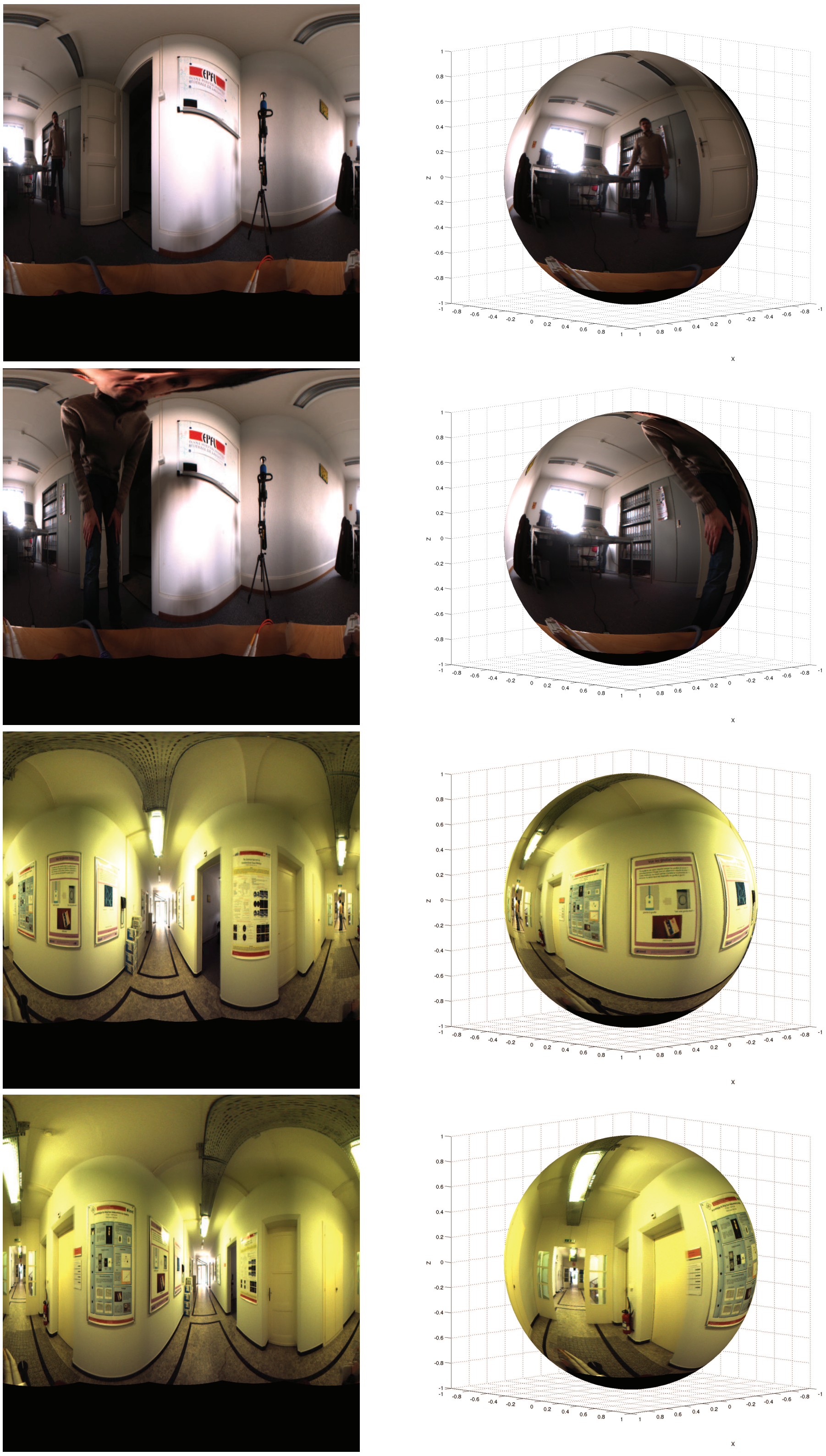
Figure 5.6: Comparison between points extracted by a standard implementation of SIFT and those extracted by the proposed scale invariant feature transform on the sphere. It can be observed how the Standard SIFT losses the extracted points of the object that goes towards the pole, while the Spherical SIFT continues to extract them without problems.



(a) Unwrapped parabolic images

(b) Parabolic images mapped on the sphere

Figure 5.7: Some of the real parabolic omnidirectional images used in our tests of repeatability and matching.



(a) Unwrapped spherical images

(b) Spherical images mapped on the sphere

Figure 5.8: Some of the real spherical omnidirectional images used in our tests of repeatability and matching.

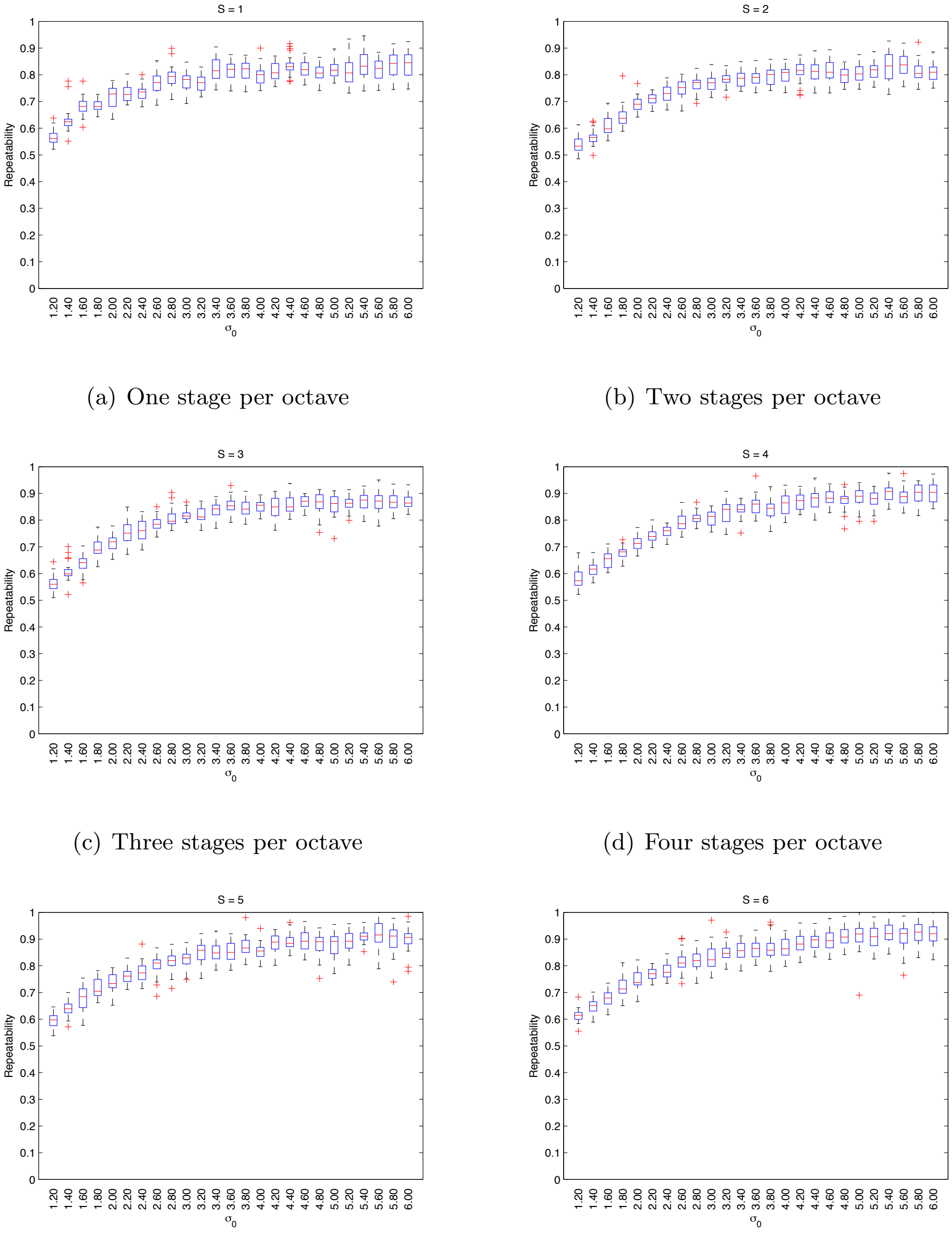
	Mean repeatability	Maximum repeatability	Minimum repeatability
Spherical SIFT	82.02%	96.39%	69.19%
Standard SIFT (with octave -1)	38.78%	93.16%	9.90%
Standard SIFT (without octave -1)	45.96%	95.44%	11.64%

Table 5.1: Repeatability values of spherical SIFT and standard SIFT.

which requires more computation time. Looking at the graphics, $\sigma_0 = 3.0\pi/1024$ and $S = 3$ are reasonable values to choose.

For comparing the performance in omnidirectional images of the scale invariant feature transform on the sphere against the standard implementation of SIFT, we have performed a repeatability test comparing both methods. Here we have corrupted omnidirectional images with zero mean additive Gaussian noise with standard deviation 0.05 (pixel values are in the range $[0, 1]$) and then, the obtained images have been rotated 50 times on the sphere a random angle ψ around X_2, producing a total of 1400 pairs of images. In Table 5.1, the mean, maximum and minimum values of repeatability for both methods are shown. As you can see, the results obtained by spherical SIFT are considerably better. Furthermore, in Figure 5.10 we have plotted repeatability values as a function of rotation angle (plotted values are obtained by grouping values in bands of 10 degrees). It can be observed that the results obtained with spherical SIFT are independent of the rotation angle and the obtained values are always better than those obtained by standard SIFT. The difference is small for rotation values of 0, 180 and 360 degrees. However, a small rotation of 10 degrees already decreases the repeatability of standard SIFT considerably. Furthermore, if only spherical images are considered (discarding the parabolic ones) i.e. we consider images that contain information on one of the poles (see Figure 5.8). Then, in this case, the repeatability values for the standard SIFT are already 6.7, 7.6 and 4.6 percentage points lower than with the spherical SIFT for 0, 180 and 360 degrees, respectively. This difference also increases considerably increasing the rotated angle.

For these chosen values, $\sigma_0 = 3.0\pi/1024$ and $S = 3$, a matching test has been performed in order to observe the effect of the threshold d (see Section 5.4). Again, real omnidirectional images have been artificially rotated on the sphere by a random angle ψ around X_2 and corrupted with zero mean additive Gaussian noise. In Figure 5.11, the results of this test can be observed. In Figure 5.11(a) the percentage of correct matchings is computed as the ratio between correct matchings and the total number of matchings. In Figure 5.11(b), the percentage of correct matchings is computed as the ratio between correct matchings and the total number of repeated points. On the one hand, for a matching threshold higher than 0.3, more than 50% of repeated points are correctly matched. On the other hand, for a matching threshold lower than 0.7 more than 80% of the matched points are correct.



(a) One stage per octave

(b) Two stages per octave

(c) Three stages per octave

(d) Four stages per octave

(e) Five stages per octave

(f) Six stages per octave

Figure 5.9: Results of the repeatability tests varying S and σ_0 over a set of 28 omnidirectional images. Note that the values of σ_0 in the graphs are in terms of relative distance between points of the spherical grid, i.e. for an image of 1024×1024 pixels, $\sigma_0 = 2.0$ in the graph means an effective $\sigma_0 = 2.0\frac{\pi}{1024}$. The boxes in the plot mark the 25th and the 75th percentile, red lines are the median and red stars are values considered as outliers (values larger than $q_{75} + 1.5(q_{75} - q_{25})$ or smaller than $q_{25} - 1.5(q_{75} - q_{25})$, where q_{25} and q_{75} are the 25th and 75th percentiles, respectively).

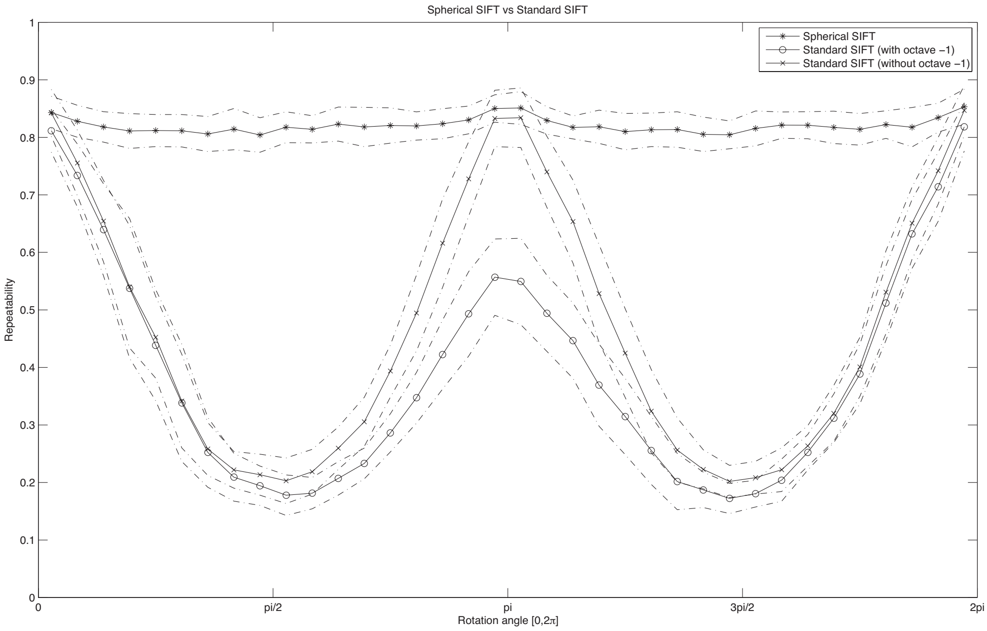


Figure 5.10: Plots of repeatability values, as a function of the randomly rotated angle on the sphere, for the spherical SIFT implementation and a standard SIFT implementation [Vedaldi 2008] with and without the octave −1. The optimal parameters for both algorithms were used. The tests were performed on 28 real omnidirectional images that were corrupted with Gaussian noise and randomly rotated 50 times, producing 1400 pairs of images. Solid lines indicate the mean value and dashed lines the mean +/− standard deviation.

In Figure 5.12, results of repeatability and correct matchings as a function of pixel noise are given. As expected, the increase of the amount of noise causes a decrease of repeatability (see Figure 5.12(a)), but observing Figures 5.12(b) and 5.12(c), we can observe the robustness of the matching, since the decrease on the percentage of correct matchings is considerably lower.

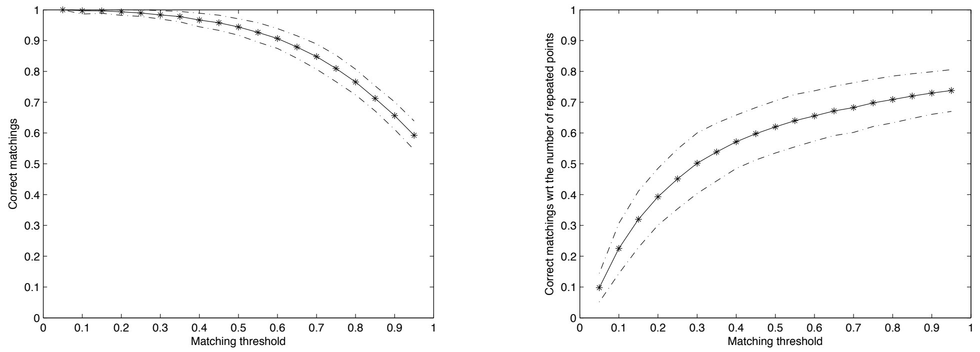
In Figures 5.13 and 5.14, some examples of LSD matching between omnidirectional images are shown. In all the computations, the above mentioned parameters ($S = 3$ and $\sigma_0 = 3.0\pi/1024$) are used, as well as a matching threshold of $d = 0.7$. In Figure 5.13, we show the matchings between a parabolic omnidirectional image and the same image rotated on the sphere and corrupted by additive Gaussian noise with zero mean and standard deviation 0.05. 184 LSD are correctly matched between the two images, over a total of 207 matched points (88.9%), with 329 extracted points on the image on the left, and 341 on the image on the right. In Figure 5.14, we show the 49 matched LSD obtained between two different parabolic images. Among the incorrect matches observed, most of them are actually locally correct, since they are the result of matching the real window with its reflection in the whiteboard, or one of the three identical markers in the whiteboard with another of them, etc.

Let us note that we do not need to use any virtual camera plane framework for performing matching, as in [Mauthner 2006]. Instead, we perform the matching directly in the spherical coordinates, in which the omnidirectional sensor outputs the images (what concerns the Ladybug2).

5.6.3 “Planar vs Omni” matching

In this section, LPD are tested for matching between points extracted from omnidirectional images (using Algorithm 7) and points extracted from a planar image (using the standard SIFT algorithm). For these experiments, the SIFT parameters proposed in [Lowe 2004] ($S = 3$ and $\sigma_0 = 1.6$) are used in both the standard SIFT algorithm and the spherical SIFT algorithm. Note that on the sphere, the equivalent σ parameter is $\sigma_0 = 1.6\pi/1024$. We do not compute octave -1 for speeding up the computation. On the other hand, the images we are working with are already at high resolution.

In a first step, a sequence of spherical images is processed for extracting LPD descriptors. The sequence is shown in Figure 5.15, in its unwrapped version and on the sphere, . This sequence consists of six images where a spherical camera moves approximately parallel to a poster on a wall. Then, a planar image of the same poster in the same scene is processed in order to extract SIFT descriptors. The result of matching both descriptors are presented in Figure 5.15. There, the



(a) Correct matchings with respect to the total number of matched points, as a function of matching threshold.

