FACULTY OF ENGINEERING OF THE UNIVERSITY OF PORTO

Visual Motion Analysis based on a Robotic Moving System

Andry Maykol Gomes Pinto

A thesis submitted for the degree of
Doctor of Philosophy in Electrical and Computer Engineering

Doctoral Program in Electrical and Computer Engineering

Advisor: Prof. António Paulo Gomes Mendes Moreira (Ph.D.)

Co-advisor: Prof. Paulo José Cerqueira Gomes da Costa (Ph.D.)

August 5, 2014

Abstract

Motion analysis is one of the most relevant areas under discussion in robotics and computer vision. Extracting high level information from the environment increases the ability of a robot to detect, identify and track the individual elements of the scene. There are several methods in the literature of vision computing that perform motion analysis with good results and for a variety of environments. However, most of these methods cannot overcome the real-time constraints that are imposed by robotic applications since they commonly take more than a few of seconds to compute. Therefore, it is crucial to overcome certain problems related with the perception and the interpretation of dynamic scenes. Vision computing is a challenging research field for small mobile robots because of their reduced computational capability (enhancing the autonomy) and limited size. In this context, visual techniques for robotic solutions are computationally more efficient than techniques for other application fields, although the improvement is usually done at the expense of using images with lower resolution and feature-based approaches.

The perception of motion can be divided into the detection, measurement and cognitive stages. Moreover, it can be performed using two distinct ways which are directly related to the movement of the observer that is capturing the scene: stationary observation or moving observation. The perception of motion based on static observers is a relatively easier problem comparatively to the perception based on moving observers because every spatial and temporal variation represents the moving objects, by neglecting illumination changes and noise. On the contrary, the observer's motion creates new paradigms that make the analysis even more complex and challenging. Nevertheless, this research discusses motion analysis based on dense optical flow fields for a new generation of robotic moving systems with real-time constraints. It focuses on a surveillance scenario where an especially designed autonomous mobile robot uses a monocular camera for perceiving motion in the environment. This mobile robot moves along a rail which enhances the surveillance capability when compared to conventional systems, mainly composed by multiple static cameras. The perception architecture of the robot includes two operating modes that are triggered according to the vehicle's motion: static perception and dynamic perception.

Scientifically, this thesis is focused in motion analysis for moving observers. It presents a spatiotemporal filter, a method to compute dense flow fields and several techniques to interpret and retrieve motion information from flow fields. The novel filtering technique is named Robust BilateraL and Temporal (RBLT) and reformulates the conventional bilateral filter. The filter relies to a spatial and temporal evolution of sequences to conduct the denoising process while preserving relevant image information. A pixel value is estimated using a robust combination of spatial characteristics of the pixel's neighborhood

and its own temporal evolution. Thus, robust statics concepts and the temporal correlation between consecutive images are incorporated which results in a reliable and configurable filter that reconstructs highly dynamic and degraded image sequences.

Furthermore, an innovative and efficient dense optical flow architecture is proposed. The designed technique captures and combines the advantages of the local and global differential optical flow methods with a hierarchical and tree-based structure, achieving a surprising balance between computational effort and flow performance. This HybridTree method is able to identify the intrinsic nature of the motion since descriptive properties of the image are retrieved and used to divide the image into regions that may have different motions. These properties are integrated in a hybrid and hierarchical optical flow structure to estimate the flow field.

The analysis of motion is conducted in this thesis by two novel techniques, namely, the Hybrid Hierarchical Optical Flow Segmentation (HHOFS) and the Wise Optical Flow Segmentation (WOFS). The first method is able to extract the moving objects from dense flow fields by performing two consecutive operations: refining and collecting. The flow field is decomposed in a set of clusters during the refining phase and descriptive motion properties are used in the collecting stage by a hierarchical scheme to merge the set of clusters that represent different motion models. The second method extracts the moving objects by performing an evaluating and resetting phase. Descriptive motion properties of the flow field are retrieved in the evaluation phase and using the HHOFS, which provides high level information on the spatial segmentation of the flow field. In the resetting operation, this information is used by a watershed-based approach to enhance the resulting clusters. In addition, this thesis presents a novel method that extracts information about the number of moving objects using the polar representation of dense optical flow fields. The model selection method is a Bayesian approach that balances the model's fitness and complexity since it combines the correlation of a histogram-based analysis with the decay ratio of the normalized entropy criterion.

The research evaluates the performance achieved by the methods in a realistic surveillance situation. Extensive experiments considering videos obtained by a mobile robot equipped with a monocular camera show that the proposed techniques achieve a good perceptual quality of filtering sequences corrupted with a strong noise component enable a fast estimation of dense flow fields and produce an efficient segmentation of regions with different types of motion. The experiments show that the methods extract reliable motion information in real-time and without using specialized computers. Moreover, the resulting analysis of motion is less computationally demanding compared to other recent methods which mean that it is suitable for most of the robotic and surveillance applications. Therefore, the proposed architecture for dynamic motion perception is capable of measuring external motions and provides relevant information that can be used to detect and track danger situations, for instance, intrusions, unrecognized objects, or even, for access control and identification of people.

Keywords: Visual Perception, Motion Analysis, Moving Observations, and Mobile Robotics.

Resumo

A análise de movimento é um dos assuntos mais discutidos hoje em dia nas áreas científicas ligadas à robótica e visão computacional. O processo de aquisição de informação considerada de alto nível num ambiente vulgar permite melhorar a capacidade de um robô em detectar, identificar e seguir os diversos elementos que constituem o cenário envolvente. Existem alguns algoritmos e métodos de visão computacional que permitem uma análise de movimento em diversos ambientes e com resultados bastante aceitáveis. Todavia, a maioria desses trabalhos científicos não contemplam os requisitos de tempo-real que são vulgarmente impostos pelas aplicações robóticas. Desta forma, é fundamental o desenvolvimento de novas técnicas de visão computacional que permitam ao robô interpretar os ambientes dinâmicos. Trata-se de uma área de investigação que apresenta muitos desafios para as aplicações robóticas de menores dimensões, pois esses sistemas são caracterizados por uma reduzida capacidade de processamento que visa melhorar a sua autonomia. Neste contexto, as técnicas de visão computacional mais orientadas à robótica limitam-se normalmente a imagens de pequena resolução e à extracção de características.

A percepção de movimento é vulgarmente dividida em três fases: detecção, medição e interpretação. Não obstante, ela pode ser obtida de duas formas distintas e relacionadas com o movimento do observador: observações estacionárias ou em movimento. A percepção de movimento através de observações estacionárias é um problema relativamente acessível, pois é assumida que toda a variação temporal e espacial de um pixel é consequência do movimento (desprezando variações de luminosidade e ruído). Em contrário, o movimento do observador origina novos paradigmas que tornam a análise de movimento bem mais complexa. Esta tese discute a análise de movimento baseado em campos de fluxo óptico e para uma nova geração de robôs (com requisitos de tempo-real).

A tese foca-se num cenário de vigilância onde um robô móvel e autónomo recorre a visão monocular de forma a compreender o movimento no ambiente. Esse robô move-se ao longo de um carril, o que melhora o alcance do sistema de vigilância quando comparado com os sistemas mais convencionais (múltiplas cameras). A arquitectura de percepção do robô inclui dois modos de operação que são activados de acordo com o movimento do veículo: percepção estática e dinâmica.

Cientificamente, esta tese está orientada para a análise de movimento com observadores móveis (percepção dinâmica) e, por isso, ela propõe um filtro com características espaço-temporais, um método de cálculo de campos de fluxo óptico e diversas técnicas que interpretam a informação de movimento. O filtro, cujo nome é "Robust BilateraL and Temporal" (RBLT), consiste numa reformulação do filtro bilateral. O filtro RBLT consegue preservar a informação mais relevante das imagens, isto porque o valor de cada pixel é estimado utilizando métodos estatísticos robustos que combinam as caracterís-

ticas espaciais dos pixéis vizinhos com a evolução temporal do valor do próprio pixel. Desta maneira, o filtro é capaz de reconstruir sequências de imagens dinâmicas e muito degradadas.

Para além do filtro, a tese apresenta uma técnica inovadora do cálculo de campos de fluxo óptico. Esta técnica captura e combina as vantagens dos métodos diferenciais (locais e globais), numa arquitectura hierárquica e baseada em uma estrutura de árvore. O método extrai propriedades da imagem que são utilizadas para guiar o cálculo do fluxo óptico e que identificam a natureza intrínseca do movimento patente na sequência. Esta arquitectura consegue alcançar um bom compromisso entre a qualidade do fluxo óptico e o custo computacional.

A análise de movimento é realizada principalmente através de dois métodos, o "Hybrid Hierarchical Optical Flow Segmentation" (HHOFS) e o "Wise Optical Flow Segmentation" (WOFS). O primeiro método extrai o movimento a partir de campos de fluxo óptico e através de uma decomposição espacial de cada campo num conjunto de regiões, com base nas propriedades inerentes a movimentos distintos. Estas regiões são posteriormente utilizadas por uma fase de junção hierárquica. O segundo método, assume o resultado proveniente do HHOFS (considerada como informação de alto nível) para guiar o processo de aperfeiçoamento do movimento aparente dos objectos. Este aperfeiçoamento é obtido com recurso às depressões que definem os contornos do movimento. O número de objectos que se deslocam nos campos de fluxo óptico é estimado através de um método de selecção de modelo que recorre à formulação Bayesiana para conjugar a complexidade e a adequação do modelo de inferência aos dados. Ele recorre à integração da análise dos histogramas de magnitude e ângulo do fluxo óptico com a relação de decréscimo do critério de entropia normalizada.

A performance de todos os métodos propostos é avaliada no cenário e contexto desta tese. Os resultados demostram que as técnicas apresentam uma boa performance em termos de qualidade perceptual da reconstrução de imagens degradadas por uma forte componente de ruído, uma rápida estimação de campos de fluxo óptico, e uma eficiente interpretação e separação dos objectos em movimento. Portanto, os métodos apresentados proporcionam uma análise de movimento bastante fiável, que é conseguida em tempo-real e sem recurso a unidades computacionalmente especializadas. As técnicas propostas nesta tese possuem um requisito computacional bem menor quando comparadas com outras técnicas da literatura e, portanto, são mais indicadas para aplicações robóticas baseadas em visão computacional. Os resultados mostram ainda que a arquitectura de análise de movimento que é utilizada nesta tese é capaz de medir movimentos externos e fornecer informações relevantes para a detecção e seguimento de situações perigosas, por exemplo, intrusões, reconhecimento de objectos, ou até mesmo, para controlo de acessos e identificação de pessoas.

Palavras chave: Percepção Visual, Análise de Movimento, Observações em Movimento e Robótica Móvel.

Acknowledgements

This work became possible thanks to the support of different people and organizations. First, I would like to thank my supervisors, Professor A. Paulo Moreira and Professor Paulo G. Costa by giving me intellectual freedom in my work, supporting my research over the last four years, engaging me in new ideas, and demanding a high quality of work.

I would like to thank Centre for Robotics and Intelligent Systems (CROB) of INESC TEC since it provided a friendly atmosphere at work and the resources that made possible all the achievements of this work. In particular, to Fernando Guedes for the availability and support in the design and technical development of the mobile robot. Besides my advisors and research group, I would like to express my sincere gratitude to the Scientific Committee of the Doctoral Program in Electrical and Computer Engineering who believed in my work since the very first beginning; and to all Professors of the Department of Electrical and Computer Engineering of the Faculty of Engineering of the University of Porto who contributed to my constant pursue for new and unsolved challenges. Especially, to Professor Miguel V. Correia and Professor Jaime S. Cardoso for all the kindness that they have demonstrated during the insightful discussion of many aspects of this research.

Last but not least, I would like to acknowledge my beautiful parents and sister. Thank you so much for pushing me to pursue my dreams, I would never be where I am today without your constant love, support and guidance; thanks to Mónica Sofia who endured the unpleasant effects of all my dedication. Thank you for your constant love.