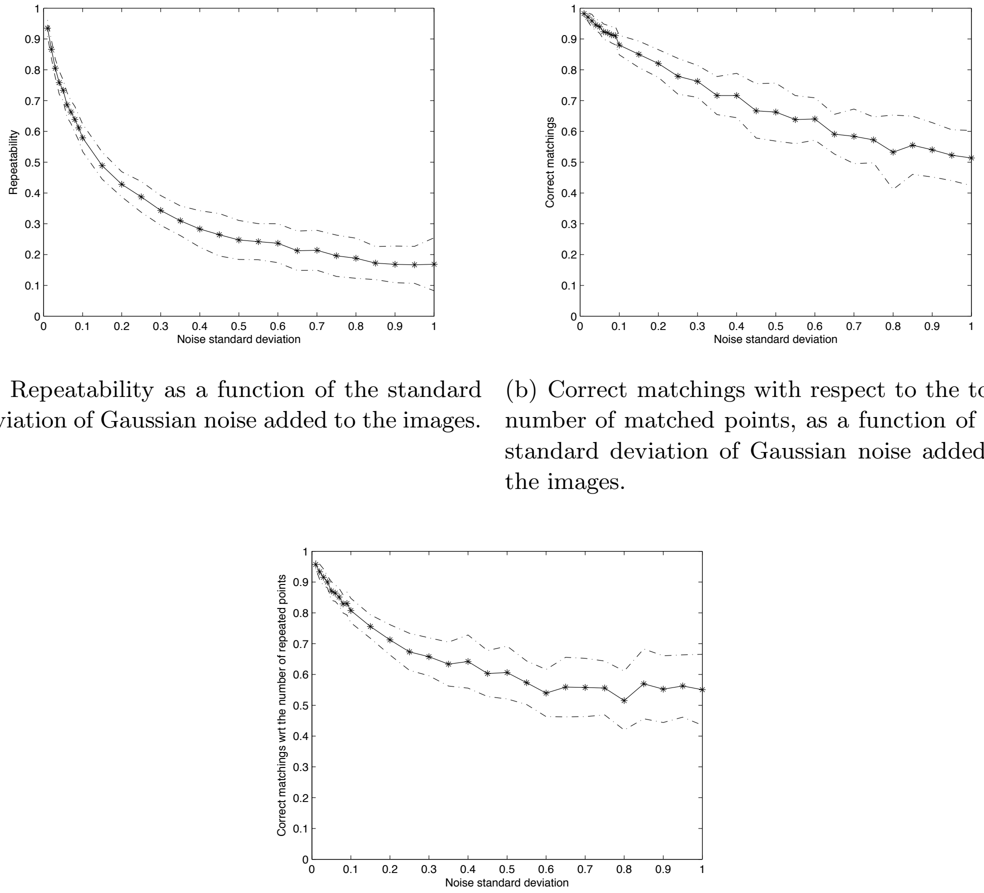
(b) Correct matchings with respect to the total number of repeated points, as a function of matching threshold.

Figure 5.11: Plots of correct matching tests performed on 28 real omnidirectional images. Correct matchings are plotted as a function of the matching threshold. Solid lines indicate the mean value and dashed lines the mean +/− standard deviation.

links between matched points are shown only on the unwrapped version so that the entire sphere is visible. As it can be observed, the proposed algorithm presents a good performance as well as a good stability of the matched points. Some of the incorrect matchings that are present are due to the fact that other posters in the corridor contain some of the images of the original poster. Using the planar to spherical mapping, we can compute the repeatability between the standard SIFT algorithm applied on the planar image and the spherical SIFT algorithm applied on the sequence of spherical images. For an overlapping error lower than 0.5, we obtain an average repeatability of 27.5%, with a maximum repeatability of 34.3% and a minimum of 21.2%. These are good results, since we obtain a good amount of repeated points even considering two completely different images (planar and spherical), with two different resolutions of the poster (in the spherical image, the poster size is 1/4 the size in the planar image) and with some artefacts in the spherical image due to the stitching performed by the Ladybug2 device.

The estimation of the planar to spherical mapping, as introduced in Section 5.5, has also been tested. First, Algorithm 7 is applied to the corresponding omnidirectional images in order to obtain the set of LPD. Then, the standard SIFT descriptors are computed for the planar images, and both descriptors LPD and SIFT are the input of Algorithm 12. Let us recall that this algorithm automatically computes the matching and the mapping.

In Figure 5.16, results obtained with images containing the EPFL logo (Figures 5.16(a) and 5.16(b)) and a poster (Figures 5.16(c) and 5.16(d)) are shown. It is interesting to note that the obtained results are satisfactory even with a highly

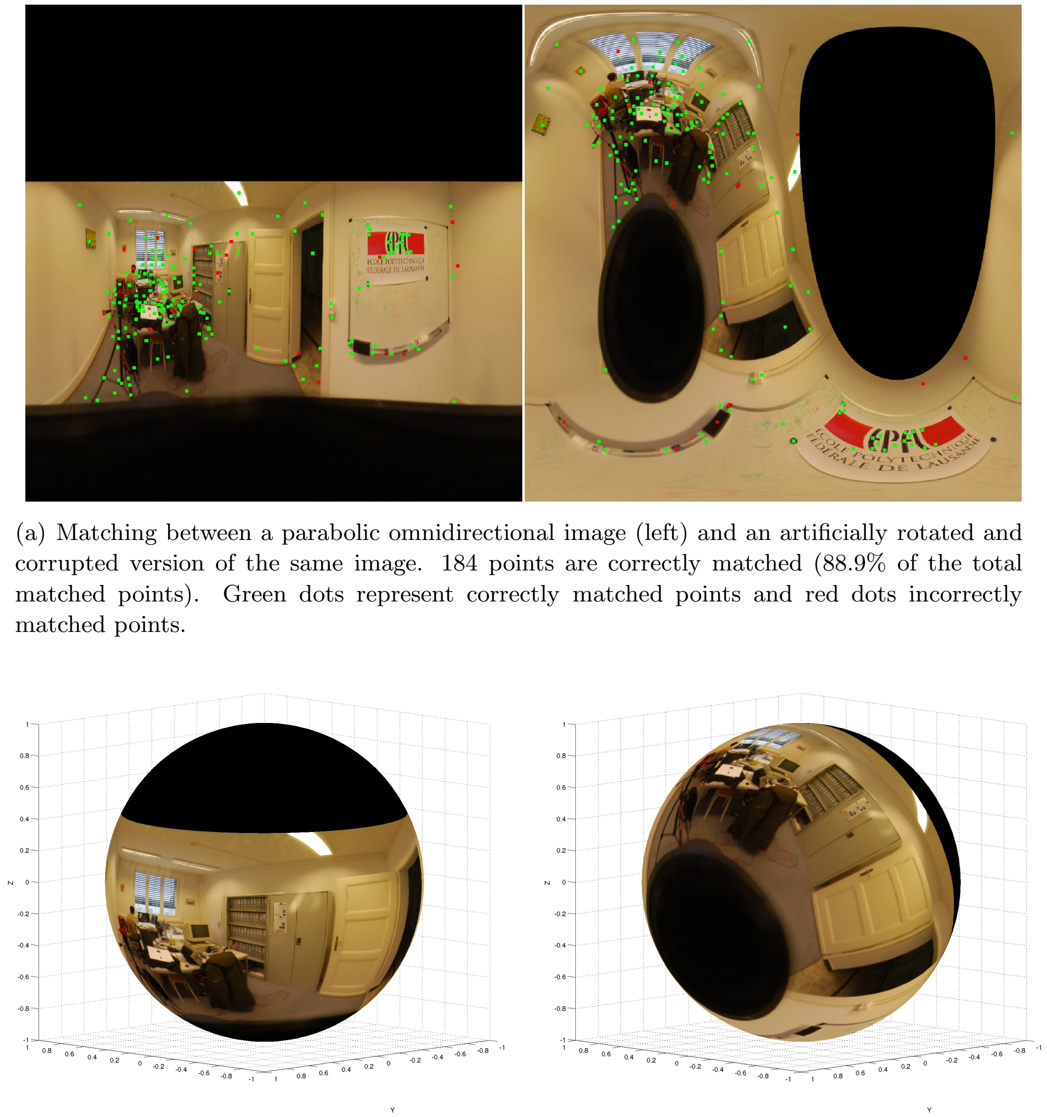


(a) Repeatability as a function of the standard deviation of Gaussian noise added to the images.

(b) Correct matchings with respect to the total number of matched points, as a function of the standard deviation of Gaussian noise added to the images.

(c) Correct matchings with respect to the total number of repeated points, as a function of the standard deviation of Gaussian noise added to the images.

Figure 5.12: Plots of repeatability and correct matching tests performed on 28 real omnidirectional images, as a function of the standard deviation of the Gaussian noise added to the images, for $\sigma_0 = 3.0\pi/1024$, $S = 3$ and a matching threshold fixed to 0.6. Solid lines indicate the mean value and dashed lines the mean +/− standard deviation.

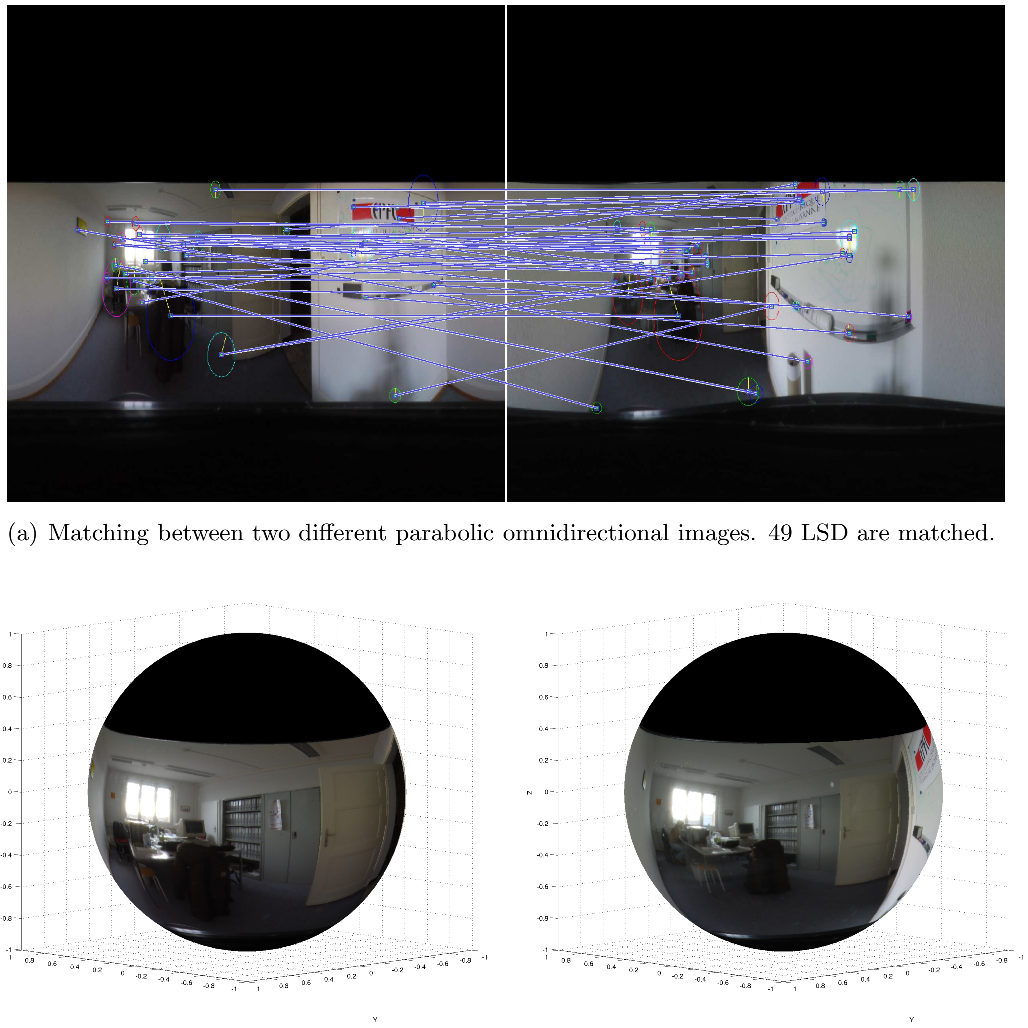


(a) Matching between a parabolic omnidirectional image (left) and an artificially rotated and corrupted version of the same image. 184 points are correctly matched (88.9% of the total matched points). Green dots represent correctly matched points and red dots incorrectly matched points.

(b) Parabolic omnidirectional image mapped on the sphere (unwrapped version in Figure5.13(a)-left)

(c) Parabolic omnidirectional image rotated 72 degrees around X_2 and mapped on the sphere (unwrapped version in Figure 5.13(a)-right)

Figure 5.13: Example of LSD matching between spherical images. The images are shown unwrapped and on the sphere. The values of the parameters obtained in the previous tests ($S = 3$ and $\sigma_0 = 3.0\pi/1024$) have been used in the computations. The matching threshold has been fixed to 0.7.



(a) Matching between two different parabolic omnidirectional images. 49 LSD are matched.

(b) Omnidirectional image in Figure 5.14(a)-left mapped on the sphere

(c) Omnidirectional image in Figure 5.14(a)-right mapped on the sphere

Figure 5.14: Example of LSD matching between spherical images. The images are shown unwrapped and on the sphere. The values of the parameters obtained in the previous tests ($S = 3$ and $\sigma_0 = 3.0\pi/1024$) have been used in the computations. The matching threshold has been fixed to 0.7.

symmetric object, as is the case of the logo, or an object with parts present in several places on the omnidirectional image, as is the case of the poster. Indeed, in these cases a matching can be locally correct while being incorrect considering the whole object or image.

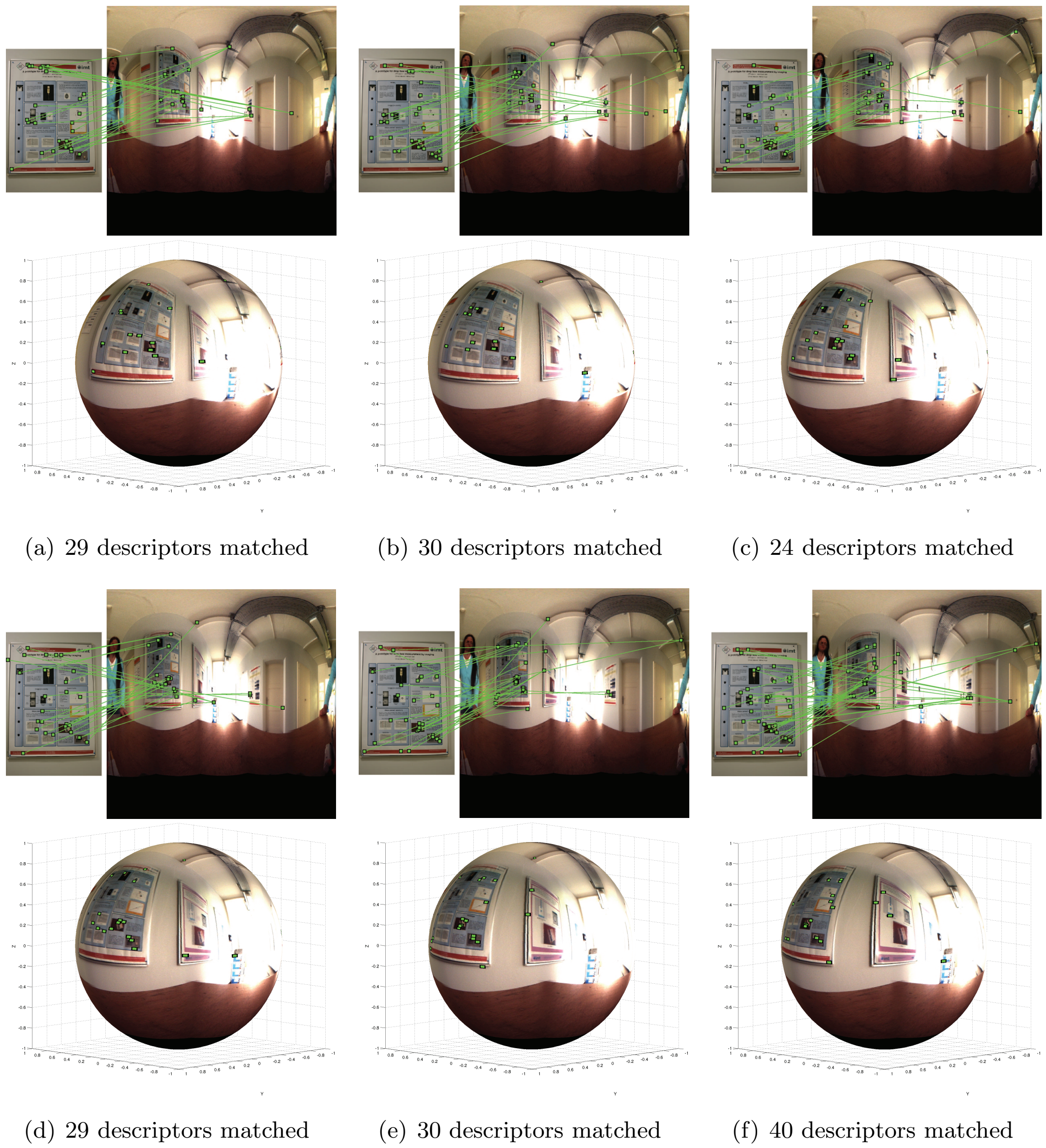
5.7 Conclusions and Future Work

After studying matching between images with a temporal correlation in Chapter 3 and Chapter 4, here we have broken the temporal relationship and have studied a general problem of matching involving omnidirectional images. This implies bigger changes between features and therefore, the robustness of the features used for matching is the crucial point of the problem.

In this Chapter, we have proposed a SIFT algorithm directly computed in spherical coordinates for omnidirectional images. It is not limited to pure spherical images, since it can also be applied to a wide variety of omnidirectional images that can be mapped on the sphere. Two types of point descriptors have been proposed: Local Spherical Descriptors (LSD) and Local Planar Descriptors (LPD). Using these descriptors, we have successfully performed point matchings between omnidirectional images, with LSD, and between omnidirectional and planar images, with LPD. For the matchings between omnidirectional images, the parameters of the algorithm have been chosen according to the results obtained on test images varying S and σ_0. For the planar vs omnidirectional case, the same parameter values as those proposed in [Lowe 2004] have been kept. Finally, point matchings obtained in this last case have been successfully used to estimate a planar to spherical mapping.

Potential applications of the proposed algorithm are global tracking in hybrid camera networks (together with the SIFT algorithm for planar images), motion estimation in omnidirectional images, object detection and extraction from omnidirectional images and, in general, any problem requiring a matching between points in omnidirectional images or between points in omnidirectional and planar images.

The main drawback of the proposed algorithm is the computation time. Indeed, for a 1024×1024 image, the complete point extraction and LSD computation takes around 33 seconds in a 3.33GHz processor. If LPD are needed, the stereographic projection for each extracted point requires around 0.1 extra seconds. Computation time depends, however, on the number of stages per octave, the σ_0 value and the number of points extracted. By optimising the code and implementing the spherical Fourier transform presented in [Tygert 2008], the computation time could be reduced.



(a) 29 descriptors matched

(b) 30 descriptors matched

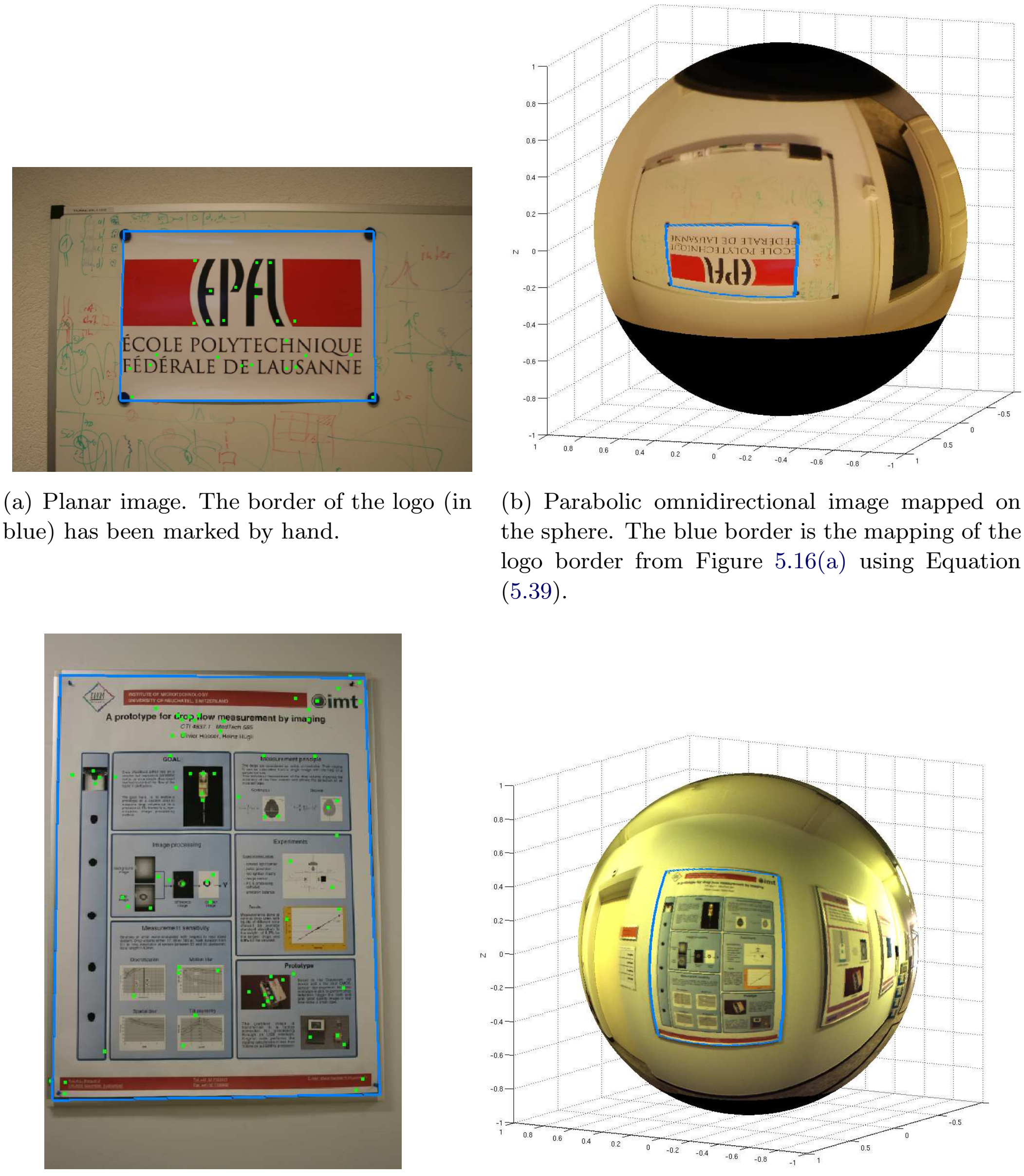
(c) 24 descriptors matched

(d) 29 descriptors matched

(e) 30 descriptors matched

(f) 40 descriptors matched

Figure 5.15: Matchings obtained between LPD of a sequence of spherical images and SIFT descriptors of a planar image. In the sequence of spherical images (shown unwrapped and on the sphere), the camera moves approximately parallel to the object present in the planar image. The matching threshold has been set to 0.6.



(a) Planar image. The border of the logo (in blue) has been marked by hand.

(b) Parabolic omnidirectional image mapped on the sphere. The blue border is the mapping of the logo border from Figure 5.16(a) using Equation (5.39).

(c) Planar image. The border of the poster (in blue) has been marked by hand.

(d) Spherical image. The blue border is the mapping of the poster border from Figure 5.16(c) using Equation (5.39).

Figure 5.16: Examples of estimation of the mapping between the boundary of an object in a planar image to this object in an omnidirectional image using Algorithm 12. The green dots in the planar image are the points whose matching has been used for estimating the mapping.

Many directions for further research can be considered starting from this work. First, it would be interesting to study methods for speeding up the feature extraction procedure. In this sense, the development of approximations of the spherical DoG, as it is done in SURF [Bay 2008] for the planar case, could help in avoiding the use of the spherical Fourier transform. Secondly, the use of local descriptors more adapted to the spherical geometry could improve the matching performance of the algorithm. Indeed, the Gradient Location and Orientation Histogram (GLOH) seems to be very appropriate for spherical images, given that is computed in polar coordinates. Finally, there is a lack on datasets of real omnidirectional images. In particular, the recording of a database of omnidirectional images with three-dimensional information would be extremely useful for a lot of researchers working on omnidirectional vision.

Chapter 6

Conclusions and Perspectives

Contents

6.1 Conclusions

This dissertation contains two slightly separated parts around a common problem: visual matching. In the first part, matching is treated between images with a temporal correlation, i.e. the first part studies the problem of Visual Tracking in video sequences. Under the assumptions of a smooth change of position, shape and appearance, Visual Tracking deals with the problem of matching the tracked object between consecutive frames of a video. In the second part, the temporal relationship is not considered. There, we study the matching between features extracted from two different images, with at least one of them omnidirectional.

In Visual Tracking, the continuous update of the target appearance model is an useful strategy for keeping the tracker constantly adapted to the target. However, bad information about the target added to the model can generate a model drift, and the resulting loss of track. In Chapter 3 we have studied this problem, developing algorithms to minimise the impact of outliers in the model update. We have developed an incremental PCA algorithm that considers weighted samples, the Incremental Temporally Weighted PCA (ITWPCA) algorithm. This algorithm, combined with a measure of the quality of a sample, is the core of the developed Incremental Temporally Weighted Visual Tracking (ITWVT) algorithm. This algorithm tracks a target and simultaneously computes a visual model of it. The samples that update the model are weighted by a measure of their quality with respect to the visual model, minimising the effect of potential outliers added to the model. This tracking approach has been complemented with the capacity of giving more importance to predefined regions of the object of interest, producing the Incremental Temporally Weighted Visual Tracking with Spatial Penalty (ITWVTSP) algorithm. The importance is given by spatial weights applied to the likelihood function of the particles, and produces a higher accuracy of the tracking. All these developed tracking algorithms have been tested in several applications:

face tracking, pedestrian tracking, vehicle tacking and object tracking. The tests have shown the robustness of the algorithm and the increase of accuracy that temporal and spatial weights provide.

A common situation that violates the smoothness hypothesis is the presence of total occlusions. When the occlusion is partial, the algorithm introduced in Chapter 3 reduces the weight of the concerned samples and their impact in the appearance model is low. This allows to continue the tracking without problems once the occlusion has disappeared. However, when the object of interest is completely occluded, i.e. no visual information of the object of interest is perceived by the visual sensor, the ITWVTSP algorithm cannot handle the situation. In Chapter 4 we have studied how this drawback can be handled by a visual tracking algorithm. We have developed the Model cOrruption and Total Occlusion Handling (MOTOH) framework, that combines the algorithms introduced in Chapter 3 with a total occlusion detector that switches between the visual model and a behavioural model of motion. When a total occlusion of the target appears, the tracking switches from the visual model to the behavioural model and continues to track the target with behavioural information instead of visual information. The detection of total occlusions has been developed by using some measures of the particle filter used for the tracking. Pedestrians have a particular behaviour that can be modelled. The Discrete Choice Pedestrian Model (DCPM) is a pedestrian walking behavioural model that reproduces individual pedestrian behaviour based on this pedestrian and its interactions with other pedestrians on the scene. The DCPM has been plugged as behavioural model of motion in the MOTOH framework, producing a pedestrian tracker that successfully handles occlusions. This have been shown in several video sequences without occlusions, with artificial occlusions and with real occlusions. The occlusion handling by behavioural tracking has also potential applications on multi-camera environments, where dead zones between cameras can be treated as occlusions and the behavioural tracking can move particles from a camera to another.

Finally, in Chapter 5, the visual matching problem has been observed from the point of view of two unrelated images, with at least one captured with an omnidirectional camera. The use of omnidirectional cameras requires the design of algorithms that take into account the geometry that these sensors embed into images. In this sense, we have developed a scale invariant feature transform computed in spherical coordinates. Actually, most of the common omnidirectional images can be bijectively mapped on a sphere, which makes that algorithms working on spherical coordinates, consider the geometry of the omnidirectional sensor. The developed algorithm extract points from images in a scale invariant framework. Two feature descriptors have been proposed for the extracted points: Local Spherical Descriptors (LSD) and Local Planar Descriptors (LPD). LSD allow to perform point matching between points on omnidirectional images. LPD can be treated as SIFT descriptors and therefore, be used for matching between points on an omnidirectional image and points on a planar image. A mapping from a region on a planar image to a region on a spherical image has been also introduced. This mapping allows to segment objects on omnidirectional images based on a segmentation in the planar image. All these matchings, between omnidirectional images or between an omnidirectional image and a planar one, have been tested on real images captured under different environmental conditions.

6.2 Perspectives and Future Lines of Research

The works presented in this dissertation could be expanded in several directions:

- **Adaptive Spatial Weights:** Spatial weights introduced in Chapter 3 were fixed and predefined. However, it is easy to imagine situations where an adaptive change of them can be interesting. For instance, in the tests performed in Chapter 3 it has been observed that a high spatial weight on a region where a partial occlusion appears can cause, at least momentarily, a loss of track. Using for instance the reconstruction error, information about the regions containing the occlusion can be obtained, and their spatial weight could be decreased accordingly. More complex strategies can be also imagined, like for instance an algorithm for incrementally learn, simultaneously with the appearance model, an optimal distribution of the spatial weights for the tracked object.

- **Interaction between particle weights and tracking parameters:** The weights of particles give valuable information that is usually not exploited. In Chapter 4, two measures extracted from these weights have shown their power in detecting total occlusions of the tracked object. However, a deeper interaction with the parameters of the tracking algorithm could be fruitful. Indeed, in the tests performed in Chapter 4, we have observed that total occlusions very close in time can difficult the recovery of the visual tracking. The information extracted from the particles could be used for boosting the update of the appearance model, by adapting parameters such as the batch size, the forgetting factor, the temporal weights or even the number of eigenvectors.

- **Multi-camera environments:** It has been already commented that the MOTOH Pedestrian Tracker has a potential application on multi-camera environments. If the occluded region is because of a dead zone, the occlusion detection is easier, since it is simply a pedestrian exiting the region monitored by one of the cameras in the network. However, in this case it is probably interesting to increase the number of hypothesis (particles) during the occlusion in order to consider more potential trajectories given by the behavioural model. The study of the application of the MOTOH framework on a multi-camera environment, with all the particular characteristics involved, is an interesting path to explore.

- **Interaction between vision and exogenous variables of behavioural models:** The Discrete Choice Pedestrian Model, used in Chapter 4, needed the information about the destination of the pedestrian for computing next step probabilities. This situation is not unusual in behaviour models, where some aspects are considered exogenous to the model and therefore given. In the MOTOH Pedestrian Tracker, only direct visual observations fed the behavioural model: current and past positions of every pedestrian currently on the scene. It is clear that pedestrian destination cannot be directly observed when the tracking starts, but visual tracking information can be collected and used on new pedestrians for inferring exogenous variables such as the destination. This would increase the quality of the data given to the behavioural model and, therefore, the quality of its output when behavioural tracking is being performed.

- **Deeper use of behavioural models in computer vision algorithms:** With the MOTOH Pedestrian Tracker, the added value obtained by combining visual and behavioural information has been shown. Behavioural information is present in numerous problems treated in Computer Vision, although it is not sufficiently exploited. Feature extraction is one of these problems. Indeed, humans are able to efficiently track and detect objects, and this experience should be further exploited. In these sense, theories about how humans look at a scene, based on early vision properties, are the foundations of saliency maps [Treisman 1980, Lindeberg 1993, Itti 2000, Rajashekar 2008]. Saliency maps can be used for detecting important regions of a scene and extract useful features and information. The use of behavioural information and models in saliency map computation, in addition to early vision properties, should be investigated since it could increase their quality and therefore the quality of extracted features and information.

- **Spherical approximation of difference of Gaussians:** In the planar case, approximations to the difference of Gaussians are sometimes considered for speeding up scale space computations (see for instance [Bay 2008]). These approximations usually consider Haar-like filters that can be very efficiently applied to an image using the integral image representation. For the omnidirectional case, the development of a similar technique on the sphere would be an important achievement that would boost the use of scale space-like techniques in omnidirectional images.

Appendix A
Complete Statistics for the Dudek Sequence

Complete statistics of the results obtained with IVT, ITWVT/M, ITWVT/R, ITWVTSP/M-Spec, ITWVTSP/M-Iso, ITWVTSP/R-Spec and ITWVTSP/R-Iso. A track is considered lost if the mean RMSE across all the frames is bigger than 10.0px.

IVT				
Losses of Track	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
5	6.8702	7.279	6.2324	0.3964

Table A.1: Statistics of the obtained results on the Dudek sequence using IVT.

ITWVT/M					
ε **Value**	**Losses of Track**	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
0.01	3	8.4058	9.7601	7.6145	0.7917
0.02	2	6.3769	7.7979	5.8036	0.6203
0.03	3	6.2378	7.3919	5.6531	0.6865
0.04	0	7.3831	8.4052	6.5772	0.5698
0.05	0	7.2318	8.3727	6.4899	0.5656
0.06	0	7.0817	8.3856	6.4095	0.5787
0.07	1	6.5527	7.5133	5.6537	0.6473
0.08	0	6.5587	7.2342	6.0085	0.3791
0.09	1	6.3484	6.9601	5.8681	0.3763
0.10	2	6.5412	7.0748	5.6764	0.5273
0.12	1	6.6076	7.0852	6.2457	0.3150
0.14	1	6.8099	7.5038	6.2408	0.4458
0.16	3	6.4705	6.8153	6.1707	0.2097
0.18	4	7.2468	8.5889	6.1936	0.8449
0.20	3	6.5875	7.2460	5.6807	0.5313
0.25	4	6.7973	7.3008	6.1397	0.4934
0.30	3	7.0953	9.9336	5.9823	1.3220
0.35	0	7.0374	8.4965	6.3425	0.7452
0.40	5	6.8801	7.5075	6.3805	0.5100
0.45	1	6.8813	8.0999	6.2154	0.5896
0.50	5	6.4757	7.0297	6.0850	0.3577
0.60	1	6.9352	8.0884	6.2065	0.6743
0.70	3	6.5571	7.5610	6.0677	0.5217
0.80	4	6.6718	7.2075	6.3131	0.3549
0.90	5	7.9066	9.9127	6.2822	1.4074

Table A.2: Statistics of the obtained results on the Dudek sequence using ITWVT/M and varying ε.