Thank you so much to everyone that have contributed in a direct or indirect manner for the success of this work.

Andry Maykol Pinto,
April, 2014

Official Acknowledgements

Andry Maykol Pinto acknowledges the Portuguese Government through the FCT (Foundation for Science and Technology) for his Ph.D. grant SFRH-BD-70752-2010, without which this research work would not have been possible.

This work is also financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT within project FCOMP - 01-0124-FEDER-022701.

"If you really want something, really work hard, take advantage of opportunities, and never give up... You will find a way."

Jane Goodall

Contents

List of Figures

List of Tables

Nomenclature

2D Two-dimensional

2DGSM 2D Gaussian Scale Mixture

3D Three-dimensional

3DFD 3D Fuzzy Directional

3DM 3D Median

3DVM 3D Vector Median

AAE Average Angular Error

AE Angular Error

AIC Akaike Information Criterion

ATM α-Trimmed Mean

AVTM Adaptive Vector directional α-Trimmed Median

BFHE Bayesian Fusion of Histogram and Entropy

BIC Bayesian Information Criterion

BL Bilateral Filter

blob Binary Large OBject

CCTV Closed-Circuit TeleVision

CLG Combined Local-Global

CML Classification Log-likelihood

CPU Central Processing Unit

CRF Conditional Random Field

DBSCAN Density-Based Spatial Clustering of Applications with Noise

EM Expectation-Maximization

EPE EndPoint Error

FAST Features from Accelerated Segment Test

FFT Fast Fourier Transformation

GMHMCF Generalized Multi-Hypothesis Motion Compensated Filter

GMM Gaussian Mixture Model

GPU Graphics Processing Unit

GVD Generalized Vector Directional

H.O.T. High Order Terms

HD High Definition

HDBOFS Hybrid Density-Based Optical Flow Segmentation

HHOFS Hybrid Hierarchical Optical Flow Segmentation

HQIC Hannan-Quinn Criterion

HS Horn-Schunck

HSV Hue, Saturation and Value (color model)

HY HybridTree

Hz hertz

IEEE Institute of Electrical and Electronics Engineers

IFSM Inter-Frame Statistical Modeling

KMNN K-Means Nearest Neighbor

LK Lucas-Kanade

LSQ Least SQuares

MAP Maximum A Posteriori

MLE Maximum Likelihood Estimator

MOb Moving Observation

MOG Mixture of Gaussians

MRF Markov Random Field

MSE Mean Squared Error

MTMP Multi-Tracking of Motion Profiles

NEC Normalized Entropy Criterion

NLM Non-Local Means

PETS Performance Evaluation of Tracking and Surveillance

PSNR Peak Signal-to-Noise Ratio

QCIF Quarter Common Intermediate Format

RANSAC RANdom SAmple Consensus

RBLT Robust Bilateral and Temporal Filter

RGB Red, Green and Blue (additive color model)

RMSE Root Mean Square Error

SAD Summation of Absolute Difference

SIFT Scale Invariant Feature Transform

SNR Signal-to-Noise Ratio

SOb Stationary Observation

SSD Sum of Squared Difference

SSIM Structural Similarity

STGSM SpatioTemporal Gaussian Scale Mixture Model

SURF Speeded Up Robust Feature

UAV Unmanned Aerial Vehicle

VBM3D Vector Block Matching and 3D filtering

VDKNNVM Vector Directional K-Nearest Neighbor with Vector Median

WOFS Wise Optical Flow Segmentation

WRSTF Wavelet-domain Reliability-based SpatioTemporal Filtering

Chapter 1

Introduction

The growing interest in robots is one of the main reasons to research for new and improved robotics applications capable of interacting with the dynamic elements of domestic environments. Automation in areas like, security, surveillance, equipment transportation, pharmaceutical industry, domestic cleaning and guiding visitors in museums, laboratories or shopping's, are currently increasing the demand for autonomous robotics applications [1]. The new generation of indoors service mobile robots must be able to conduct activities beyond those related to leisure, such as, helping people in their workplace and home. For the development of such applications, it is crucial to overcome certain problems related to perception and interpretation of dynamic scenes. The extraction of high level information increases the robots' ability to perform motion detection, tracking, object recognition and navigation. It is also imperative to increase the ability to interact with the environment, meaning that the robot must be able to detect and analyze its surrounding scene.

Densely populated environments are difficult for the navigation of mobile robots due to several factors that must be considered: the location, the speed and the trajectory of the robot, and everything else: people and other obstacles. Nowadays, there are visual techniques [7, 8, 9] for service mobile robots that make possible to carry out a reasonable navigation. However, most of these techniques are not proficient enough when the robot is subjected to realistic indoor environments since they do not extract relevant information related to the nature of dynamic interactions. Mobile robots are continually stimulated in those environments with: moving doors, humans and animals. These elements must be detected and interpreted otherwise; it will be hard to assure a safe and intelligent interaction of the robot in the environment.

[1] "World Robotics 2012 - Service Robots" from the International Federation of Robotics.

Vision-based technologies are non-invasive and acquire information in a similar manner as the human vision, which make them appealing for non-industrial robotics applications. Computer vision is a scientific field that is currently devoted to extracting and understanding high-level information of scenes. In this field, motion perception is one of the most relevant areas, and there are several models and methods to perform motion analysis in a variety of environments. It includes several application areas, namely, video surveillance, biometrics, medicine, augmented reality, gaming, automotive and movie production.

The visual perception of motion can be divided into three stages [10]: *detection*, *measurement* and *cognitive* stages. Changes in brightness, texture, color and shapes are used to identify regions of interest that might represent some motion during the detection stage. The measurement phase is responsible for evaluating the intrinsic motion parameters of elements belonging to the scene. Finally, the cognitive phase classifies the type of motion according to features and based on information that is obtained in previous stages.

Motion perception can be performed using two distinct ways which are directly related with the movement of the observer that is capturing the scene: *stationary observation* or *moving observation*. Motion perception with a static observer is quite different from the moving observer because every spatial and temporal variation represents the moving object when a static observer captures the scene, by neglecting illumination changes and noise. The pattern of motion exhibits variations in almost every pixel when it is obtained from a moving observer. These variations depend on the external moving objects as well as the relative motion between the visual sensor and the scene. Therefore, the two approaches have specific theoretical assumptions that cause significant differences in performance, flexibility and robustness for techniques of motion detection and analysis. Several approaches are being studied by the scientific community (see chapter 2); however, techniques based on moving observations are still in a preliminary stage when compared to static observations. This is a direct consequence of the egomotion (motion of the observer) because it creates new paradigms that turn the study of motion even more complex and challenging.

1.1 Motivation

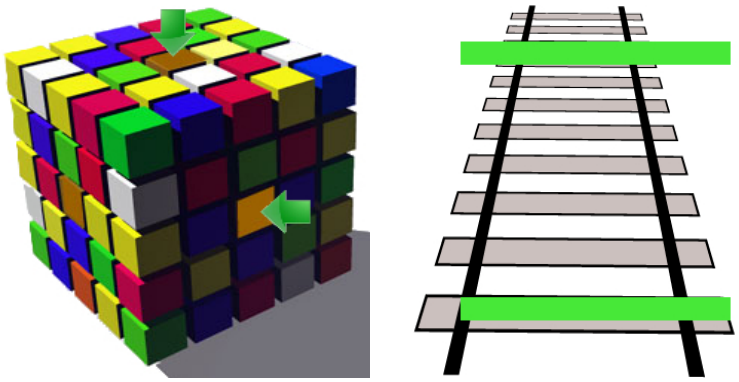
Human being have an extraordinary capability for motion perception[2] due to its remarkable visual sensing system since it makes possible to perceive, distinguish and characterize the different moving elements of the environment. The visual sensing system is

[2]Is the process of inferring about the intrinsic motion features of elements in a scene and based on the visual field.

consisted by both eyes as sensory receptors, the neural pathway and the visual cortex located in the occipital lobe.

The receptors provide clues to the brain that uses a limited amount of information to build a thrust word reality. Eyes convert the light reflected by the object into an electrical signal that is sent through the optical nerve and to the visual cortex. The visual cortex is responsible for crossing reference of each image with the memory of past experiences that was stored in the brain. Furthermore, the brain tries to identify the object and decides how it is positioned in space.

One of the most reliable ways of doing it is to use shadows created by the interaction of the source of light and the object's shape. The brain has learned to trust in shadows as a near foolproof way to know the behavior of objects in space. However, color is another reliable source of information about the world but it can be faulty in some cases. An example is depicted in Fig. 1.1(a), where the human perception tells that both squares (pointed out by green arrows) have different colors. In reality, both squares have the same color. The illusion is originated by the shadow since it contextualizes the right side of the cube as being darker than the top side and, thus, the brain thinks that the color of the right square should be lighter than the top square: this causes a misinterpretation of the color. Although, color is a helpful feature to separate different objects.



(a) (b)

Figure 1.1: 1.1(a) - Human perception tells that the color of the center square on the top side is different from the color of the center square on the right side because both sides have different shadow contexts. 1.1(b) - An illusion caused by the perspective of the rail tracks. It seems that green rectangles have different sizes.

The brain gets the 2D information from the eyes and creates a 3D reality. The visual cortex uses familiar patterns as shortcuts to understand the environment. There are several informative features that the brain tries to put together, for instance, depth and perspective. The 3D reality is created so instantaneously that the brain makes assumptions

about the surrounding environment. In some cases, these assumptions are not accurate, for example, the green rectangle on the top of Fig. 1.1(b) looks bigger than the bottom rectangle because the brain is making the assumption that the first one is placed far away on the train tracks and, thus, its size is overcompensated.

Human's eyes retrieve information that is processed in the visual cortex that tells how far things are and how quickly they are moving. In this way, the visual system extracts information through sensory experience and conducts reliable judgments based on intrinsic motion features, namely, location, direction, trajectory, magnitude, colors, boundary and shape. Perspective features are processed so effectively by the brain that they are crucial for helping humans to determine the moving elements of a scenario. The brain quickly finds these elements by merging the components of their individual motion into a seamless moving reality. Specific neurons throughout the visual cortex track the changing positions of objects between images, making possible to comprehend the visual changes as motion. In fact, motion perception plays an important role on human daily interactions, for instance, to communicate with other humans and to drive or walk in a street. Inferring about the direction and the speed of moving objects are the most critical skills in visual perception since they can detect situations of danger.

At other hand, mobile robotics applications have certain problems related to visual perception and interpretation of the dynamic scene: unmanned aerial vehicles [11, 12], unmanned surface vehicles [13] and unmanned ground vehicles [14, 15]. In these applications, suitable motion detection is crucial to feed the high level processes with relevant information. Autonomous mobile robots are mostly employed for transporting parts inside production plants and for autonomous specialized missions. Currently, several navigation procedures enable the reliable movement of such robots; however, the perception abilities are constraining the applications to a low or medium level of interaction.

The ability to interpret, understand and interact with dynamic scenes is crucial for a new generation of indoor robots. Extending their operating scope implies a safe interaction through unknown environments. Therefore, better visual computing algorithms and new behavioral laws must be developed to increase the autonomy of mobile robots.

In particular, vision-based sensing is a passive and non-intrusive technology because does not require the modification of the environment. However, the visual measurements must be analyzed and processed to make possible the extraction of clues and the recognition of movements. In computer vision, images are discrete and temporal samples of the spatial environment being photographed. These images reflect several characteristics related with the direction of view, the spatial position, the color, the illumination level and the depth of the scene [16]. Motion analysis based on image sequences is present in

various applications like, video compression, object tracking, object recognition, three-dimension reconstruction and video segmentation.