ITWVT/R					
ε Value	**Losses of Track**	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
0.01	2	6.4237	9.0563	5.4594	1.3503
0.02	0	7.1984	7.8388	6.3539	0.5411
0.03	0	6.7750	7.3613	6.3504	0.3254
0.04	0	6.3040	7.0071	5.9941	0.3099
0.05	0	6.5492	7.3757	6.0981	0.4334
0.06	0	6.5031	7.2581	5.8174	0.3876
0.07	1	6.6765	7.8109	5.8645	0.7013
0.08	1	6.7496	7.4763	6.1413	0.4116
0.09	2	7.3782	9.6218	6.4713	1.0165
0.10	4	6.5855	7.0886	6.2977	0.2958
0.12	1	6.9439	8.0860	6.1087	0.6061
0.14	3	7.0105	8.7338	6.1794	0.8975
0.16	2	7.2846	8.8147	6.1709	0.9767
0.18	4	6.6198	8.1372	6.1489	0.7598
0.20	4	6.9906	7.8886	6.1627	0.7460
0.25	2	7.1186	8.9511	6.0767	0.8914
0.30	2	7.0153	8.7097	6.0319	0.8136
0.35	4	6.9057	7.4181	6.6161	0.2888
0.40	5	6.8099	7.5384	6.1242	0.5307
0.45	3	6.7475	7.5293	5.7280	0.5996
0.50	6	6.8891	7.9685	6.0177	0.8847
0.60	1	6.7079	7.4916	5.7971	0.5352
0.70	3	6.6355	7.2467	6.2273	0.3910
0.80	5	6.7616	7.5518	6.0889	0.6239
0.90	3	7.0418	7.9330	6.3948	0.6415

Table A.3: Statistics of the obtained results on the Dudek sequence using ITWVT/R and varying ε.

ITWVTSP/M-Spec					
Max Spatial Weight	**Losses of Track**	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
1.2	3	6.1157	6.5954	5.6980	0.3029
1.4	2	5.8037	7.2448	5.1832	0.6446
1.6	1	5.2282	6.0723	4.9648	0.3404
1.8	2	5.2210	5.9466	4.7596	0.3755
2.0	5	4.9565	5.1429	4.7190	0.1765
2.2	6	5.2040	5.8718	4.8028	0.5013
2.4	6	4.9988	5.5915	4.7908	0.3953
2.6	4	5.6069	8.7299	4.7570	1.5463
2.8	6	5.3201	5.8650	4.7597	0.4560
3.0	7	5.7216	7.3102	4.7601	1.3859
3.2	8	5.0908	5.1136	5.0680	0.0322
3.4	9	8.5606	8.5606	8.5606	0.0000

Table A.4: Statistics of the obtained results on the Dudek sequence using ITWVTSP/M-Spec and varying the maximum value of spatial weight ($\varepsilon = 0.07$).

ITWVTSP/M-Iso					
Max Spatial Weight	**Losses of Track**	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
1.2	1	6.3395	7.3405	5.6972	0.4703
1.4	1	6.0340	6.7508	5.7090	0.3229
1.6	0	6.0685	6.5282	5.6664	0.2504
1.8	0	5.8406	6.7670	5.2946	0.4247
2.0	0	5.6698	6.3455	5.3115	0.3508
2.2	0	5.7224	6.1170	5.2639	0.2684
2.4	2	5.7545	6.3612	5.2255	0.3813
2.6	1	5.5202	6.2085	5.1877	0.3344
2.8	1	5.6210	6.7804	5.0467	0.5699
3.0	1	5.6613	6.3716	5.3092	0.3519
3.2	2	5.4659	6.2991	4.9586	0.4604
3.4	8	5.3848	5.7800	4.9896	0.5589

Table A.5: Statistics of the obtained results on the Dudek sequence using ITWVTSP/M-Iso and varying the maximum value of spatial weight ($\varepsilon = 0.07$).

ITWVTSP/R-Spec					
Max Spatial Weight	**Losses of Track**	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
1.2	0	5.8092	6.5682	5.1404	0.4489
1.4	0	5.7330	6.7432	5.1634	0.5469
1.6	0	5.3833	6.5522	4.6985	0.5362
1.8	2	4.8869	5.4463	4.5969	0.2637
2.0	2	5.1666	5.9561	4.7369	0.3956
2.2	5	5.3154	5.6749	4.8982	0.2972
2.4	6	6.4246	7.3709	5.5164	0.9404
2.6	5	5.9692	7.2687	5.5126	0.7376
2.8	4	6.0234	8.3291	5.3102	1.1399
3.0	2	7.0825	9.8382	5.5355	1.5283
3.2	2	6.1169	7.2153	5.1857	0.8506
3.4	5	6.5191	7.3183	5.2347	0.8819

Table A.6: Statistics of the obtained results on the Dudek sequence using ITWVTSP/R-Spec and varying the maximum value of spatial weight ($\varepsilon = 0.07$).

ITWVTSP/R-Iso					
Max Spatial Weight	**Losses of Track**	**Mean RMSE**	**Max RMSE**	**Min RMSE**	**StdDev RMSE**
1.2	1	6.5534	7.2461	6.0495	0.4174
1.4	3	6.7407	9.3743	5.6009	1.2095
1.6	2	6.2685	7.2531	5.4140	0.6783
1.8	1	6.2240	9.6946	5.0292	1.3994
2.0	0	5.5857	5.9252	5.0960	0.2718
2.2	0	5.3548	5.9136	4.6412	0.4747
2.4	0	5.2664	6.4271	4.7385	0.4663
2.6	0	5.2294	5.5833	4.8788	0.2327
2.8	0	5.2520	5.9808	4.9387	0.3533
3.0	0	5.0500	5.3742	4.6198	0.2461
3.2	2	5.2135	5.6375	4.6927	0.3145
3.4	8	5.4944	5.8743	5.1145	0.5372

Table A.7: Statistics of the obtained results on the Dudek sequence using ITWVTSP/R-Iso and varying the maximum value of spatial weight ($\varepsilon = 0.07$).

Appendix B
Plots of RMSE on the Dudek Sequence

Plots of Root Mean Squared Error per frame obtained on the Dudek sequence using TLD, IVT, ITWVT/M, ITWVT/R, ITWVTSP/M-iso, ITWVTSP/M-spec, ITWVTSP/R-iso and ITWVTSP/R-spec. Each plot corresponds to the best run of the ten performed runs for each algorithm.