The critical nature of visual perception turns motion detection and analysis for mobile robots as one of the most relevant areas discussed in the literature; existing several models and methods to perform motion analysis in a variety of environments. Vision computing is a challenging research field for small mobile robots because of the vehicle itself. The first issue is related to the limited space that is available onboard for the deployment of computer units and sensors. Other issue is related to the power consumption that enforces the autonomy of such robots. This limits the computational capability which has a direct influence in the performance of the navigation and sensing procedures. At first sight, it may seems a little strange all this effort because humans instinctively predict the direction and the velocity of moving objects. However, current computational methods do not solve this problem in an effective and robust manner, being usually conditioned by lighting changes, shadows, vibrations, occlusions and noise.

Motion detection and analysis from a stationary observer is a non-trivial research area; however, it has been explored extensively and a preliminary success has been achieved in tracking features, segmenting moving objects and pixel-wise flow based on static vision systems [17]. Currently, there is a wide diversity of motion perception methods which is justified by the inherent complexity of the problem. Unfortunately, the same cannot be said for motion perception based on moving cameras because the movement of the observer creates two independent motion components: the egomotion and the objects. The most conventional techniques for motion perception consider that visual changes are caused only by the movement of the external objects since they assume the stationary position of the observer. Therefore, they fail almost completely when the dynamic scene is captured by a non-static observer due to their inability to distinguish both motion components. The visual detection of motion from a moving observer is the most often encountered case in real life situations [18]. It is a complex and challenging problem, although, it can promote the arising of new applications.

The majority of related works about motion detection and analysis relies on fixed cameras; however, the research presented in this thesis goes one step further by discussing motion detection, measurement and understanding for a new generation of mobile robotic systems. In this context, the thesis aims to study visual motion perception based on moving observations, *i.e.*, *how the robot can extract and analyze motion of different objects using onboard visual sensing?*

This research focuses on a surveillance scenario where a mobile robot conducts its activity autonomously. Implications of this research can lead to innovations in different fields, such as, computer vision, robotics, security and surveillance.

1.2 Objectives

The main goal of this work is the development of techniques for a reliable detection and analysis of motion. It intends to increase the ability, intelligence and autonomy of an innovative mobile robot that is designed for active surveillance. The robot conducts an intelligent surveillance activity and acquires information about the environment using a monocular camera and odometry. The robot can be applied to other contexts besides the surveillance of domestic environments, such as, quality control, inspection, tree-dimensional modeling, sports and supermarkets.

In particular, this researching work is focused in four objectives:

- Presents an innovative and flexible robotic application for surveillance of indoor environments. The robot is centered in perception of motion for a realistic environment. The robotic platform and the surveillance scenario contextualizes the qualitative and the quantitative evaluation of the techniques that are proposed in this research;

- Proposes novel representations and architectures of motion perception for moving observers: increases the autonomy of the mobile robot under realistic working conditions because the techniques make possible the interpretation and the recognition of different type of motions;

- Expands the frontier of motion perception to beyond the conventional feature-based methods. Unlike the overwhelming majority of researching works, this thesis study the three phases of motion perception (detection, measurement and cognitive);

- Explores new applications - the scientific advances resulted from this thesis can lead to new applications that traditional algorithms do not allow. Inspired by the increased awareness of security issues, new surveillance applications must be developed. These systems must be able to analyze actions, behaviors and activities of a single individual or crowds, and will be used for recognize people, guide persons in complex facilities, to study the psychological aspects of crowds in shopping's or even to detect abnormal activities.

Motion perception for moving observers represents one of the most challenging areas in computer vision and robotics. This research is centered in motion perception for a realistic and practical surveillance scenario that contextualizes the evaluation and comparison of the performance achieved by the proposed algorithms with other state-of-the-art methods.

1.3 Thesis overview

This document is organized as follows:

Chapter 2 demonstrates an overview about motion perception and analysis. A short description of some works related to motion perception based on static observation is provided in section 2.2. Subsequently, the problem of motion detection for moving observers is described with detail in section 2.3: the influence of the observer's movement is mathematically analyzed and contributions to the state-of-the-art are also presented.

Chapter 3 shows a mobile robot that represents a new generation of surveillance systems [19]. As expected, this research focuses on a surveillance scenario where an especially designed autonomous mobile robot uses a monocular camera to perceive motion. The chapter presents the hardware and the distributed software of the robotic platform in section 3.2. In addition, section 3.3 depicts an architecture for visual motion perception that is formed by two modes: static perception and dynamic perception. This research is focused in studying the complex problem of dynamic perception and, therefore algorithms of motion perception for moving observers are presented in the following chapters.

Chapter 4 presents a video denoising technique, called Robust Bilateral and Temporal Filter (RBLT) [20], that satisfies the visual requirements of surveillance applications based on mobile robots. The technique resorts to spatial and the temporal evolution of sequences to conduct the filtering process while preserving relevant image information. A pixel value is estimated using a robust combination between the spatial characteristics of the pixel's neighborhood and its own temporal evolution, see section 4.2. Thus, robust statics concepts and temporal correlation between consecutive images are incorporated together which results in a reliable and configurable filter formulation that makes it possible to reconstruct highly dynamic and degraded image sequences.

Chapter 5 proposes a dense optical flow technique called, the *HybridTree* optical flow [4]. The proposed technique mimics the human motion detection based on different layers of visual details. It has two major and distinct phases: *expectation* and *sensing*. The computational requirement is an important aspect for today's applications and, for that reason, the section 5.2 aims to take advantage from the most relevant and efficient improvements of the optical flow techniques to create a balance between real-time capability and performance. The approach presented differs from other

techniques since it introduces a new perspective for optical flow computation: high level information about the image sequence is integrated into the estimation of the optical flow. The resulting flow field is satisfactory comparatively to other state-of-the-art methods. In addition, the proposed technique is more computationally efficient than other approaches, such as, CLG (combined local-global) method [21], because high level information on the image is gathered and used by a smart combination of local and global differential techniques. Combining local and global differential methods in section 5.3 proved to be beneficial as confirmed in [21]. Efficient methods were used in this research to demonstrate the simple and yet powerful concept of the *HybridTree* method, namely, modern versions of the Lucas-Kanade and Horn-Schunck.

Chapter 6 proposes two major techniques for motion analysis that measure and extract distinct motion models from dense optical flow fields: the Hybrid Hierarchical Optical Flow Segmentation (HHOFS) [22] and the Wise Optical Flow Segmentation (WOFS). The techniques were developed for the especially designed mobile robot that performs an intelligent surveillance. The major advantage of these techniques is the ability to segment in real-time the different types of motion in image sequences. These techniques were compared to standard baseline clustering methods, namely, Expectation-Maximization (EM) and K-means, and the results confirm that the proposed techniques are suitable for other robotics applications. In addition, this chapter proposes a model selection method to estimate the number of motion models from flow fields. The technique, named Bayesian Fusion of Histogram and Entropy (BFHE), combines a histogram-based approach with cost functions (that balance fitness and model complexity). Thus, the estimation of the number of clusters can be incorporated with parametric techniques that require information about the number of clusters in the data, for instance, the EM and K-means.

Chapter 7 provides the major conclusions that can be taken from this thesis. A final discussion is conducted in section 7.1: the novelty of the work and its achievements. Moreover, a set of guide-lines for future works is addressed in section 7.2.

1.4 Contributions

1.4.1 The achievements

The more relevant achievements of this thesis are described below:

1. Promoting efficient techniques of motion perception for robotic and surveillance applications;
2. A novel robotic application for active surveillance, called *EEyeRobot* . This robot is able to autonomously detect external motions while moving and using a monocular visual system;
3. An architecture for motion perception and analysis: it addresses the problem of motion perception for moving observers;
4. An innovative spatiotemporal filtering technique: Robust Bilateral and Temporal (RBLT), that can be used by stationary and non-stationary surveillance or robotics applications;
5. A filtering technique with a performance less influenced by outliers and less influenced by the type of noise that corrupts the sequence. The RBLT has a temporal filtering component based on the *temporal coherence* assumption with a self-evaluation mechanism to detect and treat violations of this assumption;
6. Filtering with a better trade-off between noise reduction and data preservation which is, especially, recommended for denoising images with low SNR (signal-to-noise ratio). Thus, the RBLT filter does not create ghosts or strange artifacts in the denoised image that compromise processes of motion analysis;
7. Promoting efficient optical flow techniques for robotic and surveillance applications;
8. An assisted optical flow estimator: *HybridTree* method, that combines local and global differential methods using cognitive information;
9. A hierarchical method to guide the flow estimation process of the *HybridTree*, enabling an optical flow enhancement while preserving the computational time requirements;
10. An efficient method to decompose the image into exclusive regions based on similarity properties: temporal differencing, texture, brightness and color;
11. A study of motion perception and analysis based on dense optical flow fields and moving observers;
12. A novel representation and architecture for motion analysis based on moving observations depicted by dense flow fields: the Hybrid Hierarchical Segmentation and the Hybrid Density-Based Segmentation;

13. An assisted technique for segmenting dense flow fields, called Wise Optical Flow Segmentation, that automatically extracts and combines cognitive information about distinct motions. This guided-based clustering technique enhances the edges of the moving objects (contours) and preserves the computational time requirements of robotics applications;

14. A model selection method to enhance the performance of the segmentation techniques, called Bayesian Fusion of Histogram and Entropy;

15. An extensive qualitative and quantitative evaluation under realistic working conditions of all techniques proposed in the thesis.

1.4.2 The publications

Publications related to this research include:

- [22] Andry Maykol Pinto, Miguel V. Correia, A. Paulo Moreira, and Paulo G. Costa. "Unsupervised Flow-based Motion Analysis for an Autonomous Moving System", *Image and Vision Computing* (Elsevier), 22(6-7):391-404, 2014;

- [20] Andry Maykol Pinto, Paulo G. Costa, Miguel V. Correia, and A. Paulo Moreira. "Enhancing dynamic videos for surveillance and robotic applications: The robust bilateral and temporal filter", *Signal Processing: Image Communication* (Elsevier), 29(1):80-95, 2014;

- [4] Andry Maykol Pinto, A. Paulo Moreira, Miguel V. Correia, and Paulo G. Costa. "A Flow-based Motion Perception Technique for an Autonomous Robot System", *Journal of Intelligent and Robotic Systems* (Springer), in press, 2013, doi:10.1007/s10846-013-9999-z;

- [23] Andry Maykol Pinto, A. Paulo Moreira, Paulo G. Costa, and Miguel V. Correia. "Revisiting Lucas-Kanade and Horn-Schunck". *Journal of Computer Engineering and Informatics* (JCEI), 1(2):23-29, 2013;

- [19] Andry Maykol Pinto, Paulo G. Costa, and A. Paulo Moreira. "An Architecture for Visual Motion Perception of a Surveillance-based Autonomous Robot", in Proceedings of the *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 205-211, 2014;

- [24] Andry Maykol Pinto, A. Paulo Moreira, and Paulo G. Costa. "Streaming Image Sequences for Vision-based Mobile Robots", in Proceedings of the *Portuguese Conference on Automatic Control (CONTROLO)*, in press, 2014;

Chapter 2

Overview of Motion Analysis

Currently, there is a wide diversity of motion perception methods which is justified by the inherent complexity of the problem. Visual motion perception with static observers has been extensively explored for many surveillance applications. In contrast, detecting moving objects on image sequences obtained from mobile robots is more difficult because of an additional two-dimensional motion component created by the egomotion of the vehicle.

The most popular algorithms for motion perception are presented in this chapter. Related works announced in the recent literature are briefly discussed: sections 2.2.1, 2.2.2 and 2.2.3 outline the visual techniques for static observers, and later, section 2.3 focuses on some interesting approaches for motion detection and analysis based on moving observations, which is the theme of this thesis. Therefore, this chapter introduces the major differences of motion perception for static and moving observers.