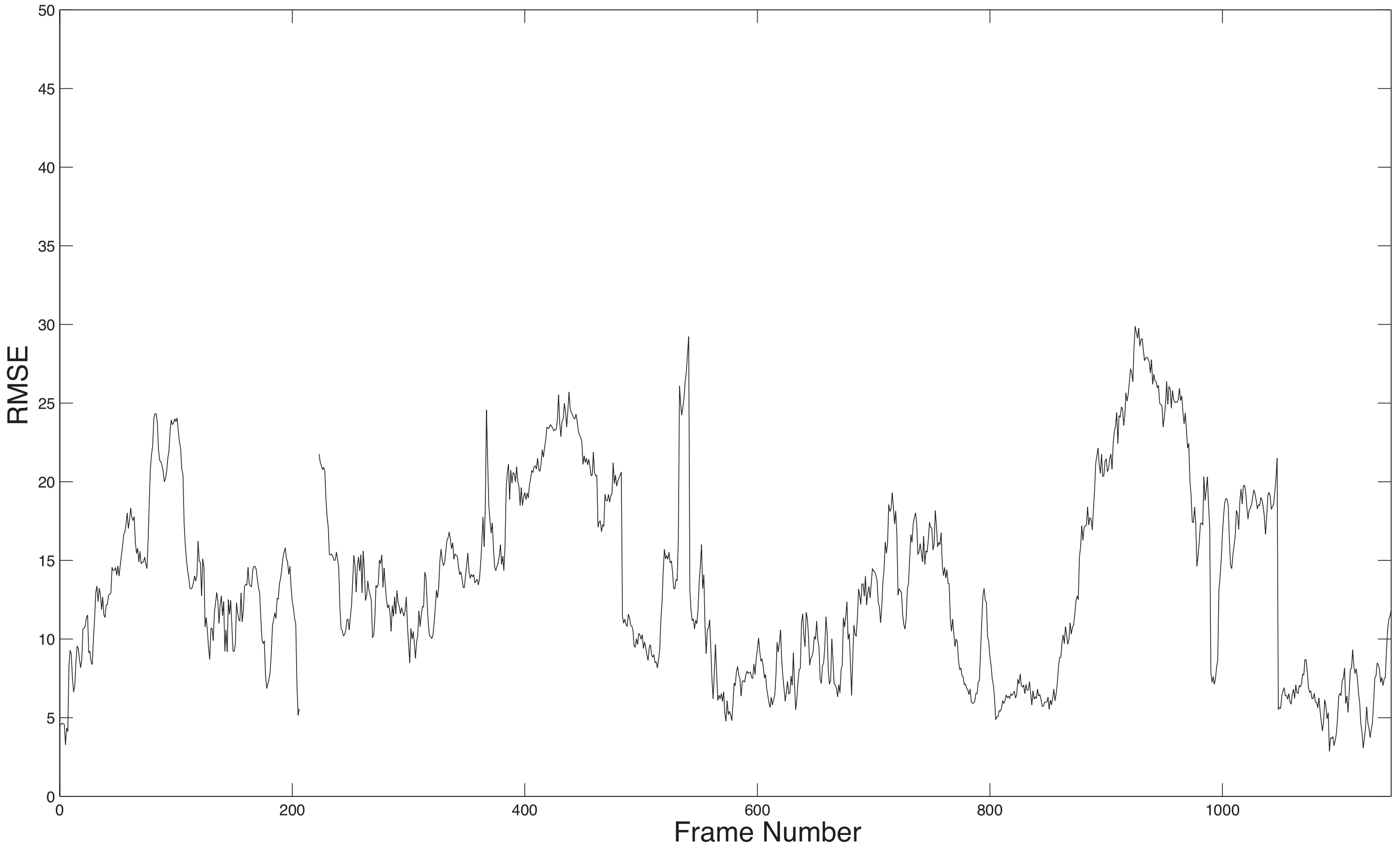


Figure B.1: RMSE per frame on the Dudek sequence using TLD

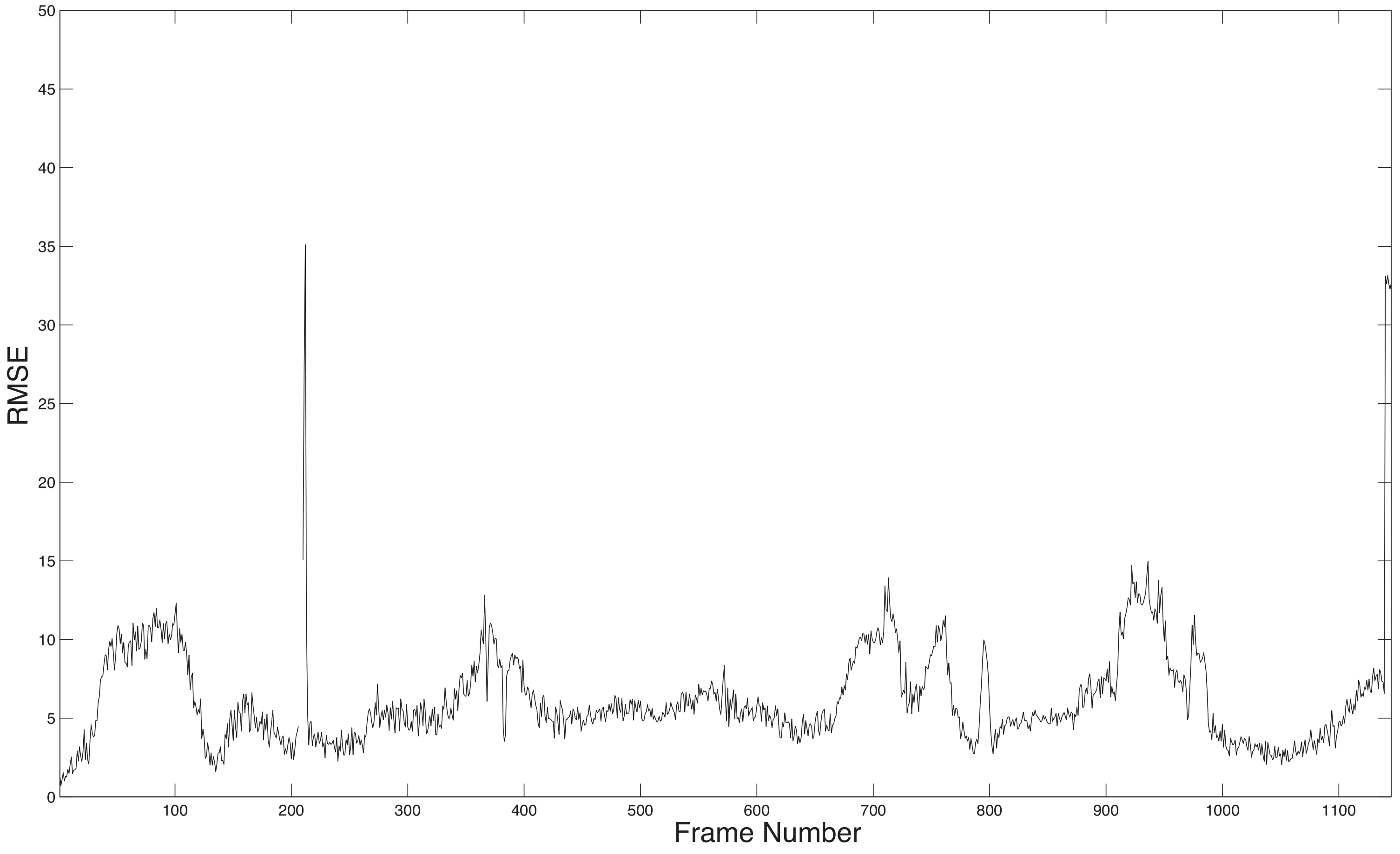


Figure B.2: RMSE per frame on the Dudek sequence using IVT

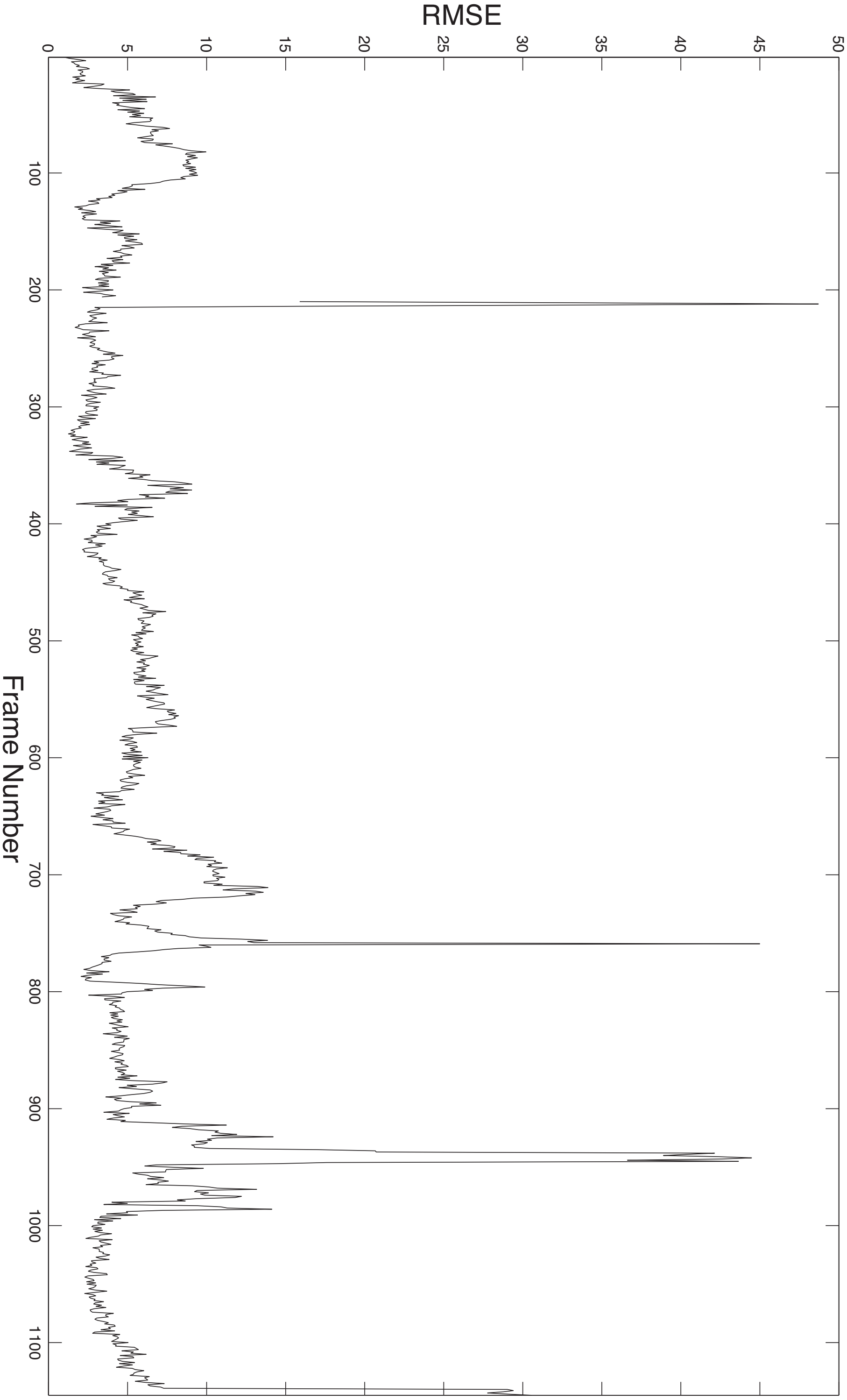


Figure B.3: RMSE per frame on the Dudek sequence using ITWVT/M

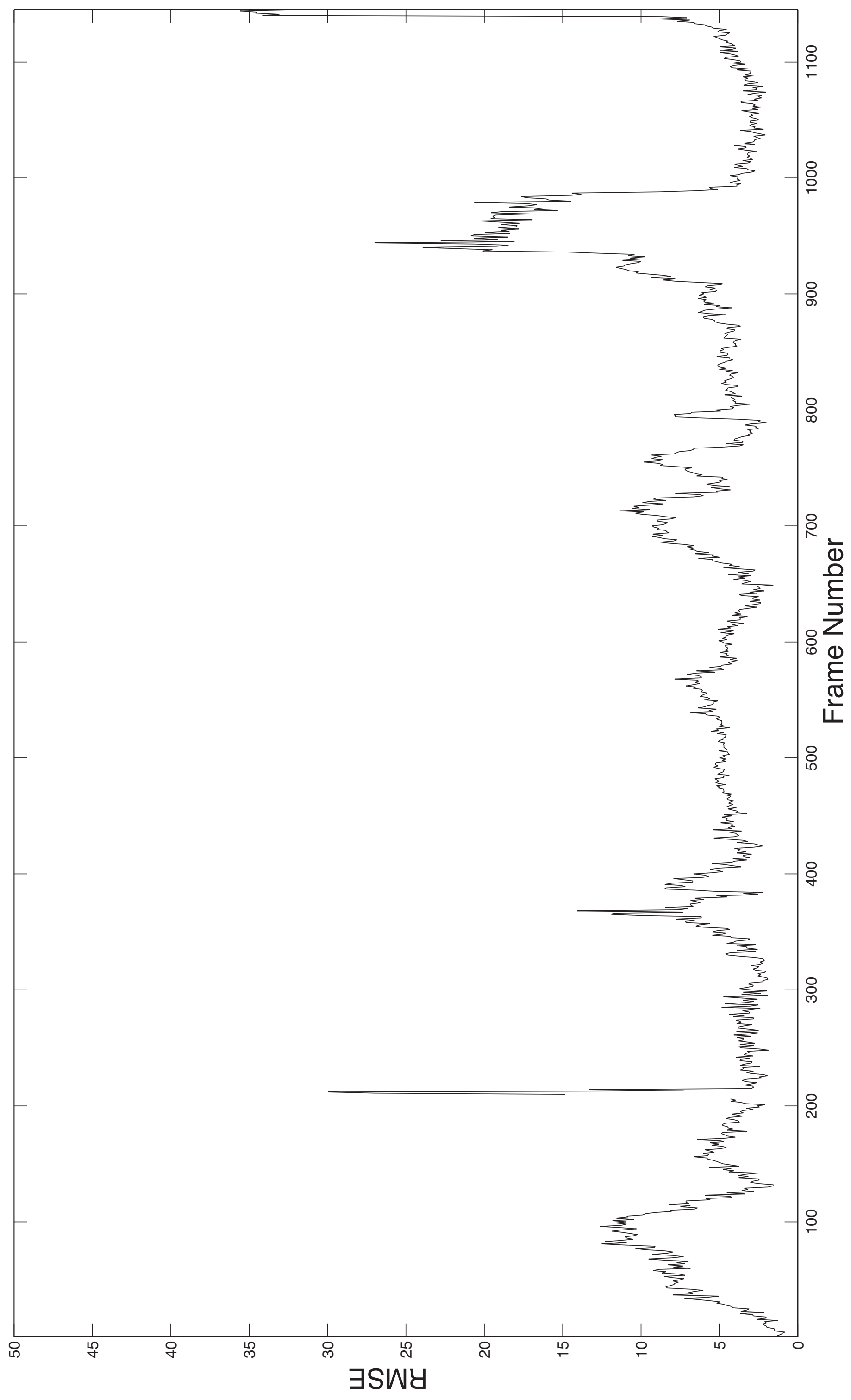


Figure B.4: RMSE per frame on the Dudek sequence using ITWVT/R

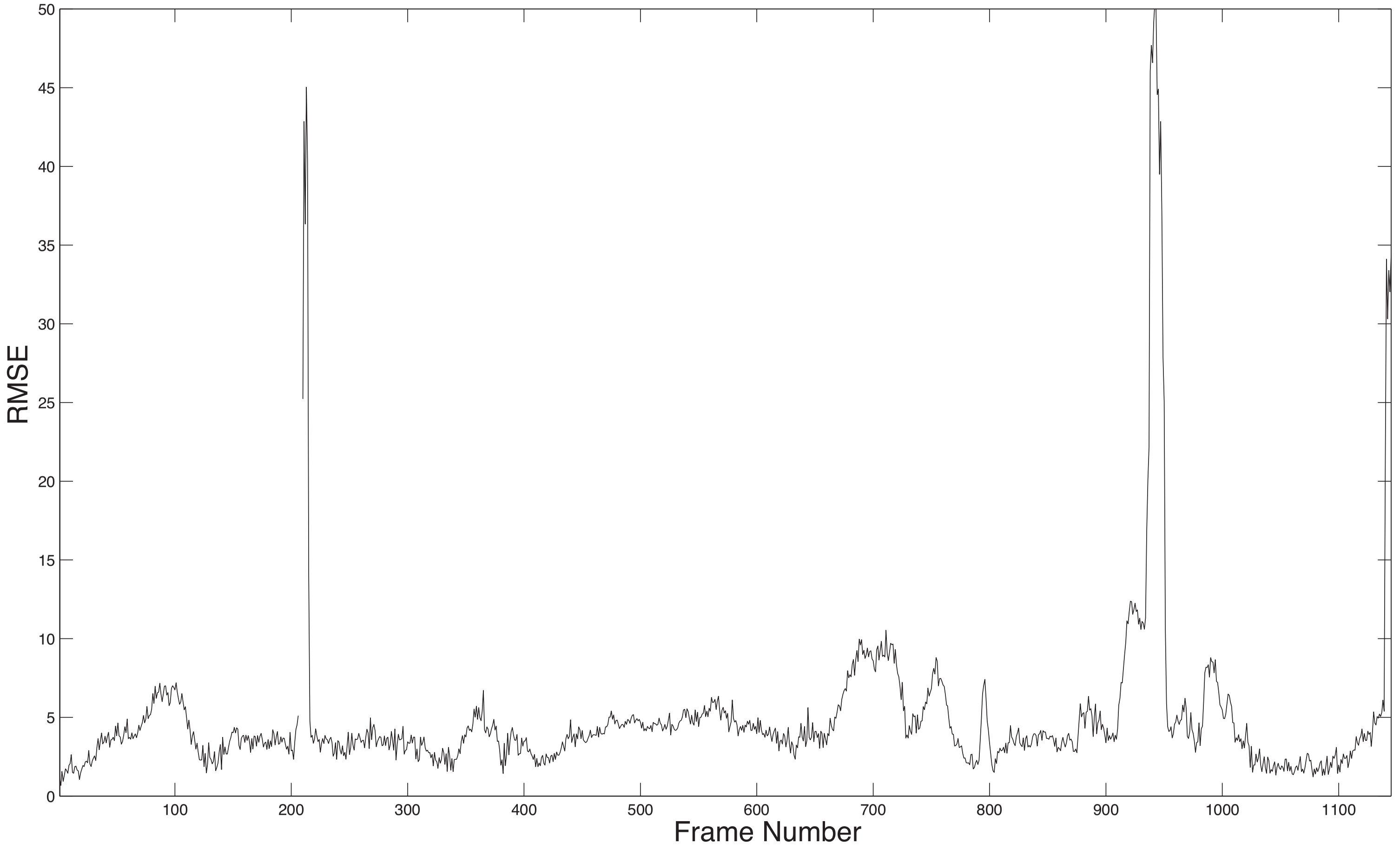


Figure B.5: RMSE per frame on the Dudek sequence using ITWVTSP/M-iso

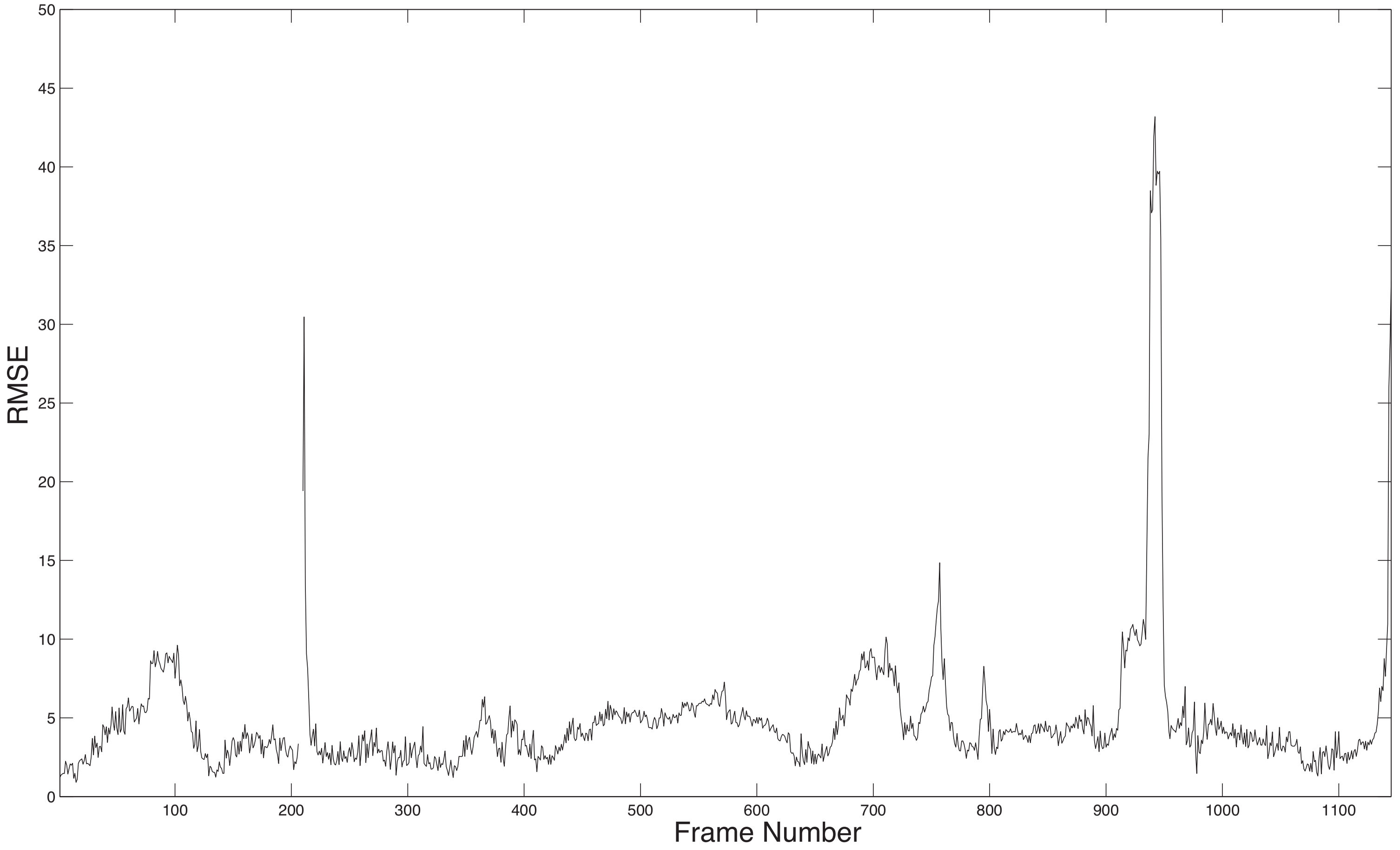


Figure B.6: RMSE per frame on the Dudek sequence using ITWVTSP/M-spec

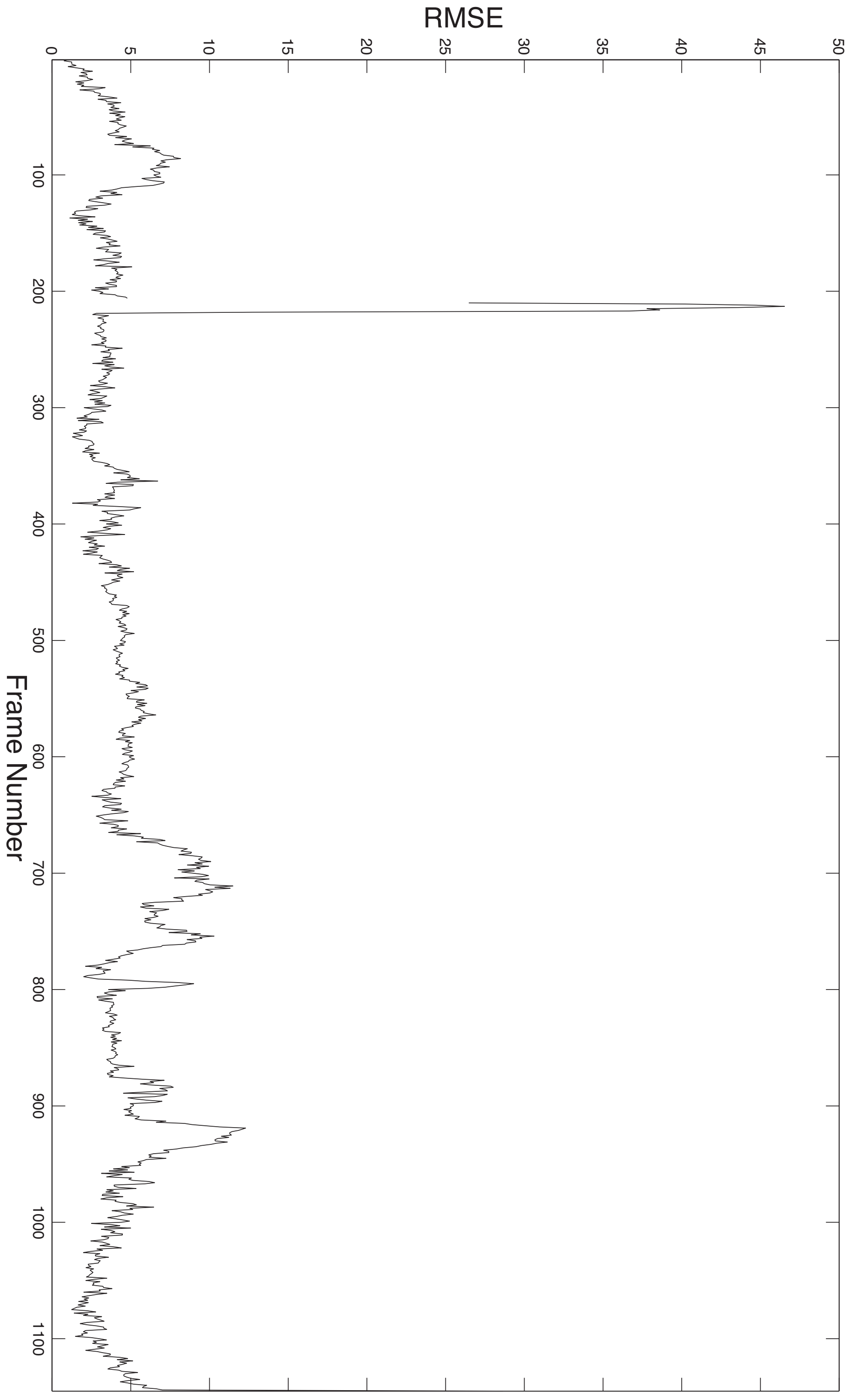


Figure B.7: RMSE per frame on the Dudek sequence using ITWVTSP/R-iso

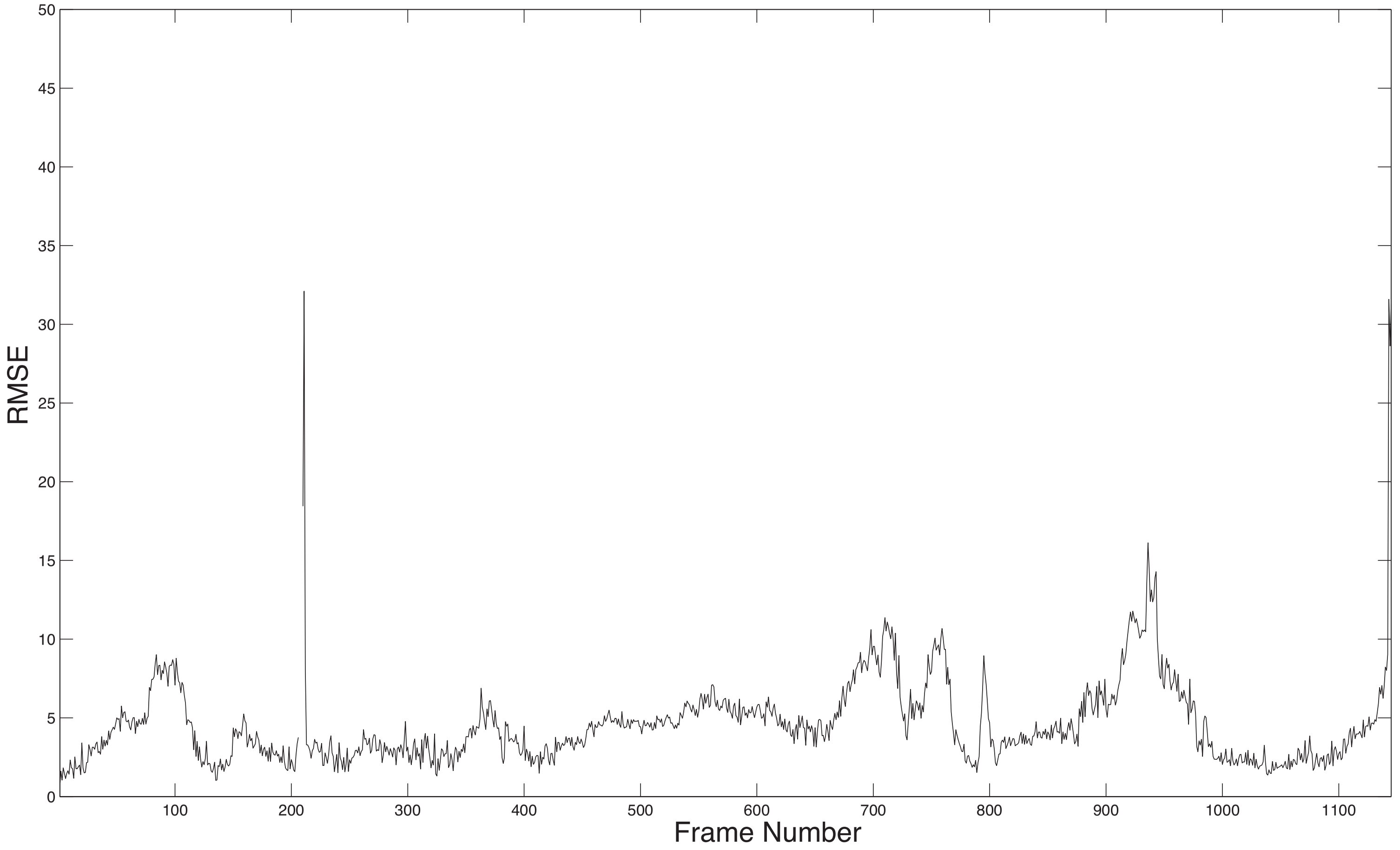


Figure B.8: RMSE per frame on the Dudek sequence using ITWVTSP/R-spec

Bibliography

[Alahi 2011] Alexandre Alahi. *Vision-Based Scene Understanding with Sparsity Promoting Priors.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2011. (Cited on page 61.)

[Alonso 2007] I.P. Alonso, D.F. Llorca, M.A. Sotelo, L.M. Bergasa, P. Revenga de Toro, J. Nuevo, M. Ocana and M.A.G. Garrido. *Combination of Feature Extraction Methods for SVM Pedestrian Detection.* Intelligent Transportation Systems, IEEE Transactions on, vol. 8, no. 2, pages 292–307, june 2007. (Cited on page 61.)

[An 2011] Jun-Ho An and Kwang-Seok Hong. *Finger gesture-based mobile user interface using a rear-facing camera.* In Consumer Electronics (ICCE), 2011 IEEE International Conference on, pages 303–304, 2011. (Cited on page 9.)

[Antonini 2004] Gianluca Antonini, Santiago Venegas, Jean-Philippe Thiran and Michel Bierlaire. *A discrete choice pedestrian behavior model for pedestrian detection in visual tracking systems.* In Proceedings of the Advanced Concepts for Intelligent Vision Systems, Brussels, Belgium, 2004. (Cited on page 63.)

[Antonini 2005] Gianluca Antonini. *A discrete choice modeling framework for pedestrian walking behavior with application to human tracking in video sequences.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2005. 3382. (Cited on page 63.)

[Antonini 2006] Gianluca Antonini, Santiago Venegas, Michel Bierlaire and Jean-Philippe Thiran. *Behavioral priors for detection and tracking of pedestrians in video sequences.* International Journal of Computer Vision, vol. 69, no. 2, pages 159–180, 2006. (Cited on page 63.)

[Arulampalam 2002] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp. *A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking.* Signal Processing, IEEE Transactions on, vol. 50, no. 2, pages 174–188, feb 2002. (Cited on pages 18 and 21.)

[Baillargeon 1985] Renee Baillargeon, Elizabeth S. Spelke and Stanley Wasserman. *Object permanence in five-month-old infants.* Cognition, vol. 20, no. 3, pages 191–208, 1985. (Cited on page 60.)

[Barut 1986] Asim Orhan Barut and Ryszard Rączka. Theory of group representations and applications. World Scientific, 2 édition, 1986. ISBN 9971502178. (Cited on page 101.)

[Baseggio 2010] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi and L. Schenato. *Distributed perimeter patrolling and tracking for camera networks*. In Decision and Control (CDC), 2010 49th IEEE Conference on, pages 2093 –2098, dec. 2010. (Cited on page 9.)

[Baumberg 2000] A. Baumberg. *Reliable Feature Matching across Widely Separated Views*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 774–781, 2000. (Cited on page 97.)

[Bay 2008] H. Bay, A. Ess, T. Tuytelaars and L. J. Van Gool. *Speeded-Up Robust Features (SURF)*. Computer Vision and Image Understanding, vol. 110, no. 3, pages 346–359, June 2008. (Cited on pages 98, 129 and 134.)

[Ben-Akiva 1985] M.E. Ben-Akiva and S.R. Lerman. Discrete choice analysis: Theory and application to travel demand. MIT Press, 1985. (Cited on page 64.)

[Berclaz 2010] Jérôme Berclaz. *Pedestrian localization, tracking and behavior analysis from multiple cameras*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2010. (Cited on page 61.)

[Bergman 1999] Niclas Bergman. *Recursive Bayesian Estimation Navigation and Tracking Applications*. PhD thesis, Linköping University, 1999. (Cited on page 21.)

[Bierlaire 2006] Michel Bierlaire. *A theoretical analysis of the cross-nested logit model*. Annals of Operations Research, vol. 144, no. 1, pages 287–300, 2006. (Cited on page 70.)

[Bierlaire 2009] Michel Bierlaire and Thomas Robin. *Pedestrians Choices*. In H. Timmermans, editeur, Pedestrian Behavior. Models, Data Collection and Applications, pages 1–26. Emerald Group Publishing Limited, 2009. ISBN:978-1-84855-750-5. (Cited on page 63.)

[Bilinski 2009] Piotr Bilinski, Francois Bremond and Mohamed Becha Kaaniche. *Multiple object tracking with occlusions using HOG descriptors and multi resolution images*. In Crime Detection and Prevention (ICDP 2009), 3rd International Conference on, pages 1–6, dec. 2009. (Cited on page 16.)

[Birchfield 1998] S. Birchfield. *Elliptical head tracking using intensity gradients and color histograms*. In Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pages 232–237, jun 1998. (Cited on page 13.)

[Birchfield 2005] S.T. Birchfield and Sriram Rangarajan. *Spatiograms versus histograms for region-based tracking*. In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, volume 2, pages 1158–1163, 2005. (Cited on page 29.)

[Bogdanova 2007] Iva Bogdanova, Xavier Bresson, Jean-Philippe Thiran and Pierre Vandergheynst. *Scale-space analysis and active contours for omnidirectional images*. IEEE Transactions on Image Processing, vol. 16, no. 7, pages 1888–1901, 2007. (Cited on page 98.)

[Boult 2001] T. E. Boult, R. J. Micheals, X. Gao and M. Eckmann. *Into the Woods: Visual Surveillance of Noncooperative and Camouflaged Targets in Complex Outdoor Settings*. Proceedings of IEEE, vol. 89, no. 10, pages 1382–1402, October 2001. (Cited on page 98.)

[Bowyer 2001] Kevin Bowyer, Christine Kranenburg and Sean Dougherty. *Edge Detector Evaluation Using Empirical ROC Curves*. Computer Vision and Image Understanding, vol. 84, no. 1, pages 77–103, 2001. (Cited on page 13.)

[Brand 2006] Matthew Brand. *Fast Low-Rank Modifications of the Thin Singular Value Decomposition*. Linear Algebra and Its Applications, vol. 415, no. 1, pages 20–30, 2006. (Cited on page 30.)

[Breitenstein 2009] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier and L. Van Gool. *Robust tracking-by-detection using a detector confidence particle filter*. In Computer Vision, 2009 IEEE 12th International Conference on, pages 1515–1522, oct 2009. (Cited on pages 10 and 61.)

[Brox 2010] T. Brox, B. Rosenhahn, J. Gall and D. Cremers. *Combined Region and Motion-Based 3D Tracking of Rigid and Articulated Objects*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 3, pages 402–415, 2010. (Cited on page 98.)

[Bulow 2004] Thomas Bulow. *Spherical diffusion for 3D surface smoothing*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 12, pages 1650–1654, December 2004. (Cited on page 102.)

[Bur 2006] A. Bur, A. Tapus, N. Ouerhani, R. Siegwart and H. Hugli. *Robot navigation by panoramic vision and attention guided features*. In Proceedings of the 18th International Conference on Pattern Recognition, pages 695–698, Washington DC, 2006. (Cited on page 94.)