2.1 Introduction

Motion is an important characteristic for visual perception since it supports a wide range of computer applications including perceptual organization [25], object recognition [26, 27], scene understanding [28, 27], tracking [29, 30], human-machine interaction [31, 8], autonomous robot navigation [7, 8, 9], augmented reality [32, 28], video-coding [33, 34], remote monitoring [15] and three-dimensional reconstruction [32]. There are different sensors that can help to perceive motion although, vision-based sensors are preferred because they represent an intuitive, non-invasive, lightweight and cheap solutions for monitoring external moving objects within certain areas.

The main goal of visual motion perception is to segment image sequences into *background* and *foreground* regions: regions with absence of temporal displacements and

regions with moving objects, respectively. The three-dimensional trajectory of a moving object is projected into the image plane which creates two-dimensional trajectories whose derivatives represent two-dimensional velocities. Several motion perception techniques have been proposed in the last decade because the pixel's variation of images under real conditions can be caused by the objects' movement or by physical phenomena. Motion perception techniques can be classified according to viewer's motion relative to the environment, namely, *Stationary Observation* (SOb) and *Moving Observation* (MOb).

◇ **Definition 1:** *Stationary Observation (SOb) - images are obtained by a fixed camera placed on the environment, i.e., the scene is captured by a static observer.*

◇ **Definition 2:** *Moving Observation (MOb) - moving platform equipped with a vision system captures the surrounding scene and provides observations from different points of view.*

Usually, motion perception is performed by surveillance systems with SOb; however, reliable solutions based on MOb make possible to automate new environments by taking advantage of the observer's displacement. In this context, MOb methods encourage a new generation of more flexible and autonomous systems. However, only a few research projects are robust enough to make possible the movement of the viewer [35]. In this case, conventional SOb methods cannot be applied directly because an estimative of the egomotion is required to compensate the movement of the observer.

2.2 Motion analysis based on stationary observations

Nowadays, the research based on static observations focuses on how to increase the quality of motion segmentation in scenes with illumination changes, dynamic background objects and temporal occlusions [36]. It is possible to identify in the literature three motion perception methods for conventional SOb systems [36, 37]: background subtraction, temporal differencing and optical flow.

2.2.1 Background subtraction

As is suggested by its name, a background subtraction technique separates objects of interest (*foreground*) from the rest of the image (*background*). It is the most typical and traditional approach to segment moving objects based on static cameras because makes it possible to recover shapes and features of the foreground objects. The literature is very rich in background subtraction approaches although, only the most relevant and traditional methods are presented in this introduction.

The background subtraction approach is consisted basically by two distinct phases: the creation (and update) of the reference model and the subtraction of the reference model from the current image. The segmentation of moving objects is accomplished by a subtraction operation which defines the classification of each pixel between foreground and background. The moving objects are retrieved after the classification, enabling a further interpretation and analysis of objects by high level algorithms. The scientific community recognizes background methods as those that provide the best compromise between performance and reliability [37]. They are commonly used due to their effectiveness and low-computational time which make them suitable for surveillance applications with real-time constraints.

Although, background subtraction techniques have some limitations that need to be considered [37, 38]:

- Variations of the background scene due to changes in lighting conditions (sudden clouding or light switch);
- Repetitive movements of non-static background objects, such as, bushes blowing in the wind and trees branches;
- Low quality of the image sensor or scenes poorly illuminated;
- Shadows of moving foreground objects create local changes of the background illumination;
- Camouflage, *i.e.*, visual similarity between the background and the foreground.

Therefore, the most critical aspects of the background subtraction methods are related with the process of update the reference model, the non-stationary background and the environmental illumination. Over the last few years, a large diversity of techniques has been presented. These techniques have different strengths and weaknesses in terms of segmentation accuracy and computational requirement and, hence, their success relies on the ability to solve the highest number of limitations. The most common approaches for motion segmentation based on background subtraction are the following: Running Gaussian Average [39], CodeBook [40], Mixture of Gaussians (MOG) [41, 42], Kernel Density Estimators [43], Mean-shift based estimation [44], Eigenbackgrounds [45, 46] and Markov Random Field [47, 48]. They differ in the way that the reference model is created.

The Mixture of Gaussians (MOG) is perhaps the most well-known background subtraction technique for environments with non-static backgrounds. The technique was initially proposed by Stauffer and Grimson [49, 50], and it models the intensity of each pixel

through a mixture of Gaussians. The reference is learned using an unsupervised technique that models the pixel as a statistical process which means, the intensity value is fitted by multiple Gaussian distributions.

For sequential images, each pixel is considered as foreground if none of the Gaussian distributions represent the intensity value. In this case, the mean value of the last distribution (with less representatively) is replaced by the current pixel value [51]. Contrarily, the pixel fits the reference model if its intensity value is up to 2.5 times the standard deviation of any Gaussian distribution. Afterwards, parameters of the distribution are updated using a running average, and weights are recomputed and normalized. The number of the Gaussian distributions for each pixel is usually a fixed number between three and five. In addition, all distributions are ranked at the end of each segmentation and according to the ratio of weight and standard deviation. This reduces the time spent on fitting pixels since more frequent distributions are ordered first. The reference model of the MOG is continuously updated to enable gradual changes in lighting conditions and repetitive background motions. Nevertheless, segmenting moving objects that stop for long periods of time is a difficult task. Also, the computational complexity of MOG techniques is considerable which limits its usability in some applications, for instance, segmentation in real-time of dynamic environments. A more detailed description of this approach goes beyond the scope of this research; however, an interest survey can be found in *T. Bouwmans et al. (2008)* [52].

The CodeBook method is another advanced background subtraction technique. It deals with the temporal fluctuations of pixels and the structural periodic motion of elements that belong to the background [40]. A pixel is represented by one or more codewords. Each codeword defines an interval that delimits the range of intensity values. A learning mechanism creates the reference model, and pixels of image sequences are clustered into a set of codewords which represents the distribution of the intensity values.

Whenever a pixel value is close to a boundary of any pre-existent codeword then, the interval of this codeword grows by a learning factor [53]. On the other hand, if the pixel value falls outside the current set of codewords then, a new codeword is formed. Codewords are continuously updated by deleting the oldest codebooks and based in the largest *negative runtime* (the longest time during which the codeword was not accessed). In this way, codewords rarely accessed are removed from the pixel's codebook and using a temporal filtering. The principle is that, the removed codewords were probably formed by foreground objects or noise. This improves the quality of the segmentation and the computational time. The number of codewords inside a codebook is not necessarily the same for all pixels which enables the representation of other parametric distributions to besides Gaussian, for instance, Exponential, Weibull and Lognormal distributions. After

reference model be created, the process of motion detection involves testing the brightness or the color of the current frame and the reference model. Each pixel is classified as foreground if its brightness or color lies outside the boundary of all codewords that constitute the corresponding codebook. If a pixel is classified as background, the codeword that was recently accessed is relocated in the front of the codebook to improve the computational performance. The CodeBook method has several features, namely, is robust to illumination changes, encodes moving elements that belong to the background, allows moving foreground objects during the creation of the reference model and it is computational efficient because motion segmentation does not involve calculating the probability using floating-point operations, unlike the MOG [40]. *Kyungnam Kim et al. (2005)* [40] propose two improvements related to the adaptive codebook updating and the layered modeling. The work was focused on a procedure to update the initial background model in order to increase the robustness of the method. The reference model is initially defined as permanent. The frequency of pixels is analyzed using the assumption of pixels that reappear for a certain period of time are incorporated in the reference model and codewords that are not accessed for a long time should be erased. Therefore, a pixel can have one of the following four classifications: permanent reference model, permanent background, non-permanent background and foreground. Sequences segmented through this CodeBook technique were compared to MOG and Kernel methods, and results showed that the proposed technique made possible to retrieve the shapes of objects classified as foreground with a better quality. In addition, the technique satisfies the real-time restrictions and limited amount of memory of surveillance applications.

Some background subtraction methods have limitations related with two commonly adopted assumptions that are often violated: the pixels are independent and the temporal evolution of the background is slow [53]. These assumptions are not truth for real-life situations since they ignore the spatial dependency among neighboring pixels which leads to inconsistent (noisy) predictions. Moreover, the background might change too much over time, for instance, due to wind, camera jitter, illumination changes, windows and doors [54].

Motion segmentation based on kernel estimation and a Markov Random Field (MRF) is presented in *Yaser Sheikh and Mubarak Shah (2005)* [55]. The research models the spatial dependencies of the observed intensity values, the temporal persistence of the foreground (current foreground objects contain substantial evidences for future segmentation procedures, *i.e.*, the foreground keeps the same consistency of color and the same spatial area), and the maximum a posteriori estimation (MAP) based in MRF that uses the spatial context for assessing the background and the foreground model. The kernel estimation of the spatial dependencies is based in Gaussian models, and resorts to a nonparamet-

ric method since it does not make any assumption about the shape of the probability density function of the feature space. Contrarily to conventional background subtraction approaches, the temporal persistence is considered in this approach as a property of realistic foreground objects. The likelihood function was obtained through a mixture of kernel density estimator and a *Parzen* classifier was used to classify the pixel as foreground or background. The classifier's threshold is computed using a priori knowledge of the spatial neighborhood information and the MAP-MRF framework. The approach was evaluated for different environment's conditions, *e.g.*, fountains, tree branches, grass, lake water, oscillating sea and ceiling fans. Moving foreground objects were successfully detected by the proposed technique, unlike the MOG method because the dynamic elements of the background were considered as foreground objects.

Background subtraction techniques are being implemented on a wide range of hardware and are currently used in a variety of surveillance scenarios; however, approaches are still not computationally efficient for real-time applications. Moreover, there are several other background subtraction techniques that incorporate another characteristic although, this section intended to give only a brief overview of the most often encountered methods.

2.2.2 Temporal differencing

Temporal differencing is the simplest method to detect moving objects. The traditional temporal differencing method is defined by an absolute subtraction of consecutive images. In this way, pixels whose absolute difference is greater than a pre-defined threshold are classified as foreground, otherwise, they are classified as background.

This method is very sensitive to any kind of movement but adapts quickly to changes in lighting conditions since the reference model is not created. However, it is not suitable for environments having moving elements belonging to the background because they are difficult to distinguish from the foreground. The method is also misleading for foreground objects that stop for a short period of time and for foreground objects with homogeneous colors since the temporal differencing may not detect all the foreground pixels (causing an internal cavity).

Despite the limitations, there are several researching works that resort to temporal differencing as part of more complex motion detection architecture. Moreover, it is used especially to start other algorithms like background subtraction and optical flow techniques [56] since it can improve the computational efficiency.

P. Spagnolo et al. (2006) [37] present a combination of background subtraction and temporal differencing. They estimated the radiometric similarity between corresponding

pixels of consecutive images in order to identify moving points through temporal analysis. The radiometric similarity turns the approach more robust to noise because a local window is used to compute the similarity level between images. Pixels of the current image considered as foreground are used to update the reference model and, in this way, the variation that is exhibited by all pixels with the same intensity are considered during the computation of the photometric gain. The algorithm compensates illumination changes which is an important feature for sudden changes of the light condition caused by the light switches. If the percentage of moving pixels is higher than 60% of the image then a sudden variation of light occurred, meaning that the temporal segmentation is compromised and the previous segmentation should be considered instead as the result of the background subtraction. In this case, the process updates the reference model by including the global variation of the light. *Murali and Girisha (2009)* [57] proposed a motion segmentation using three consecutive images and based on pixel-by-pixel disparity. Multiple correlations on the RGB (red, green and blue) colored space are used with a statistic model to remove shadows after the background and the foreground classifications. Multiple correlations of three sequential images made possible the evaluation of the linearity relationship between pixels of the three images. A spatial clustering method resorts to the spatial distance between pixels to fill the holes of the object segmented by the temporal differencing and the process of eliminating shadows. The research focused on static observations (single camera) and the authors claim that their approach adapts to non-static environments, is computational efficient and robust to illumination changes. However, the dataset used during the evaluation of the segmentation results was derived from the performance evaluation of tracking and surveillance (PETS) database and no experiment was performed to assess about the illumination robustness or even the computational complexity.