[Canny 1986] John Canny. *A Computational Approach to Edge Detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PAMI, no. 6, pages 679–698, nov 1986. (Cited on page 13.)

[Chen 2008] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan and M. Abidi. *Heterogeneous Fusion of Omnidirectional and PTZ Cameras for Multiple Object Tracking*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 8, pages 1052–1063, August 2008. (Cited on page 98.)

[Chen 2010] Jianjun Chen, Guocheng An, Suofei Zhang and Zhenyang Wu. *A mean shift algorithm based on modified Parzen window for small target tracking*. In

Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 1166–1169, march 2010. (Cited on page 16.)

[Cheng 1995] Yizong Cheng. *Mean shift, mode seeking, and clustering*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 17, no. 8, pages 790–799, aug 1995. (Cited on page 18.)

[Collett 1981] Peter Collett and Peter Marsh. Patterns of public behavior: Collision avoidance on a pedestrian crossing, volume 41, pages 199–218. DE GRUYTER MOUTON, 1981. (Cited on page 68.)

[Comaniciu 2003] D. Comaniciu, V. Ramesh and P. Meer. *Kernel-based object tracking*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 5, pages 564–577, may 2003. (Cited on pages 14 and 18.)

[Cootes 1995] T.F. Cootes, C.J. Taylor, D.H. Cooper and J. Graham. *Active shape models-their training and application*. Comput. Vis. Image Underst., vol. 61, pages 38–59, January 1995. (Cited on page 16.)

[Cootes 2001] T.F. Cootes, G.J. Edwards and C.J. Taylor. *Active Appearance Models*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, pages 681–685, June 2001. (Cited on page 29.)

[Czyz 2007] Jacek Czyz, Branko Ristic and Benoit Macq. *A particle filter for joint detection and tracking of color objects*. Image and Vision Computing, vol. 25, no. 8, pages 1271–1281, 2007. (Cited on page 10.)

[Daamen 2003a] W. Daamen and S.P. Hoogendoorn. *Controlled experiments to derive walking behaviour*. European Journal of Transport and Infrastructure Research, vol. 3, no. 1, pages 39–59, 2003. (Cited on page 70.)

[Daamen 2003b] W. Daamen and S.P. Hoogendoorn. *Experimental Research of Pedestrian Walking Behavior*. Transportation Research Record, vol. 1828, pages 20–30, 2003. (Cited on page 70.)

[Daamen 2004] W. Daamen. *Modelling Passenger Flows in Public Transport Facilities*. PhD thesis, Delft University of Technology, The Netherlands, 2004. (Cited on pages 63 and 70.)

[Dalal 2005] Navneet Dalal and Bill Triggs. *Histograms of Oriented Gradients for Human Detection*. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1 of *CVPR '05*, pages 886–893, 2005. (Cited on page 16.)

[del Rincón 2011] J.M. del Rincón, D. Makris, C.O. Uru nuela and J.-C. Nebel. *Tracking Human Position and Lower Body Parts Using Kalman and Particle Filters Constrained by Human Biomechanics*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 41, no. 1, pages 26–37, feb 2011. (Cited on page 61.)

[Doucet 2009] A. Doucet and A.M. Johansen. *A Tutorial on Particle Filtering and Smoothing: Fifteen years Later.* In D. Crisan and B. Rozovsky, editeurs, Handbook of Nonlinear Filtering. Oxford University Press, 2009. (Cited on pages 18 and 21.)

[Driscoll 1994] James R. Driscoll and Jr. Dennis M. Healy. *Computing Fourier transforms and convolutions on the 2-sphere.* Advances in Applied Mathematics, vol. 15, no. 2, pages 202–250, 1994. (Cited on page 99.)

[Ehlgen 2008] T. Ehlgen, T. Pajdla and D. Ammon. *Eliminating Blind Spots for Assisted Driving.* IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 4, pages 657–665, dec 2008. (Cited on page 98.)

[Enzweiler 2009] M. Enzweiler and D.M. Gavrila. *Monocular Pedestrian Detection: Survey and Experiments.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 12, pages 2179–2195, 2009. (Cited on page 10.)

[Ess 2009] A. Ess, B. Leibe, K. Schindler and L. Van Gool. *Robust Multiperson Tracking from a Mobile Platform.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 10, pages 1831–1846, oct 2009. (Cited on pages 60 and 61.)

[Fischler 1981] M. A. Fischler and R. C. Bolles. *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.* Communications of the ACM, vol. 24, pages 381–395, 1981. (Cited on page 96.)

[Galleguillos 2010] Carolina Galleguillos and Serge Belongie. *Context based object categorization: A critical survey.* Computer Vision and Image Understanding, vol. 114, no. 6, pages 712–722, 2010. Special Issue on Multi-Camera and Multi-Modal Sensor Fusion. (Cited on page 10.)

[Gerónimo 2010] D. Gerónimo, A.M. López, A.D. Sappa and T. Graf. *Survey of Pedestrian Detection for Advanced Driver Assistance Systems.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 7, pages 1239 –1258, jul 2010. (Cited on page 10.)

[Geyer 2001] Christopher Geyer and Konstantinos Daniilidis. *Catadioptric Projective Geometry.* International Journal of Computer Vision, vol. 45, no. 3, pages 223–243, 2001. (Cited on page 98.)

[Goedeme 2005] Toon Goedeme, Tinne Tuytelaars, Luc Van Gool, Gerolf Vanacker and Marnix Nuttin. *Omnidirectional Sparse Visual Path Following with Occlusion-Robust Feature Tracking.* In Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS, 2005. (Cited on pages 94 and 98.)

[Grabner 2006] H. Grabner, M. Grabner and H. Bischof. *Real-time Tracking via On-line Boosting.* In British Machine Vision Conference (BMVC), volume 1, pages 47–56, 2006. (Cited on page 29.)

[Hadj-Abdelkader 2008] H. Hadj-Abdelkader, E. Malis and P. Rives. *Spherical Image Processing for Accurate Visual Odometry with Omnidirectional Cameras.* In Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS, 2008. (Cited on page 98.)

[Hager 1998] G.D. Hager and P.N. Belhumeur. *Efficient region tracking with parametric models of geometry and illumination.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 20, no. 10, pages 1025 –1039, 1998. (Cited on page 17.)

[Haj 2010] M. Al Haj, A.D. Bagdanov, J. Gonzàlez and F.X. Roca. *Reactive Object Tracking with a Single PTZ Camera.* In Pattern Recognition (ICPR), 2010 20th International Conference on, pages 1690–1693, aug. 2010. (Cited on page 18.)

[Hall 1998] P. Hall, D. Marshall and R. Martin. *Incremental Eigenanalysis for Classification.* In British Machine Vision Conference, pages 286–295, 1998. (Cited on page 33.)

[Hansen 2007a] P. Hansen, P. Corke, W. Boles and K. Daniilidis. *Scale invariant feature matching with wide angle images.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1968–1694, 2007. (Cited on page 99.)

[Hansen 2007b] P. Hansen, P. Corke, W. Boles and K. Daniilidis. *Scale-Invariant Features on the Sphere.* In International Conference on Computer Vision, pages 1–8, 2007. (Cited on page 99.)

[Harris 1988] C. Harris and M. Stephens. *A Combined Corner and Edge Detection.* In Proceedings of The Fourth Alvey Vision Conference, pages 147–151, 1988. (Cited on pages 13 and 97.)

[Helbing 1995] D. Helbing and P. Molnar. *Social force model for pedestrian dynamics.* Physical review E, vol. 51, no. 5, pages 4282–4286, 1995. (Cited on page 63.)

[Helbing 2007] Dirk Helbing, Anders Johansson and Habib Z. Al-Abideen. *The Dynamics of Crowd Disasters: An Empirical Study.* Phys. Rev. E, vol. 75, no. 4, page 046109, Apr 2007. (Cited on page 63.)

[Hidaka 2006] A. Hidaka, K. Nishida and T. Kurita. *Face Tracking by Maximizing Classification Score of Face Detector Based on Rectangle Features.* In Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on, pages 48–55, jan 2006. (Cited on page 13.)

[Hoogendoorn 2002] S.P. Hoogendoorn, P.H.L. Bovy and W. Daamen. *Microscopic pedestrian wayfinding and dynamics modelling.* In M. Schreckenberg and S.D. Sharma, editeurs, Pedestrian and Evacuation Dynamics, pages 123–155. Springer, 2002. (Cited on page 63.)

[Horn 1981] B.K.P. Horn and B.G. Schunck. *Determining Optical Flow.* Artificial Intelligence, vol. 17, pages 185–203, 1981. (Cited on page 13.)

[Hu 2009] Weiming Hu, Xue Zhou, Min Hu and Steve Maybank. *Occlusion Reasoning for Tracking Multiple People.* Circuits and Systems for Video Technology, IEEE Transactions on, vol. 19, no. 1, pages 114–121, jan 2009. (Cited on pages 60 and 61.)

[Huang 2008] Kaiqi Huang, Liangsheng Wang, Tieniu Tan and Steve Maybank. *A real-time object detecting and tracking system for outdoor night surveillance.* Pattern Recognition, vol. 41, no. 1, pages 432–444, 2008. (Cited on page 9.)

[Huang 2009] Dong Huang, Zhang Yi and Xiaorong Pu. *A New Incremental PCA Algorithm With Application to Visual Learning and Recognition.* Neural Processing Letters, vol. 30, pages 171–185, 2009. (Cited on page 33.)

[Hubert 2005] Mia Hubert, Peter J. Rousseeuw and Karlien Vanden Branden. *ROBPCA: a new approach to robust principal component analysis.* Technometrics, vol. 47, pages 64–79, 2005. (Cited on page 30.)

[Hubert 2009] Mia Hubert, Peter J. Rousseeuw and Tim Verdonck. *Robust PCA for skewed data and its outlier map.* Computational Statistics & Data Analysis, vol. 53, no. 6, pages 2264–2274, April 2009. (Cited on page 30.)

[Isard 1998] Michael Isard and Andrew Blake. *CONDENSATION Conditional Density Propagation for Visual Tracking.* International Journal of Computer Vision, vol. 29, pages 5–28, 1998. (Cited on pages 18, 20 and 24.)

[Isard 2001] M. Isard and J. MacCormick. *BraMBLe: a Bayesian multiple-blob tracker.* In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, volume 2, pages 34–41, 2001. (Cited on page 61.)

[Itti 2000] Laurent Itti and Christof Koch. *A saliency-based search mechanism for overt and covert shifts of visual attention.* Vision Research, vol. 40, pages 1489–1506, 2000. (Cited on page 134.)

[Iwahori 2008] Y. Iwahori, N. Enda, S. Fukui, H. Kawanaka, R. Woodham and Y. Adachi. *Efficient Tracking with AdaBoost and Particle Filter under Complicated Background.* In I. Lovrek, R. Howlett and L. Jain, editeurs, Knowledge-Based Intelligent Information and Engineering Systems, volume 5178 of *Lecture Notes in Computer Science*, pages 887–894. Springer Berlin / Heidelberg, 2008. (Cited on page 29.)

[Jackson 2004] Donald A. Jackson and Yong Chen. *Robust principal component analysis and outlier detection with ecological data.* Environmetrics, vol. 15, no. 2, pages 129–139, 2004. (Cited on page 30.)

[Jepson 2003] A.D. Jepson, D.J. Fleet and T.F. El-Maraghi. *Robust online appearance models for visual tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 10, pages 1296–1311, oct. 2003. (Cited on pages 14 and 41.)

[Kadir 2001] Timor Kadir and Michael Brady. *Saliency, Scale and Image Description.* International Journal of Computer Vision, vol. 45, no. 2, pages 83–105, 2001. (Cited on page 97.)

[Kalal 2010] Z. Kalal, J. Matas and K. Mikolajczyk. *P-N learning: Bootstrapping binary classifiers by structural constraints.* In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 49 –56, june 2010. (Cited on pages 13, 39 and 60.)

[Kasturi 2009] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova and J. Zhang. *Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 2, pages 319–336, feb 2009. (Cited on page 27.)

[Kelly 2008] P. Kelly and N.E. O'Connor. *Vision-based analysis of pedestrian traffic data.* In Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on, pages 133–140, jun 2008. (Cited on page 61.)

[Kelly 2009] Philip Kelly, Noel E. O'Connor and Alan F. Smeaton. *Robust pedestrian detection and tracking in crowded scenes.* Image and Vision Computing, vol. 27, no. 10, pages 1445–1458, 2009. Special Section: Computer Vision Methods for Ambient Intelligence. (Cited on page 60.)

[Kim 2008] ZuWhan Kim. *Real time object tracking based on dynamic feature grouping with background subtraction.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, june 2008. (Cited on page 13.)

[Kriegel 2008] Hans-Peter Kriegel, Peer Kriegelöger, Erich Schubert and Arthur Zimek. *A general framework for increasing the robustness of PCA-based correlation clustering algorithms.* In Proceedings of the 20th international conference on Scientific and Statistical Database Management, pages 418–435, 2008. (Cited on page 33.)

[Lan 2004] Xiangyang Lan and D.P. Huttenlocher. *A unified spatio-temporal articulated model for tracking.* In Computer Vision and Pattern Recognition,

2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 1, pages 722–729, 2004. (Cited on page 14.)

[Lee 2005] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang and David Kriegman. *Visual tracking and recognition using probabilistic appearance manifolds.* Computer Vision and Image Understanding, vol. 99, no. 3, pages 303–331, 2005. (Cited on page 29.)

[Leibe 2008] B. Leibe, K. Schindler, N. Cornelis and L. Van Gool. *Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 10, pages 1683–1698, oct 2008. (Cited on page 13.)

[Leichter 2010] Ido Leichter, Michael Lindenbaum and Ehud Rivlin. *Mean Shift tracking with multiple reference color histograms.* Computer Vision and Image Understanding, vol. 114, no. 3, pages 400–408, 2010. (Cited on page 16.)

[Levy 2000] A. Levy and M. Lindenbaum. *Sequential Karhunen-Loeve Basis Extraction and its Application to Images.* IEEE Transactions on Image Processing, vol. 9, no. 8, pages 1371–1374, 2000. (Cited on pages 22 and 30.)

[Leykin 2006] A. Leykin and R. Hammoud. *Robust Multi-Pedestrian Tracking in Thermal-Visible Surveillance Videos.* In Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on, pages 136–143, jun 2006. (Cited on page 60.)

[Li 2001] Tsai-Yen Li, Ying-Jiun Jeng and Shih-I Chang. *Simulating virtual human crowds with a leader-follower model.* In Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings, pages 93–102, 2001. (Cited on page 67.)

[Li 2010] Xin Li, Kejun Wang, Wei Wang and Yang Li. *A multiple object tracking method using Kalman filter.* In Information and Automation (ICIA), 2010 IEEE International Conference on, pages 1862–1866, june 2010. (Cited on page 18.)

[Lindeberg 1993] Tony Lindeberg. *Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention.* International Journal of Computer Vision, vol. 11, pages 283–318, 1993. (Cited on page 134.)

[Lindeberg 1998] Tony Lindeberg. *Feature Detection with Automatic Scale Selection.* International Journal of Computer Vision, vol. 30, no. 2, pages 79–116, November 1998. (Cited on page 97.)

[Lowe 2004] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, November 2004. (Cited on pages 13, 97, 103, 106, 114, 121 and 126.)

[Lu 2001] Wenmiao Lu and Yap-Peng Tan. *A color histogram based people tracking system.* In Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, volume 2, pages 137–140, may 2001. (Cited on page 16.)

[Lu 2006] Wei-Lwun Lu and J.J. Little. *Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor.* In Computer and Robot Vision, 2006. The 3rd Canadian Conference on, page 6, june 2006. (Cited on page 16.)

[Lucas 1981] B.D. Lucas and Takeo Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision.* In International Joint Conference on Artificial Intelligence, pages 674–679, 1981. (Cited on pages 13 and 18.)

[Maggio 2007] E. Maggio, F. Smerladi and A. Cavallaro. *Adaptive Multifeature Tracking in a Particle Filtering Framework.* Circuits and Systems for Video Technology, IEEE Transactions on, vol. 17, no. 10, pages 1348–1359, oct 2007. (Cited on page 14.)

[Maggio 2009] Emilio Maggio and Andrea Cavallaro. *Accurate appearance-based Bayesian tracking for maneuvering targets.* Computer Vision and Image Understanding, vol. 113, no. 4, pages 544–555, 2009. (Cited on page 18.)

[Maggio 2010] E. Maggio and A. Cavallaro. Video tracking: Theory and practice. Wiley and Sons, 2010. (Cited on pages 11, 12 and 26.)

[Marimon 2007a] D. Marimon and T. Ebrahimi. *Orientation histogram-based matching for region tracking.* In Image Analysis for Multimedia Interactive Services, 2007. WIAMIS '07. Eighth International Workshop on, page 8, june 2007. (Cited on page 16.)

[Marimon 2007b] David Marimon and Touradj Ebrahimi. *Combination of video-based camera trackers using a dynamically adapted particle filter.* In Proc. 2nd International Conference on Computer Vision Theory and Applications (VISAPP07), pages 363–370, 2007. (Cited on page 9.)

[Matas 2002] J. Matas, O. Chum, M. Urba and T. Pajdla. *Robust wide baseline stereo from maximally stable extremal regions.* In Proceedings of the British Machine Vision Conference, pages 384–393, 2002. (Cited on page 97.)

[Mauthner 2006] T. Mauthner, F. Fraundorfer and H. Bischof. *Region matching for omnidirectional images using virtual camera planes.* In Proceedings of the Computer Vision Winter Workshop, 2006. (Cited on pages 99 and 121.)

[McKenna 1999] Stephen J. McKenna, Yogesh Raja and Shaogang Gong. *Tracking colour objects using adaptive mixture models.* Image and Vision Computing, vol. 17, no. 3-4, pages 225–231, 1999. (Cited on page 16.)

[Menegatti 2006] E. Menegatti, A. Pretto, A. Scarpa and E. Pagello. *Omnidirectional vision scan matching for robot localization in dynamic environments.* IEEE Transactions on Robotics, vol. 22, no. 3, pages 523–535, jun 2006. (Cited on page 98.)

[Meynet 2008] Julien Meynet, Taner Arsan, Javier Cruz-Mota and Jean-Philippe Thiran. *Fast multi-view face tracking with pose estimation.* In Proceeedings of the 16th European Signal Processing Conference, 2008. (Cited on page 13.)

[Mikolajczyk 2005a] K. Mikolajczyk and C. Schmid. *A Performance Evaluation of Local Descriptors.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pages 1615–1630, October 2005. (Cited on pages 97 and 98.)

[Mikolajczyk 2005b] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. van Gool. *A comparison of affine region detectors.* International Journal of Computer Vision, vol. 65, no. 1, pages 43–72, 2005. (Cited on pages 113 and 114.)

[Moreno 2002] F. Moreno, A. Tarrida, J. Andrade-Cetto and A. Sanfeliu. *3D real-time head tracking fusing color histograms and stereovision.* In Pattern Recognition, 2002. Proceedings. 16th International Conference on, volume 1, pages 368–371, 2002. (Cited on page 16.)

[Munder 2008] S. Munder, C. Schnorr and D.M. Gavrila. *Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models.* Intelligent Transportation Systems, IEEE Transactions on, vol. 9, no. 2, pages 333–343, jun 2008. (Cited on pages 60 and 61.)

[Mundy 2006] Joseph Mundy. *Object Recognition in the Geometric Era: A Retrospective.* In Jean Ponce, Martial Hebert, Cordelia Schmid and Andrew Zisserman, editeurs, Toward Category-Level Object Recognition, volume 4170 of *Lecture Notes in Computer Science*, pages 3–28. Springer Berlin / Heidelberg, 2006. (Cited on page 10.)

[Nawaz 2011] T. Nawaz and A. Cavallaro. *PFT: A protocol for evaluating video trackers.* In Image Processing, 2011. ICIP 2011. IEEE Conference on, sep 2011. (Cited on page 27.)

[Ndiour 2010] I.J. Ndiour and P.A. Vela. *A local extended Kalman filter for visual tracking.* In Decision and Control (CDC), 2010 49th IEEE Conference on, pages 2498–2504, dec. 2010. (Cited on page 18.)

[Nghiem 2007] A.T. Nghiem, F. Bremond, M. Thonnat and V. Valentin. *ETISEO, performance evaluation for video surveillance systems.* In Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on, pages 476 –481, sep 2007. (Cited on page 27.)

[Olson 2008] D.L. Olson and Dursun Delen. Advanced data mining techniques. Springer, 2008. (Cited on page 25.)

[Pai 2004] Chia-Jung Pai, Hsiao-Rong Tyan, Yu-Ming Liang, Hong-Yuan Mark Liao and Sei-Wang Chen. *Pedestrian detection and tracking at crossroads*. Pattern Recognition, vol. 37, no. 5, pages 1025–1034, 2004. (Cited on page 60.)

[Pal 2008] Amit Pal. *Robust face tracking with occlusion detection and varying intensity*. In Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on, pages 1–6, sept 2008. (Cited on page 60.)

[Papadourakis 2010] Vasilis Papadourakis and Antonis Argyros. *Multiple objects tracking in the presence of long-term occlusions*. Computer Vision and Image Understanding, vol. 114, no. 7, pages 835–846, 2010. (Cited on pages 16, 29 and 60.)

[Park 2004] Sangho Park and Jake K. Aggarwal. *A hierarchical Bayesian network for event recognition of human actions and interactions*. Multimedia Systems, vol. 10, no. 2, pages 164–179, 2004. (Cited on page 29.)

[Pearson 1901] K. Pearson. *On lines and planes of closest fit to systems of points in space*. Philosophical Magazine, vol. 2, no. 6, pages 559–572, 1901. (Cited on page 30.)

[Peng 2005] Ning-Song Peng, Jie Yang and Zhi Liu. *Mean shift blob tracking with kernel histogram filtering and hypothesis testing*. Pattern Recognition Letters, vol. 26, no. 5, pages 605–614, 2005. (Cited on page 29.)

[Piccoli 2011] Benedetto Piccoli and Andrea Tosin. *Time-Evolving Measures and Macroscopic Modeling of Pedestrian Flow*. Archive for Rational Mechanics and Analysis, vol. 199, pages 707–738, 2011. (Cited on page 63.)

[Piotto 2009] N. Piotto, N. Conci and F.G.B. De Natale. *Syntactic Matching of Trajectories for Ambient Intelligence Applications*. Multimedia, IEEE Transactions on, vol. 11, no. 7, pages 1266–1275, nov 2009. (Cited on page 60.)

[Polat 2003] Ediz Polat, Mohammed Yeasin and Rajeev Sharma. *Robust tracking of human body parts for collaborative human computer interaction*. Computer Vision and Image Understanding, vol. 89, no. 1, pages 44–69, 2003. (Cited on page 9.)

[Rajashekar 2008] U. Rajashekar, I. van der Linde, A.C. Bovik and L.K. Cormack. *GAFFE: a gaze-attentive fixation finding engine*. IEEE Transactions on Image Processing, vol. 17, no. 4, pages 564–573, April 2008. (Cited on page 134.)

[Reddy 1996] B.S. Reddy and B.N. Chatterji. *An FFT-based technique for translation, rotation, and scale-invariant image registration.* Image Processing, IEEE Transactions on, vol. 5, no. 8, pages 1266–1271, aug 1996. (Cited on page 13.)

[Reinartz 2006] Peter Reinartz, Marie Lachaise, Elisabeth Schmeer, Thomas Krauss and Hartmut Runge. *Traffic monitoring with serial images from airborne cameras.* ISPRS Journal of Photogrammetry and Remote Sensing, vol. 61, no. 3-4, pages 149–158, 2006. Theme Issue: Airborne and Spaceborne Traffic Monitoring. (Cited on page 9.)

[Robin 2009] Thomas Robin, Gianluca Antonini, Michel Bierlaire and Javier Cruz. *Specification, estimation and validation of a pedestrian walking behavior model.* Transportation Research Part B: Methodological, vol. 43, no. 1, pages 36–56, 2009. (Cited on pages 5, 62, 63, 64, 65, 69 and 71.)

[Rodrigues 1840] Olinde Rodrigues. *Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire.* Journal de Mathématiques Pures et Appliquées, vol. 5, pages 380–440, 1840. (Cited on page 108.)

[Roh 2007] Myung-Cheol Roh, Tae-Yong Kim, Jihun Park and Seong-Whan Lee. *Accurate object contour tracking based on boundary edge selection.* Pattern Recognition, vol. 40, no. 3, pages 931–943, 2007. (Cited on page 14.)

[Roller 1993] D. Roller, K. Daniilidis and H.H. Nagel. *Model-based object tracking in monocular image sequences of road traffic scenes.* International Journal of Computer Vision, vol. 10, pages 257–281, 1993. (Cited on page 14.)

[Ross 2008] David A. Ross, Jongwoo Lim, Ruei-Sung Lin and Ming-Hsuan Yang. *Incremental Learning for Robust Visual Tracking.* International Journal of Computer Vision, vol. 77, no. 1-3, pages 125–141, 2008. (Cited on pages 14, 16, 18, 21, 23, 24, 25, 29, 30, 33, 36, 39 and 40.)