Section 2.3 presents additional researching works that use temporal differencing to detect and analyze moving objects. In that section, motion perception is performed based on MOb and, therefore, the temporal differencing cannot be applied directly between consecutive images.

2.2.3 Optical flow

Optical flow is one of the most well-known techniques for motion detection. It analyzes the spatial and the temporal evolution of pixels and assigns the respective motion vector. Usually, the term ”optical flow” is sometimes confused with motion projection of the three-dimensional objects in the two-dimensional image plane; however, they are not the same thing. In fact, there are factors that affect the estimation of the optical flow (for

instance, quantization, reflections and noise), which differentiate the estimation of motion projected on the image plane from the real projection. Although, there is a close relationship between both [58]. Optical flow techniques provide a motion vector for each pixel; however, they are usually computational expensive to be used in real-time applications and without special hardware [59].

2.2.3.1 Introduction

Motion extraction based on visual sensors captures 3D surfaces. Surfaces that move along a 3D path, **X**(t), are projected onto the image plane and, therefore, a 2D path, **x**(t), is produced for each image point that represents the moving object.

Important definitions are presented below to clarify some issues related with the terminology of motion perception based on optical flow techniques.

◇ **Definition 3:** *Image flow - is the projection onto the 2D image plane of the 3D velocity vector of objects. Can be considered as a bi-dimensional projection of the objects' movement.*

◇ **Definition 4:** *Optical flow - is an approximation to the image flow. Represents the movement that is really captured by the camera, i.e., the visible displacement of pixels. In an ideal situation, it is expected that the optical flow and the image flow are both equal.*

◇ **Definition 5:** *Scene flow - is the 3D optical flow and specifies how much each voxel moves between adjacent volumes.*

The optical flow is a vector (magnitude and direction) formed by the intensity variation of pixels along time and is commonly represented by a vector field [10]. It is an approximation to the two-dimensional projection of the real movement on the image plane because there are several situations where the optical flow and the image flow are not equal. Some of these situations are presented in [10, 60].

2.2.3.2 Mathematical definition

The optical flow estimation generally assumes that all temporal changes of the pixel's intensity are caused by motion. The pixel *I(x, y, t)* moves by δ_x and δ_y during the time δ_t to *I(x+*δ_x*, y+*δ_y*, t+*δ_t*)*. Mathematically, the intensity of pixels are translated according to Eq. 2.1, where *I(x, y, t)* is the image intensity at pixel coordinates (x, y) and time t.

$$I(x,y,t) = I(x+\delta_x, y+\delta_y, t+\delta_t). \tag{2.1}$$

Other assumption is related to the capture of rigid motion in a scene, whereas deformations of the object's shape that might occur during consecutive images are not granted (they are not particularly distinguishable).

◇ **Remark 1:** *Taylor series expansion is briefly recalled for a continuously differentiable function* $f(x)$ *in* $\Re \to \Re$. *For* $\delta_x \to 0$, *the* $f(x_0+\delta_x) = f(x_0) + \delta_x \frac{df(x_0)}{dx} + \sum_{i=2,3,\ldots} \frac{1}{i!} \frac{d^i f(x_0)}{d^i x}$. *The last term is called* $H.O.T.$ *(high order terms) and is usually ignored because this sum of terms has a small contribution when* $\delta_x \to 0$ *(*$H.O.T.$ *is zero if* $f(x)$ *is a linear function in* $[x_0, x_0+\delta_x]$*).*

The one-dimensional case of the Taylor series expansion can be straightforwardly generalized for the two-dimensional case, see Eq. 2.2:

$$I(x+\delta_x, y+\delta_y, t+\delta_t) = I(x,y,t) + \frac{\partial I(x,y,t)}{\partial x}\delta_x + \frac{\partial I(x,y,t)}{\partial y}\delta_y + \frac{\partial I(x,y,t)}{\partial t}\delta_t + H.O.T. \tag{2.2}$$

This optical flow formulation is possible due to some implicit assumptions, namely, the *brightness consistency* and the *temporal persistence*.

◇ **Definition 6:** *Brightness consistency - the appearance of the brightness patterns that represents part of an object does not change as it moves between consecutive images. The surface exhibits the Lambertian*[1] *reflectance which means that the apparent brightness of the surface is the same regardless of the observer's angle of view. The surface luminance is isotropic and, therefore, this assumption justifies Eq. 2.1.*

◇ **Definition 7:** *Temporal persistence - temporal increments are small relative to the magnitude of motion. This means that objects do not move too fast and the image can be approximated by a linear function (*$H.O.T. = 0$*). The approximation of the first order of the Taylor series expansion is relatively true due to the small neighborhood that is commonly considered [62].*

The two-dimensional *motion constraint* is obtained by dividing the Eq. 2.2 with δ_t:

$$\frac{\partial I(x,y,t)}{\partial x}\frac{\delta_x}{\delta_t} + \frac{\partial I(x,y,t)}{\partial y}\frac{\delta_y}{\delta_t} + \frac{\partial I(x,y,t)}{\partial t} = 0. \tag{2.3}$$

[1] Johann Heinrich Lambert (1728 - 1777) was a Swiss German mathematician, physicist, astronomer and philosopher, who provided the first rigorous proof that the value of π is irrational (cannot be expressed as the quotient of two integers) [61].

The intensity derivatives of the image can be expressed as:

$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y} \text{ and } I_t = \frac{\partial I}{\partial t}. \tag{2.4}$$

Therefore, the equation of motion constraint can be rewritten as it follows:

$$\nabla I^T \cdot \mathbf{v} + I_t = 0, \tag{2.5}$$

where $\nabla I = (I_x, I_y)^T$ denotes the spatial intensity gradient which can be computed using derivative operators, the $\mathbf{v} \equiv (\frac{\delta_x}{\delta_t}, \frac{\delta_y}{\delta_t}) = (u, v)^T$ is a 2D flow vector (x and y component) and I_t denotes the temporal gradient at time t. The temporal gradient cancels the inner product of the spatial gradient and the optical flow vector in Eq. 2.5 due to the brightness constraint.

2.2.3.3 Aperture problem

Ideally, the variation of the pixel's brightness is a result of the moving objects in the scene. However, in realistic environmental conditions may be caused also by photometric effects, illumination changes, variations of the object's surface and lens distortions.

In addition, it is difficult to determine the spatial variation of pixels that belong to objects with homogeneous brightness because the spatial movement of each pixel is estimated using the spatio-temporal information of its neighbors. Thus, motion estimation according to every spatial direction may be impossible to determine with a complete accuracy and, therefore, the movement cannot be estimated correctly. This problem is referred as *aperture problem*.

◇ **Definition 8:** *Aperture problem - denotes the inability to measure or to fully estimate motion in regions of the image that do not exhibit distinguishable characteristics. For example, flat regions, untextured surfaces or even line segments whose ends lie beyond the boundaries of the field of view. In these cases, it is possible to estimate only the normal component of motion, regardless of the technique [10].*

The two-dimensional motion constraint of Eq. 2.5 is the fundamental principle of the optical flow computation. However, it is only one equation with two unknown variables which means that the measurements are underconstrained and an unique solution cannot be obtained for a single pixel, see Fig. 2.2. Therefore, it is only possible to define the normal component according to the constrained line in the *velocity space* (velocity component in the same direction of the spatial gradient). The flow vector is estimated

by considering the pixel's neighborhood and the size of the gradient operators [63]. Detecting motion using a small aperture may lead to local information that is insufficient to define exactly the direction of the flow vector. For small apertures, the optical flow cannot be fully recovered which means that the pixel's neighborhood is unable to estimate the tangential velocity component. For instance, the aperture problem is caused by the edges and because of the small size of the aperture since the edges of Fig. 2.1 have gradient information only in one direction. In this case, the normal component can be estimated by Eq. 2.5 while the tangential component is missing. The structural information is enough to fully estimate the flow vector for pixels with the gradient information defined in both directions, for instance, a corner.

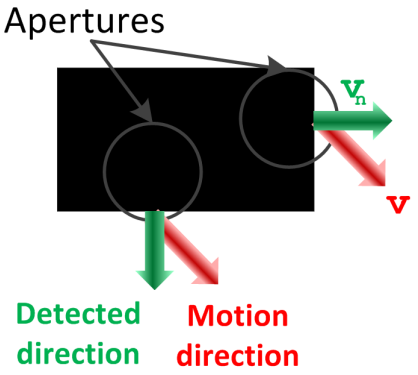


Figure 2.1: Image flow (red) and the optical flow (green) of a moving rectangle and using circular apertures. The tangential component $\mathbf{v}_t$ of the flow vector cannot be estimated.

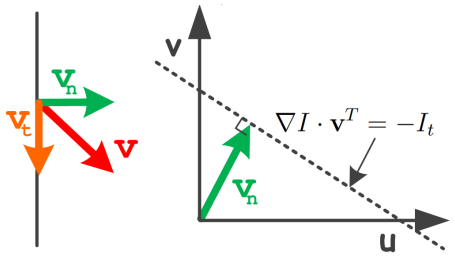


Figure 2.2: The 2D motion constraint originates a line (of dots) in the velocity space $\mathbf{v}= (u, v)$. The normal vector $\mathbf{v}_n$ is the velocity with lowest magnitude that lies on the line that is obtained by the equation of motion constraint.

$$\mathbf{v}_n = -\frac{I_t}{\|\nabla I\|^2}\nabla I \tag{2.6}$$

The v_n is the velocity component that can be retrieved in the absence of complete structural information and by a basic visual perception. A reliable estimation of the optical flow usually implies the incorporation of additional constraints. The most common way to introduce the necessary structural information is to use a neighborhood context, namely, the *spatial coherence* assumption.

◇ **Definition 9:** *Spatial coherence - embodies the assumption that surrounding pixels belonging to the same surface are likely to move together and share a similar motion [64, 53]. It may also be referred as the gradient constraint [65].*

2.2.3.4 Temporal aliasing

◇ **Definition 10:** *Temporal aliasing - a phenomenon characterized in the frequency domain by an overlap of the spectral contents of the continuous-time signal. It leads to a reconstructed signal that is different from the original continuous-time signal. The temporal aliasing is a common problem for detecting motion in sequential images and it is caused by sampling rates that are lower than the sampling criterion.*

Nyquist-Shannon sampling criterion defines that a continuous-time signal being observed should be sampled in a frequency at least twice the maximum frequency contained in the signal under observation. The sampling process converts the observed signal into a discrete-time signal and this theorem assures that it is possible to reconstruct and to recover the original signal from the discrete-time signal. Derivative filters used to compute the spatial gradient in digital images are sensitive to high frequencies because the sampling process of a continuous-time signal introduces replicas of the spectrum at intervals of $2\pi/T$ radians, where T denotes the time between frames. These replicas can be detected by the gradient operator which misleads the estimation of motion and, thus, two aliasing artifacts should be avoided during motion detection:

- Spatial sampling - photoreceptors of the visual sensor conduct a discretization of the scene that may not satisfy the sampling criterion. This means, the spatial variation of the scene has a frequency higher than half of the density of photoreceptors;
- Temporal sampling - the variation of brightness in the scene is discretized in a frequency higher than half of the sensor's frame rate. In practice, the temporal aliasing is caused by a lower sampling rate of the scene when compared to the speed of objects that are being captured.

Optical filters [10] can be used to remove the frequency components of spatial variations of the scene that exceed the sampling criterion. Also, there are image processing techniques that minimize the spatial aliasing effect. Before resampling the image in lower resolutions, anti-aliasing methods remove frequency components of the image signal that are higher than the sampling frequency of the visual sensor.