[Santis 2009] Alberto De Santis and Daniela Iacoviello. *Robust real time eye tracking for computer interface for disabled people.* Computer Methods and Programs in Biomedicine, vol. 96, no. 1, pages 1–11, 2009. (Cited on page 9.)

[Scaramuzza 2008] D. Scaramuzza and R. Siegwart. *Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles.* IEEE Transactions on Robotics, vol. 24, no. 5, pages 1015–1026, oct 2008. (Cited on pages 94 and 98.)

[Semertzidis 2010] T. Semertzidis, K. Dimitropoulos, A. Koutsia and N. Grammalidis. *Video sensor network for real-time traffic monitoring and surveillance.* Intelligent Transport Systems, IET, vol. 4, no. 2, pages 103 –112, june 2010. (Cited on page 9.)

[Shan 2007] Caifeng Shan, Tieniu Tan and Yucheng Wei. *Real-time hand tracking using a mean shift embedded particle filter.* Pattern Recognition, vol. 40, no. 7, pages 1958–1970, 2007. (Cited on page 18.)

[Shin 2005] Jeongho Shin, Sangjin Kim, Sangkyu Kang, Seong-Won Lee, Joonki Paik, Besma Abidi and Mongi Abidi. *Optical flow-based real-time object tracking using non-prior training active feature model.* Real-Time Imaging, vol. 11, no. 3, pages 204–218, 2005. Special Issue on Video Object Processing. (Cited on page 13.)

[Sirmacek 2009] B. Sirmacek and C. Unsalan. *Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory.* IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 4, pages 1156–1167, apr 2009. (Cited on page 98.)

[Skočaj 2007] Danijel Skočaj, Ales Leonardis and Horst Bischof. *Weighted and robust learning of subspace representations.* Pattern Recognition, vol. 40, no. 5, pages 1556–1569, 2007. (Cited on pages 33, 34 and 37.)

[Skočaj 2008] D. Skočaj and A. Leonardis. *Incremental and robust learning of subspace representations.* Image Vision Comput., vol. 26, pages 27–38, January 2008. (Cited on page 33.)

[Smith 1997] S. M. Smith and J. M. Brady. *Susan - a new approach to low level image processing.* International Journal of Computer Vision, vol. 23, pages 45–78, 1997. (Cited on page 97.)

[Stauffer 1999] C. Stauffer and W.E.L. Grimson. *Adaptive background mixture models for real-time tracking.* In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on., volume 2, pages 637–663, 1999. (Cited on page 29.)

[Stauffer 2000] C. Stauffer and W.E.L. Grimson. *Learning patterns of activity using real-time tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pages 747–757, aug 2000. (Cited on page 13.)

[Sundaresan 2009] A. Sundaresan and R. Chellappa. *Multicamera Tracking of Articulated Human Motion Using Shape and Motion Cues.* Image Processing, IEEE Transactions on, vol. 18, no. 9, pages 2114–2126, sept. 2009. (Cited on pages 14 and 61.)

[Tamimi 2006] Hashem Tamimi, Henrik Andreasson, André Treptow, Tom Duckett and Andreas Zell. *Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT.* Robotics and Autonomous Systems, vol. 54, no. 9, pages 758 – 765, 2006. Selected papers from the 2nd European Conference on Mobile Robots (ECMR '05). (Cited on pages 94 and 98.)

[Teknomo 2000] K. Teknomo, Y. Takeyama and H. Inamura. *Review on microscopic pedestrian simulation model.* In Proceedings Japan Society of Civil Engineering Conference, Morioka, Japan, March 2000. (Cited on page 70.)

[Teknomo 2002] K. Teknomo. *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model.* PhD thesis, Tohoku University, Japan, Sendai, 2002. (Cited on page 70.)

[Tipping 1999] Michael E. Tipping and Chris M. Bishop. *Probabilistic Principal Component Analysis.* Journal of the Royal Statistical Society, Series B, vol. 61, pages 611–622, 1999. (Cited on page 23.)

[Tomasi 1991] Carlo Tomasi and Takeo Kanade. *Detection and Tracking of Point Features.* Rapport technique CMU-CS-91-132, International Journal of Computer Vision, April 1991. (Cited on page 18.)

[Treisman 1980] Anne M. Treisman and Garry Gelade. *A feature-integration theory of attention.* Cognitive Psychology, vol. 12, no. 1, pages 97–136, 1980. (Cited on page 134.)

[Tuytelaars 2007] Tinne Tuytelaars and Krystian Mikolajczyk. *Local Invariant Feature Detectors: A Survey.* Foundations and Trends in Computer Graphics and Vision, vol. 3, no. 3, pages 177–280, 2007. (Cited on page 97.)

[Tygert 2008] Mark Tygert. *Fast algorithms for spherical harmonic expansions, II.* Journal of Computational Physics, vol. 227, no. 8, pages 4260–4279, 2008. (Cited on page 126.)

[Valgren 2007] Christoffer Valgren and Achim Lilienthal. *SIFT, SURF and Seasons: Long-term Outdoor Localization Using Local Features.* In Proceedings of the European Conference on Mobile Robots (ECMR), 2007. (Cited on pages 94 and 98.)

[Van Gool 1996] L. J. Van Gool, T. Moons and D. Ungureanu. *Affine/Photometric Invariants for Planar Intensity Patterns.* In Computer Vision - ECCV'96, pages 642–651, 1996. (Cited on page 97.)

[Vedaldi 2008] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms.* `http://www.vlfeat.org/`, 2008. (Cited on pages 113 and 120.)

[Veenman 2001] C.J. Veenman, M.J.T. Reinders and E. Backer. *Resolving motion correspondence for densely moving points.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 1, pages 54–72, jan 2001. (Cited on pages 14 and 17.)

[Venegas 2005] Santiago Venegas, Gianluca Antonini, Jean-Philippe Thiran and Michel Bierlaire. *Automatic Pedestrian Tracking Using Discrete Choice*

Models and Image Correlation Techniques. In Bengio S. and Bourlard H., editeurs, Machine Learning for Multimodal Interaction, volume 3361 of *Lecture Notes in Computer Science*, pages 341 – 348. Springer, 2005. ISBN:978-3540245094. (Cited on pages 61 and 63.)

[Verma 2003] R.C. Verma, C. Schmid and K. Mikolajczyk. *Face detection and tracking in a video by propagating detection probabilities.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 10, pages 1215–1228, oct 2003. (Cited on page 13.)

[Vincenty 1975] T. Vincenty. *Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations.* Survey Review, vol. XXIII, no. 176, pages 88–93, 1975. (Cited on page 108.)

[Wang 2008a] Junqiu Wang and Yasushi Yagi. *Integrating Color and Shape-Texture Features for Adaptive Real-Time Object Tracking.* Image Processing, IEEE Transactions on, vol. 17, no. 2, pages 235–240, feb. 2008. (Cited on pages 12 and 14.)

[Wang 2008b] Xiaoyan Wang, Yangsheng Wang, Xuetao Feng and Mingcai Zhou. *On edge structure based adaptive observation model for facial feature tracking.* In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1–4, dec 2008. (Cited on page 13.)

[Wang 2009] Zhaowen Wang, Xiaokang Yang, Yi Xu and Songyu Yu. *CamShift guided particle filter for visual tracking.* Pattern Recognition Letters, vol. 30, no. 4, pages 407–413, 2009. (Cited on page 18.)

[Wang 2010] Zhelong Wang, Chuan Dai and Hongyu Zhao. *A real time object tracking system for contrast media injection.* In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on, pages 3749–3753, oct 2010. (Cited on page 13.)

[Wu 2006a] Bo Wu and Ram Nevatia. *Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection.* In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 951–958, june 2006. (Cited on page 61.)

[Wu 2006b] Ying Wu and Ting Yu. *A field model for human detection and tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 28, no. 5, pages 753–765, may 2006. (Cited on page 61.)

[Wu 2008] Peiliang Wu, Lingfu Kong, Fengda Zhao and Xianshan Li. *Particle filter tracking based on color and SIFT features.* In Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on, pages 932–937, july 2008. (Cited on page 14.)

[Yamane 1998] T. Yamane, Y. Shirai and J. Miura. *Person tracking by integrating optical flow and uniform brightness regions.* In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, volume 4, pages 3267–3272, may 1998. (Cited on page 13.)

[Yang 1996] Jie Yang and Alex Waibel. *A real-time face tracker.* In Applications of Computer Vision, 1996. WACV '96., Proceedings 3rd IEEE Workshop on, pages 142 –147, dec 1996. (Cited on page 16.)

[Yang 2002] M.-H. Yang, D.J. Kriegman and N. Ahuja. *Detecting faces in images: a survey.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 1, pages 34–58, 2002. (Cited on page 10.)

[Yilmaz 2004] Alper Yilmaz, Xin Li and Mubarak Shah. *Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, pages 1531–1536, November 2004. (Cited on page 14.)

[Yilmaz 2006] Alper Yilmaz, Omar Javed and Mubarak Shah. *Object tracking: A survey.* ACM Comput. Surv., vol. 38, December 2006. (Cited on page 11.)

[Yu 2010] Gang Yu, Zhiwei Hu, Hongtao Lu and Wenbin Li. *Robust object tracking with occlusion handle.* Neural Computing and Applications, pages 1–8, 2010. (Cited on page 60.)

[Yuen 2005] D.C.K. Yuen and B.A. MacDonald. *Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison.* IEEE Transactions on Robotics, vol. 21, no. 2, pages 217–226, apr 2005. (Cited on page 94.)

[Zabih 1994] Ramin Zabih and John Woodfill. *Non-parametric local transforms for computing visual correspondence.* In Computer Vision - ECCV'94, Lecture Notes in Computer Science, pages 151–158. Springer Berlin/Heidelberg, 1994. (Cited on page 97.)

[Zach 2007] C. Zach, T. Pock and H. Bischof. *A duality based approach for realtime tv-l1 optical flow.* In In Ann. Symp. German Association Patt. Recogn, pages 214–223, 2007. (Cited on page 13.)

[Zhou 2004] S.K. Zhou, R. Chellappa and B. Moghaddam. *Visual tracking and recognition using appearance-adaptive models in particle filters.* Image Processing, IEEE Transactions on, vol. 13, no. 11, pages 1491–1506, 2004. (Cited on page 18.)

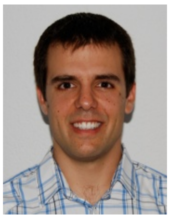
[Zhou 2009] Huiyu Zhou, Yuan Yuan and Chunmei Shi. *Object tracking using SIFT features and mean shift.* Computer Vision and Image Understanding, vol. 113, no. 3, pages 345–352, 2009. Special Issue on Video Analysis. (Cited on pages 13 and 94.)

[Zhou 2010] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candès and Yi Ma. *Stable Principal Component Pursuit*. In Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on, pages 1518–1522, 2010. (Cited on page 30.)

Javier Cruz Mota

Avenue de la Gare 6
CH-1028 Préverenges

Tel.: +41 76 458 92 72
E-Mail: me@javiercruz.com
Web: http://www.javiercruz.com/



Education

PhD in Electrical Engineering (Ecole Polytechnique Fédérale de Lausanne – EPFL) **2007 – 2011**
Co-directed PhD Thesis between the TRANSP-OR Laboratory and the LTS5 Laboratory

Erasmus grant to develop my **Master Thesis** in Telecommunications Engineering **2006**
Developed at the Signal Processing Laboratory (LTS5) of EPFL

MS in Telecommunications Engineering (Universitat Politècnica de Catalunya – UPC) **2000 – 2006**

MS in Mathematics (UPC) **2000 – 2005**

Professional Experience

Transp-OR – EPFL (Research and Teaching Assistant) **2007 – Present**

Working on several research projects related to visual tracking, omnidirectional image processing and mathematical modelling and simulation. Teaching assistant for the course "Modelling of Energy and Transport Systems" of the Master in Civil Engineering.

TSC – UPC (Research Engineer) **2006**

In charge of the implementation of the facial detection and recognition module for the CHIL smart room at UPC. CHIL (Computers in the Human Interaction Loop) is a 24 million € project financed by the European Commission, where the UPC and its Signal Theory and Communications department (TSC) participate.

IRI – CSIC (Student Research Assistant) **2004 – 2005**

Successful design and implementation in MatLab of a solver for the inverse kinematic problem of spherical and planar robotic mechanisms. These types of mechanisms are for instance used in surgical applications. The Institut de Robòtica i Informàtica Industrial (IRI) is a joint research centre of the UPC and the Spanish Council for Scientific Research (CSIC).

Technical Skills

Image and Signal Processing:

Visual tracking; particle filtering; Fourier analysis and transforms in Euclidean and non-Euclidean spaces; image and video segmentation; pattern recognition; wavelets; mathematical modelling: principal component analysis (PCA), discrete choice modelling, time series, neural networks; spatial diversity processing: beamforming, direction of arrival estimation (DOA), smart antennas, MIMO.

IT:

C, C++, OpenCV, MatLab, Linux expert, Assembler, multithreaded programming, PHP, Qt, BASH Shell scripting, Java, Maple, R, VHDL, PSpice/OrCAD, Puff, LabView, Biogeme, SQL, TCP/IP networking, LaTeX.

Electronics:

Analog and digital circuit analysis and design, FPGA, microwave analysis, measurements of circuits at low and high frequencies (oscilloscope, logic analyser, spectrum analyser, vector network analyser, time domain reflectometer), antennas, circuits for RF signal processing, RF IC design, CMOS for RF circuits.

Languages

- **Spanish** and **Catalan**: mother tongues.
- **French**: advanced in writing and fluent in speaking.
- **English**: fluent in writing and advanced in speaking. Four weeks intensive course in Dublin in 2005.
- **German**: complete beginner.

Personal Information

I was born in Barcelona (Spain) the 12th of May of 1982. I have the Spanish citizenship and a B Swiss working permit since January 2007. I am a meteorology and astronomy lover, and I enjoy also running, skiing and programming free software under the GPL license.

Publications

Journal Papers:

- T. Robin, M. Bierlaire and J. Cruz-Mota (to appear). Dynamic facial expression recognition with a discrete choice model, *Journal of Choice Modelling* (accepted for publication on June 2011).
- M. Sorci, G. Antonini, J. Cruz-Mota, T. Robin, M. Bierlaire and J.-Ph. Thiran (2010). Modelling human perception of static facial expressions, *Image and Vision Computing* 28(5):790-806. DOI: 10.1016/j.imavis.2009.10.003
- T. Robin, G. Antonini, M. Bierlaire and J. Cruz-Mota (2009). Specification, estimation and validation of a pedestrian walking behaviour model, *Transportation Research Part B: Methodological* 43(1):36-56. DOI: 10.1016/j.trb.2008.06.010

Book Chapters:

- M. Sorci, T. Robin, J. Cruz-Mota, M. Bierlaire, J.-Ph. Thiran and G. Antonini (2010). Capturing human perception of facial expressions by discrete choice modelling. In S. Hess and A. Daly (ed) *Choice Modelling: The State-of-the-Art and the State-of-Practice* (ISBN: 978-1-84950-772-1) pp. 101-136. Emerald Group Publishing Limited.

Conference Papers:

- T. Robin, M. Bierlaire and J. Cruz-Mota (2010). Travellers well-being measuring and dynamic facial expression recognition. *Proceedings of the World Conference on Transport Research*, 2010.
- T. Robin, M. Bierlaire and J. Cruz-Mota (2009). Dynamic facial expression recognition using a behavioural model. *Proceedings of the 9th Swiss Transport Research Conference (STRC)*, 2009.
- T. Robin, G. Antonini, M. Bierlaire and J. Cruz-Mota (2008). Specification, estimation and validation of a pedestrian walking behaviour model. *Proceedings of the European Transportation Research Conference (ETC)*, 2008.
- J. Meynet, T. Arsan, J. Cruz and J.-Ph. Thiran (2008). Fast multi-view face tracking with pose estimation. *Proceedings of the 16th European Signal Processing Conference (EUSIPCO)*, 2008.
- M. Sorci, G. Antonini, B. Cerretani, J. Cruz-Mota, T. Robin, M. Bierlaire and J.-Ph. Thiran (2008). Modelling human perception of static facial expressions. *Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2008.

Technical Reports:

- T. Robin, M. Bierlaire and J. Cruz-Mota (2010). Dynamic facial expression recognition with a discrete choice model. *Technical report TRANSP-OR 100423*. Transport and Mobility Laboratory, ENAC, EPFL.
- M. Sorci, T. Robin, J. Cruz-Mota, M. Bierlaire and J.-Ph. Thiran (2009). Modelling human perception of facial expressions. *Technical report TRANSP-OR 090706*. Transport and Mobility Laboratory, ENAC, EPFL.
- J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire and J.-Ph. Thiran (2009). Scale invariant feature transform on the sphere: theory and applications. *Technical report TRANSP-OR 090426*. Transport and Mobility Laboratory, ENAC, EPFL.
- J. Cruz-Mota, M. Bierlaire and J.-Ph. Thiran (2008). Multiple-view scenes: reconstruction and virtual views. *Technical report TRANSP-OR 080710*. Transport and Mobility Laboratory, ENAC, EPFL.
- T. Robin, G. Antonini, M. Bierlaire and J. Cruz-Mota (2007). Specification, estimation and validation of a pedestrian walking behaviour model. *Technical report TRANSP-OR 071116*. Transport and Mobility Laboratory, ENAC, EPFL.