The shutter of the camera influences the temporal aliasing. A specific exposure time acts like a band-pass filter and attempts to prevent the temporal aliasing [10]. The temporal aliasing problem is usually seen in the stroboscopic effect: wheels of the vehicle seem to move backwards. Other anti-aliasing technique uses Gaussian pyramids during the optical flow computation. The optical flow is estimated from the coarsest scale to the finest scale of the pyramid. The coarsest level image is considerably blurred and

its velocity is lower; however, is used as initial guess for the estimations in finer levels. Derivatives calculate the residual motion of the image sequence when the original scale level is reached [65]. This hierarchical approach [66] stabilizes motion at finer scales and it is widely used in computer vision. Although, it has a major drawback related to error propagation since the computation of the optical flow in the coarsest level might have a large estimation error.

2.2.3.5 Erroneous situations

The brightness constancy assumption rarely holds in practice. The luminance of a surface is not isotropic due to specular highlights since objects are not illuminated uniformly. However, the assumption works relatively well in realistic conditions [65]. The optical flow computational is usually sensitive to some situations that introduce a significant amount of error in the estimation of the flow field.

- The flickering of light sources such as fluorescent lights must be avoided;
- The background pattern formed by transparent structures or diffuse materials creates angular variations of the radiance level for the point being observed from different points-of-view;
- The camera frame rate is often insufficient to capture motion which leads to an observability problem;
- The first order approximation of the Taylor series expansion creates additional errors during the optical flow estimation for objects with large movements;
- A moving object with a non-rigid shape originates an apparent motion that is not particularly distinguishable from the real movement;
- A small aperture of the camera avoids the refocusing effect originated by volumetric refractions [67].

These issues contribute for the difference between the optical flow (apparent velocity) and the image flow. In some cases, it is possible to refine iteratively the initial estimation of the optical flow by using *Newton's* method. The iterative process will converge to better estimation results within about five iterations [53] if the initial guess is sufficiently good enough.

2.2.3.6 Optical flow techniques

The estimation of the optical flow is an underconstrained problem and, there are different algorithms that resort to additional assumptions to assist the computation of the flow. The techniques can be classified into:

- Differential methods - use spatio-temporal derivatives of brightness to estimate the optical flow. Partial derivatives are easy to compute; however, additional constraints are required to get a unique solution. Usually, a neighbor context is defined which introduces the necessary constraints that support a more robust calculation of the flow vector. Differential-based techniques can be organized into global [68] and local [69] methods. This subclassification is directly related to the neighbor concept that is used by the techniques during the estimation;

- Region-based matching methods - use an iterative process to detect motion. They establish a spatial correlation between a small region in frame A and a similar sized region in frame B. A correlation function makes it possible to estimate the motion vector that minimizes the sum of squared differences or the sum of the absolute differences (or maximizes the normalized cross-correlation). The region size should be small to prevent excessive smoothness since the motion vector is an average of all pixels inside that region. Region-based techniques are usually computational more demanding because sub-pixel accuracy is needed to express the peak of the correlation function and the integer precision may be insufficient for some real-time applications;

- Frequency-based methods - the comparison between different regions is performed using the Fourier domain of the temporal changes of images. Is more complex than other techniques because it requires several fast Fourier transformations (FFT), although, it gives very accurate results which makes it possible to estimate motion in cases where region-based methods fail [70]. Frequency-based methods can be used to extract repetitive motion patterns since the conventional motion vector is converted to a spatio-temporal frequency domain. In the frequency space, the non-zero energy that is associated to a two-dimensional translational pattern lies on a plane through the origin. Thus, the optical flow is estimated by searching for a plane that fits better into the spectrum of the spatio-temporal signal. The work [60] presents an interesting result that proves the relation between frequency-based methods, block-based methods and even some differential-based methods (Lucas-Kanade technique [69]).

It has been a long way since Horn-Schunck [68] and Lucas-Kanade [69]. Despite being an interesting story all remarkable evolutions occurred over the last three decades, this research provides a review of the most relevant improvements that were occurred after these two timeless methods.

The latest and the most significant advances in accuracy, robustness and usability for differential optical flow algorithms are briefly discussed. Current modern concepts incorporate a multiscale (coarse-to-fine) refinement [21, 71] to deal with large displacements and to prevent aliasing, and iterative approaches using interpolation for warping images [72, 73]. The introduction to robust statistics by [74] is an important improvement because replacing the quadratic penalty function, by non-quadratic and non-convex functions increases the robustness of the estimation to outliers caused by occlusions and noise. This technique prevents flow smoothing in motion discontinuities and is commonly used by several techniques [75, 76, 77].

Another relevant improvement is the *gradient constancy assumption*, proposed by [78]. This technique allows small variation in the brightness value. Gradient and brightness constancy assumption are originally combined with a non-quadratic penalty function in [78]; however, better results were obtained by imposing a separate robust penalty function for each assumption [79]. Furthermore, the normalized brightness and the gradient constraints are employed in [80] and were recently improved by [81]. Color image sequences are integrated into the optical flow methods by considering alternative color spaces [80] that provide photometric invariances. The [82] propose the optical flow in harmony method. Their data term combines the brightness and gradient assumption with normalization to avoid an overweighting of the term at larger gradient locations. It uses the HSV (Hue, Saturation and Value) color space, such as in [83], with a separate robust penalty function for each channel. The anisotropic smoothness term considers the directional information of the data term, which is also robustified with a penalty function. Finally, temporal average ranging of the derivatives, median and bilateral filtering proved to be essential for modern optical flow techniques [73]. These practices play an important role to the accuracy of state-of-the-art methods; however, some of them are computational demanding, requiring special programming techniques and hardware to estimate the flow field in less than a couple of seconds, namely, multi-threading architectures and GPU (graphics processing unit). This strong computational effort means that the approaches are not very appropriate for the current robotics system.

Focusing on applications that compute the optical flow, *Denman, Fookes and Sridharan (2009)* [84] segment motion using a fixed camera and propose an adaptive background segmentation. The foreground is retrieved using a block-based optical flow technique to ensure temporal consistency of the moving objects. *Shui-gen Wei et al. (2011)* [59]

propose a motion detection method based on the Horn-Shunck [68] with a self-adaptive threshold. The optical flow of two consecutive images is initially computed, converted to a gray scale image and then, the binarization is accomplished using the Otsu method. *J. Wendi and Jianqin Han (2011)* [85] study the segmentation of a moving object with the problem of camouflage. A pyramidal Lucas-Kanade [69] technique obtains the pattern of motion which is used to extract motion models of the object and the background. The segmentation is achieved by a Kalman filter that takes into consideration the location and the magnitude of the flow vectors. *Marco Tagliasacchi (2007)* [58] presents a genetic-based optical flow estimation algorithm. The current frame is segmented using a watershed algorithm and by grouping the pixels with the same spatial position and similar color. The author assumes motion coherence in pixels of the same region, which means that the velocity field is smooth and only abrupt along the region boundaries. The affine model is applied to capture the motion (making it possible to describe complex motions) and the six affine coefficients are computed based on a genetic algorithm. Each six-parameter solution is an individual population that is explored by the genetic algorithm and the initial population is selected at random. The objective function reflects the energy of the frame difference according to the values of the six-parameter. This approach performs better at the border of the objects when compared to the Lucas-Kanade (due to the discrete approximation of the partial derivatives); however, the computation is too complex for a real-time performance since it took more than one second to compute a 176×144 image with a Pentium M 1.6GHz .

Naoya Ohnishi and Atsushi Imiya (2006) [86] demonstrate an algorithm that computes the dominate plane using the pyramidal Lucas-Kanade [69]. Assuming the largest area for the dominant plane and a finite distance from the camera to the plane, they prove that the matched features of consecutive frames will be based on a projection of the dominant plane that is represented by an affine model (the homography can be approximated by an affine transformation if the camera displacement is small). The affine coefficients are computed using three randomly selected pair of points, and the dominant plane is detected using the difference from the optical flow and the planar flow. The planar flow that is used by the following images can be estimated by applying the least-squared method and by computing the dominant plane afterwards. The results show that authors obtain less than 25% of error during the initial estimation of the affine coefficients. The resulting error after the fifth frame is less than 5% meaning that the convergence time is faster. The approach fails when the selected points are mismatched pairs since the affine coefficients are not estimated properly. *José Martín et al. (2005)* [87] propose a pipelined architecture to compute the Horn-Schunck [68]. The calculation of the different stages is conducted simultaneously due to the serialization of the data. In this way, the system computes part

of the optical flow at some particular pixel position and calculates, at the same time, the partial derivatives of the following pixel. The hardware implementation of this approach achieved a real-time performance with little latency.

2.3 Motion analysis based on moving observations

The visual motion perception with a moving observer is a complex and challenging problem with difficult solution because the observer's motion must be quantified first to provide clues that are used during the extraction of the moving objects. Methods and algorithms for motion perception based on MOb are quite recent. MOb represent a fundamental problem to several applications, especially, for mobile robotics and surveillance systems. In the past few years, the number of works that resorts to moving cameras is increasing, and yet, many of these attempts use only active cameras. For instance, pan-and-tilt cameras offer a larger coverage area when compared to static cameras and they have improved the flexibility of the remote monitoring.

Nevertheless, visual motion perception with MOb is expected to have a large and revolutionary impact in the surveillance of new locations since more conventional methods like the background subtraction and the temporal differencing cannot be applied directly without a method to compensate the egomotion. The formulation of the problem that arises from the egomotion is detailed below.

2.3.1 Introduction

Most methods that were presented so far assume that images are captured with static observers. The displacement of the observer increases the complexity of the SOb-based formulation because new aspects need to be considered: two independent movements are blended together. Hence, the 2D visual information captured by a moving observer has the following motion components:

- Egomotion (motion of the observer);
- Movement of the external objects.

Typically, methods for SOb assume the same spatial correspondence over time of each pixel. Therefore, motion is detected by performing a temporal analysis of the brightness. The formulation of motion perception with MOb is more complex because it cannot assume the spatial correspondence of pixels in frames at different time instants, *i.e.*, the position of each pixel changes over time and even for scenes without moving objects. In

this context, an estimation of the egomotion must be previously obtained to compensate the motion component that does not reflect moving objects. This means, motion analysis is conditioned by the displacement of the observer that is capturing the environment.

Therefore, the problem of motion perception based on MOb has two steps: the egomotion estimation and the disassociation of motion components.

2.3.2 Interpretation of the observer's motion:

The three-dimensional movement of the observer influences the process of motion perception because motion causes a two-dimensional projection. Hence, the two-dimensional motion that results from the three-dimensional translational and rotational of the observer must be quantified. The mathematical formulation of this section follows closely the fundamentals presented by *H.C Longuet-Higgins and K. Prazdny (1980)* [88].

A monocular observer moving in a static scene has a motion profile with two components: the translational, $\dot{\Gamma}= (t_X, t_Y, t_Z)^T$, and the rotational velocity, $\Omega= (w_X, w_Y, w_Z)^T$. Both components are described in the camera coordinate system. That coordinate system moves together with the camera and, therefore, the coordinates of a static point $\mathbf{P}= (X, Y, Z)^T$, represented in the camera coordinate system, change over time.

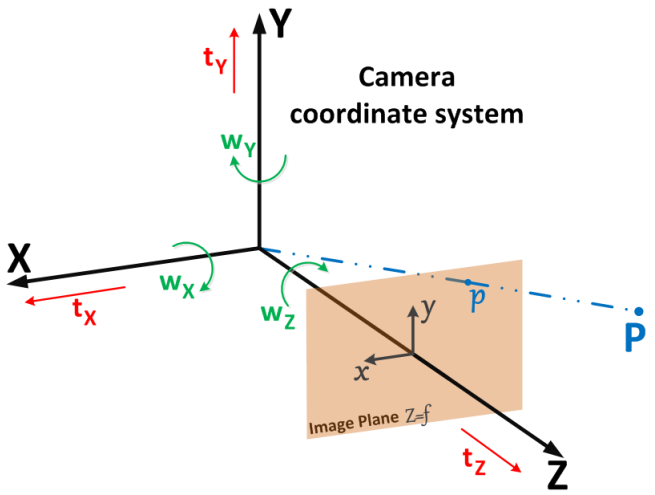


Figure 2.3: The geometry of the image formation. The camera coordinate system moves with translational (red) and rotational (green) velocity. A static point is represented by **P** in the camera coordinate system and its projection into the image plane is portrayed by **p**.

The velocity of **P** is in opposite direction to the camera movement and can be described by $\dot{\mathbf{P}}$:

$$\dot{\mathbf{P}} = -\dot{\Gamma} - \Omega \times \mathbf{P}. \tag{2.7}$$

Considering the pin-hole camera model:

$$x = f\frac{X}{Z}, \text{ and } y = f\frac{Y}{Z}, \tag{2.8}$$

where f is the focal length. The two-dimensional velocity of $\mathbf{p}$ is defined as:

$$\mathbf{v} \equiv (u,v)^T = \begin{bmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{bmatrix}. \tag{2.9}$$

Substituting the Eq. 2.8 into Eq. 2.9 it gives:

$$\mathbf{v} = \frac{f}{Z}\begin{bmatrix} \dot{X} - \frac{X\dot{Z}}{Z} \\ \dot{Y} - \frac{Y\dot{Z}}{Z} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} f\dot{X} - \dot{Z}x \\ f\dot{Y} - \dot{Z}y \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix}\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}. \tag{2.10}$$

Now, from the Eq. 2.7,

$$\mathbf{v} = \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\dot{\Gamma} + \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix}\Omega. \tag{2.11}$$

Then, the two-dimensional velocity can be expressed like:

$$\mathbf{v} = \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\dot{\Gamma} + \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\begin{bmatrix} 0 & 1 & -\frac{y}{f} \\ -1 & 0 & \frac{x}{f} \\ \frac{y}{f} & -\frac{x}{f} & 0 \end{bmatrix}\Omega$$

$$= \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\dot{\Gamma} + \begin{bmatrix} \frac{xy}{f} & -f - \frac{x^2}{f} & y \\ f + \frac{y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix}\Omega. \tag{2.12}$$

Equation 2.12 does not include the position of the point represented in the three-dimensional camera coordinate system. Therefore, the apparent motion is a vector sum of the translational and the rotational velocity of the camera. The inverse of the depth appear in the translational component of the expression, which means that a scaling factor is assigned to its calculation [10]. This phenomenon is often called as the *parallax effect*.

◇ **Definition 11:** *Parallax effect - represents the inability to distinguish between a near object that moves slowly from a distant object that moves quickly, and vice versa, if the camera moves and the object remains static on the environment.*

Suppose the visual observation of two static points, P_1 and P_2, at different depths, Z_1 and Z_2. If the observer moves along the environment (with a non-null translational

component) the apparent motion vector of each point is different due to the parallax effect. This is the reason why some researches avoid monocular vision and resort to stereoscopic systems or 3D sensors.

2.3.3 The apparent motion of MOb:

The apparent motion of each pixel is a combination of the egomotion, $\mathbf{v}_{ego} = (u_{ego}, v_{ego})^T$, and the objects motion, $\mathbf{v}_{obj} = (u_{obj}, v_{obj})^T$. These motion vectors are two dimensional projections of three-dimensional motions. The motion vector of the i^{th} pixel, $\mathbf{v}^i_{motion} = (u^i_{motion}, v^i_{motion})^T$, for an image captured by a moving observer can be expressed as:

$$\begin{cases} \mathbf{v}^i_{motion} = \mathbf{v}^i_{ego} + \mathbf{v}^i_{obj}, & \text{if } i \text{ represents moving objects (foreground)} \\ \mathbf{v}^i_{motion} = \mathbf{v}^i_{ego}, & \text{if } i \text{ represents static objects (background)} \end{cases} \tag{2.13}$$

Equation 2.13 depicts the problem of using the relative velocities and shows the importance of knowing the egomotion because the object's motion can be obtained by $\mathbf{v}_{obj} = \mathbf{v}_{motion} - \mathbf{v}_{ego}$.

The foreground and background velocity can be re-written by combining Eqs. 2.12 and 2.13:

$$\begin{aligned} \mathbf{v}_{motion} = {} & \frac{1}{Z_{obj}} \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \begin{bmatrix} t^{ego}_X \\ t^{ego}_Y \\ t^{ego}_Z \end{bmatrix} + \begin{bmatrix} \frac{xy}{f} & -f - \frac{x^2}{f} & y \\ f + \frac{y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix} \begin{bmatrix} w^{ego}_X \\ w^{ego}_Y \\ w^{ego}_Z \end{bmatrix} \\ & + \frac{1}{Z_{obj}} \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \begin{bmatrix} t^{obj}_X \\ t^{obj}_Y \\ t^{obj}_Z \end{bmatrix} + \begin{bmatrix} -\frac{xy}{f} & f + \frac{x^2}{f} & -y \\ -f - \frac{y^2}{f} & \frac{xy}{f} & x \end{bmatrix} \begin{bmatrix} w^{obj}_X \\ w^{obj}_Y \\ w^{obj}_Z \end{bmatrix}; \end{aligned} \tag{2.14}$$

$$\mathbf{v}_{motion} = \frac{1}{Z_{scene}} \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \begin{bmatrix} t^{ego}_X \\ t^{ego}_Y \\ t^{ego}_Z \end{bmatrix} + \begin{bmatrix} \frac{xy}{f} & -f - \frac{x^2}{f} & y \\ f + \frac{y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix} \begin{bmatrix} w^{ego}_X \\ w^{ego}_Y \\ w^{ego}_Z \end{bmatrix}, \tag{2.15}$$

where Z_{scene} is the depth to the scene (static), and Z_{obj} is the depth to the moving object.

The egomotion is normally estimated using a motion model, for instance, translational, Euclidean, similarity, affine and projective. These models have the ability to represent several types of movement with different properties. Usually, the affine and the projective models describe complex types of movement (translation, rotation, zooming,

etc) because the number of degrees of freedom is high; however, they are more computational expensive. Hence, a suitable motion model should be selected according to the requirements of each application, *i.e.*, the computational requirement and the quality of the representation of motion. Moreover, the estimation of $\mathbf{v}_{motion}$ and $\mathbf{v}_{ego}$ might have numerous errors due to numerical approximations, sensor noise and photometric effects like, reflections, shadows, transparency and changes in lighting.

2.3.4 Techniques for visual motion perception with MOb:

Visual motion detection and analysis for moving observers is becoming an active research field and preliminary techniques use typically one of the following approaches:

- *Organizing the background into moisacs* [12, 89, 90, 91] - A mosaic approach uses the background subtraction to detect motion. Firstly, the mosaic is created using spatial registration and tonal alignment techniques. Creating a mosaic background presents several disadvantages due to the photometric and spatial misalignments [36];

- *Modifying background subtraction methods* [36, 92, 93] - In some applications, conventional background subtraction methods are extended in order to incorporate the displacement of the visual sensor. Usually, this approach adds spatial information, and allows motion perception along some pre-defined movements;

- *Optical flow and geometrical models* [18, 94, 87] - The optical flow approaches resort to dense optical flow fields and to sparse flows (only some features to extract information about egomotion of the visual sensor). The egomotion is computed based on motion models. The cluster techniques can be applied to segment pixels that correspond to moving objects. This type of approach is commonly used in applications where the observer has several degrees of freedom.

Motion perception and analysis are an extremely important problems for several mobile robotic applications, especially for unmanned aerial vehicles (UAV) [95, 91, 96, 97]. An interesting survey about imaging perception techniques applied to robotics can be seen in [98].

Aryo Ibrahim et al. (2010) [12] present a mosaic technique for an UAV application. The technique maps the areas and detects moving objects. The authors match invariant features SURF (speeded up robust feature) or SIFT (scale invariant feature transform) between frames in order to compute the matrix that describes the geometrical transformation (projective model). The matrix is used by the warping process which aligns the current

frame with previous frames. The mosaic approach has misalignments and fragmentations problems in urban areas. They blur the aligned images (using Gaussian convolutions) to overcome these situations. The location of moving objects is obtained by a learning technique that was proposed by *D. Lee (2005)* [99] which is based on Gaussian mixtures. The learning method discriminates the density of blobs (binary large object)[2] in two groups: low density (background) and high density (foreground) sets.

Jing Li et al. (2011) [89] focus on monitoring the highway traffic flow using an airbone monocular camera. The goal of this research is to detect moving vehicles. They detect the road by extracting areas with similar intensities. The detection of edges is conducted through Canny's method and the extracted regions are represented by blobs. The authors identify the road by assuming that the blob of the road is larger than other blobs. A simplified Lucas-Kanade method combined with an image registration technique makes it possible to obtain the motion vector between consecutive frames. This defines the egomotion (affine model) of the vehicle. Finally, the moving objects are retrieved by the temporal differencing approach. The researching work presented in [90] proposes the detection of moving objects using the Kanade-Lucas-Tomasi feature tracking [69]. The homography is calculated using the 5-point RANSAC (random sample consensus) with all features of the two consecutive frames. The RANSAC analyses inconsistent features during the computation of the homography matrix and an online-boosting algorithm is used to follow moving objects.

A tracking application that resorts to a pyramidal Lucas-Kanade optical flow is presented in *Jay et al. (2011)* [100]. The researching work intends to identify and to extract regions where the flow field does not represent the UAV's egomotion, for instance, for tracking a target that moves at different velocity comparatively to the background. The authors compute the pure movement of the target by searching along the diagonal direction of images, which increases the incoherence of the assumptions made by the Lucas-Kanade technique; however, it reduces the computational complexity. They assume that a single camera is perpendicularly mounted down toward earth. Tracking of a moving object is accomplished using shape and color information. The experiments conducted have a trajectory error of 1.3 meters. An approach for motion clustering and classification based on consecutive images and a free-moving camera is presented in *Jiman et al. (2010)* [101]. The approach uses an optical flow technique and the random sample consensus (RANSAC) which removes outliers (scattered points) in the flow field. The flow field in Cartesian coordinates is transformed into polar axis (magnitude and orientation), and then, divided into blocks. The initial number and the respective cluster center are ob-

[2] A blob is an region of pixels with similar spatial characteristics.

tained by counting the selected block and by computing the density of the moving points. The clusters are redefined using the RANSAC, where each point is assigned to the initial cluster. Iteratively, the Euclidean distances to the clusters are computed and each point is updated with the cluster that has the minimum distance. The foreground and background are classified using the eigenvalue analysis based on the scatter of the cluster distributions, because they assume that the background is more scattered than the moving objects (due to their highest number of pixels). The authors estimate different motion models for the background model since the distances between the camera to the objects are not fixed. Therefore, the image is divided into blocks and the perspective model is computed for each block. The major problem of this approach is related to textureless backgrounds since the flow field is computed using features.

Fernández-Caballero et al. (2010) [15] present a human detection method based on a thermal infrared camera mounted on an autonomous mobile robot. The detection is accomplished using a combination of optical flow and temporal differencing. The non-pyramidal Lucas-Kanade is used when the robot moves and the temporal differencing is used to detect human candidates based on thermal signatures when the robot stops. This dual method tries to take advantage of both approaches, and the optical flow is not recommended when the observer is stationary because it cannot provide reliable clues in homogeneous regions. The traditional consecutive image difference is used when the robot stops. Afterwards, the resulting temporal image is binarized using a small threshold in order to remove the ghost effect. The authors focus on detecting motion interactions; however, the thermal camera facilitates the detection of humans.

Abhijit Kundu et al. (2010) [102] focus their research in detecting moving objects using a monocular vision system mounted in a robotic platform. The features from accelerated segment test (FAST) corners are extracted at different image pyramidal levels and the zero-mean sum of squared difference (SSD) is used to match the features between consecutive images (foreground and background matching). Only some frames are used to triangulate the three dimensional points and the epipolar geometry gives the initial estimative of the camera's pose. Then, an iterative process redefines the estimation of the cameras' pose by minimizing the first order approximation of the reprojection error, called *Sampson error*, that is calculated from the structure from motion (epipolar geometry). A recursive Bayes framework uses the constraint of the geometric view to compute the probability of each feature being considered as dynamic or static. Features with high probability values are considered as moving objects and, afterward, they are clustered by the spatial proximity and the motion coherence by applying move-in-unison model. The approach was evaluated using an image dataset and it needs about 10 milliseconds to process images with a resolution of 512×284; however, the method was implemented with

a multi-threading architecture and information about the CPU (central processing unit) is missing, which makes it difficult to compare to other methods.

Quian Yu and Gerard Medioni (2008) [103] propose a mosaic approach for a moving observer and they assume that the depth of the scene is much smaller than the distance between the object and the camera. This means that all the captured points are approximately in the same plane. This assumption enables the homography-based approach. After computing the homography between two consecutive frames, the egomotion of the camera is compensated by warping the sequence of frames to a reference frame. The accumulation of errors affects the registration and compromises the motion detection process. To prevent registration errors from spreading, the authors adopt a sliding window and only a number of frames are considered. The movement of the sliding window demands a high computational effort because all the registration processes must be executed. However, the algorithm was implemented in GPU and the time it took to compute a 320×240 image was less than 100milliseconds.

A method to calculate the distance between the target and the moving camera is presented in *Masaaki Shibata et al. (2008)* [104]. The block-matching optical flow with the matching criterion based on the summation of absolute difference (SAD) provides a flow vector for each block. The object distance is calculated using the optical flow and the camera motion. The method was developed for translational movements of the camera and it uses only the most reliable flow vectors during the calculation of the target distance, leading to very accurate distance estimation.

In *Ming-Yu Shih et al. (2007)* [105], the detection of moving objects is accomplished based on the temporal differencing between two consecutive pairs of frames (three frames). The method is based on two phases: blob detection and shape extraction. The affine transformation compensates the displacement of the two frames relatively to the middle frame. Motion models are refined through dense flow fields obtained from the middle and the compensated frames. In this way, the magnitude of motion vectors makes it possible to distinguish misalignments from moving objects. The connected components analysis compares the value of each pixel to all its neighbors in order to create blob structures for the moving regions of the image. The background model of the pixel is compensated using the affine motion model and shapes of moving objects are retrieved by combining the results from three background subtraction models (one for each channel). Finally, the models of the background are updated continuously by considering the position of moving blobs as foreground masks. *Ninad Thakoor et al. (2004)* [18] use temporal differencing approach with motion compensation. The egomotion model (affine motion) is computed using the hierarchical Lucas-Kanade optical flow technique over three consecutive frames. Affine motion parameters are computed iteratively using

a reweighed least square. The forward and the backward model is obtained relatively to the middle frame which makes it possible to generate an estimative of the reference model (for background subtraction). Thus, frame differences are consecutively combined and the moving pixels are detected from the middle frame. The presented approach is interesting since moving objects are detected; however, their boundary is not extracted completely.

S. Berrabah et al. (2006) [92] use a moving camera mounted on the top of a mobile robot and a Bayesian approach to estimate the egomotion. They resort to a background subtraction method (MOG) combined with a maximum a posteriori probability and a Markov random field. The MRF takes advantage of the spatial and the temporal dependency of moving objects that are depicted on image sequences. The egomotion is compensated from the current frame and during the robot's movement by applying an approach based on dense motion analysis. The compensated frame is used by a MOG technique with a probability inference procedure that characterizes the fitness of motion and segments the moving objects. *Rita Cucchiara et al. (2004)* [93] use region growing with color information to segment the image in different regions by assuming that each region contains part of one object. A topological graph is used to represent regions as nodes and the arcs depict the spatial relation between nodes. Features like the size, the position of the centroid and the bounding-box are also associated to the graph. The translational model for the egomotion is adopted by the authors to enable a real-time estimation of motion vectors. These flow vectors are computed by a region matching procedure that is described in the paper. A Markov random field (MRF) optimizes the spatial graph through an energy function that represents moving objects by regions with similar motion properties.

An approach that estimates the egomotion of a monocular camera using feature correspondence and the Lucas-Kanade optical flow with outlier removal can be found in [106]. The egomotion model is estimated using a bilinear model due to the aliasing problem and the model's parameters are estimated using the matching features between consecutive images (least square optimization). Temporal differencing is performed between the current and the compensated image. Object detection and tracking is executed using a Bayes probabilistic formulation. An adaptive particle filter performs multi-modal object detection and tracking. After that, the Expectation-Maximization (EM) algorithm is used to cluster the particles and, as a result, the moving regions are extracted by thresholding the Gaussian mixture function. The approach was evaluated in different environments and using aerial and ground robots. The particle filter is able to detect one moving object; however, the technique processes images with a resolution of 320×240 in 5 frames per second, considering 5000 particles and an embedded Pentium III 1GHz.

In addition, several techniques for segmenting motion using parametric and non-parametric machine learning approaches can be found in [107, 108, 109]. *Gheissari, Bab-Hadiashar and Suter (2006)* [108] propose a motion segmentation algorithm that estimates the scale of the noise based on a selective statistical estimator and a model selection. Thus, it simultaneously recovers the scale of noise since the dataset is partitioned into two groups (the inliers and the outliers), and the segmentation of data is reduced to a hypothesis-testing procedure. *Alexiadis and Sergiadis (2009)* [110] use a weighted fuzzy c-mean clustering procedure to obtain the velocity estimates for color sequences. The dense optical flow fields are computed using square blocks and the estimated velocity is assigned to the center of the block after a median convolution. The authors separate the different types of motion in two-dimensional hypercomplex Fourier domain and resort to an energy-minimization-based approach. They assume that the velocity of the moving objects (translational motions) is smoothly time-varying. *Bugeau and Peréz (2008)* [109] address the problem of motion detection and segmentation in dynamic scenes with small camera movements. They use the Lukas-Kanade optical flow to compute the sparse flow field from features obtained by the Harris corner. Only the features with a high confidence in their optical flow estimative are considered in further steps. The characterization of the features is achieved by the mean values of brightness and texture for grayscaled images, and by the brightness of the three channels for colored sequences. Afterward, these points are clustered using a variable bandwidth mean-shift technique, and finally, the segmentation is conducted using graph cuts.

Kai-Kuang Ma and Hay-Yun Wang (2002) [111] present a region-based nonparametric and spatio-temporal segmentation technique. The flow field is estimated through the Lucas-Kanade method and the segmentation method has two steps: pre-clustering (a smoother optical-flow field is obtained by blurring small textured areas and then, these areas are merged based on the dominant region of the neighborhood) and post-clustering (the spatial segmentation is conducted by a fuzzy c-mean with a smoothing operation to improve the semantic meaning of homogeneous regions). The authors focus on unsupervised segmentation; hence, they provide a method to estimate the number of moving objects by analyzing the phase histogram. Dominant motions are retrieved through an adaptive threshold. The experiments were conducted with static video sequences, which justify more or less the fact that they have despised the information of magnitude. This research is quite interesting; however, the technique cannot be applied to the context of this thesis because it does not detect different objects that share a similar direction of motion.

The research work presented in [112] detects salient regions in the sequence. It proposes a sparse approach since feature points are tracked over time to pursue saliency

detection as violation of co-visibility. The co-visibility is defined in terms of epipolar equivalence which means, is coherent with the rigid egomotion. The optical flow of a set of features is used to estimate the velocity of the viewer and to determine the salient regions. The method was tested on aerial video sequences which are expected to have a significant amount of features. Moreover, results show that the method does not achieve a real-time computation (32.6 seconds) because M-estimators are used to improve the segmentation procedure by removing outliers. Feature-based techniques are usually preferred due to a lower computational demand although, the realistic environment of the current thesis does not provide sufficient clues for sparse approaches.

Samuel Schulter et al. (2013) [113] present a block-wise motion segmentation method. The anisotropic Huber-L1 method computes the optical flow and motion segmentation is conducted using the conditional random field (CRF). The camera movement is robustly estimated as an affine model via RANSAC and considering a small part of the border of the flow field. The result is robust since the temporal coherence of moving objects removes the outliers. The clustering is conducted by a bag-of-words model built on dense scale-invariant feature transform (SIFT) features. The similarity between clusters is computed using the Chi-squared distance and object categories are discovered from the videos by learning the appearance model for each cluster through a Hough forest method. A segmentation technique that uses long term point trajectories based on dense optical flow is presented in [114]. These long term point trajectories made possible the analyzation of the temporal coherence consistency of clusters over many frames. The authors define the distance between trajectories as the maximum difference of their motion. The results show that the proposed method achieves an accurate pixel-wise segmentation; however, the method takes 497 seconds to compute 10 frames of the ”people1” sequence in the Hopkins dataset. This time is not affordable by most of the robotic systems and especially by mobile robots.

Eibl and Norbert (2008) [115] evaluate the performance of several clustering methods namely, K-means, self-tuning spectral clustering and nonlinear dimension reduction Isomap. Authors defend that one most important factor for clustering dense flow fields is the proper choice of the distance measure. They consider the feature space as being formed by the coordinates of pixel and motion vectors, whose values are normalized by taking into consideration the mean and standard deviation of each feature. Results show the difficulty of segmenting dense flow fields because no technique was outperformed and, thereby, the choice of the most suitable clustering technique and distance metric must be investigated for a specific context and environment. *Hu, Ali and Shah (2008)* [116] use a direct neighborhood graph and a hierarchical agglomerative approach to group flow vectors into coherent motion patterns. A direct neighborhood graph exploits the geometric

structure and the proximity of flow vectors. Small clusters are removed from the results because the authors assume that flow fields may have a noise component that causes isolated and meaningless clusters. The approach is interesting; however, the segmentation of a single frame resorts to motion information of all frames from the video. This is computational expensive and limits its applicability to post-processing (batch) applications.

2.4 Final considerations

The robust perception of motion based on moving observations enables new market niches since it reinforces the autonomy of mobile robots in areas that are not explored yet. Specifically, a better interpretation of the scenario makes it possible to automate new surveillance processes that are currently carried out through remote monitoring. In robotics, it is well known that cognitive processes are executed more easily if information of motion is available, for instance, tracking, recognition and obstacle avoidance.

This chapter introduced several techniques for motion detection and analysis using visual sensors: the basic concepts of motion perception for a stationary observer and later for a moving observer were presented. Moreover, the chapter presents some researching works that study in detail the problem of motion perception for a MOb. Techniques for motion analysis with a MOb can be organized into three approaches, for instance, organizing the background into moisacs, modifying background subtraction methods; and optical flow and geometrical models. The advantages and the disadvantages of each approach are discussed. As it can be noticed, the detection of motion for a MOb is inherent more difficult than for a SOb because the egomotion creates an additional velocity component that depends on the structure of the environment (the depth) and the magnitude of motion.

Therefore, this thesis studies motion detection, measurement and analysis for the surveillance scenario that is presented in chapter 3. To date, no similar robot or testing scenario has been found in the literature which demonstrates the innovation level of this research; however, interesting studies related to motion perception were found in UAV-based researches.

Chapter 3

The EEyeRobot

This chapter presents a mobile robotic system designed for active surveillance. The robot is called *EEyeRobot* and uses a monocular camera to acquire information about the environment, reports security issues and autonomously navigates along the rail that is placed in the ceiling. The robot has an architecture for visual motion perception that is formed by two modes: static perception and dynamic perception. Motion perception with a static observer is quite different from the moving observer because when a static observer captures the scene, every spatial and temporal variation represents part of the moving object (neglecting illumination changes and noise). Therefore, the perception system, that is proposed in this chapter, resorts to methods based on dense optical fields when the robot is moving and to more conventional techniques when the robot is standstill.

The chapter[1] is organized as follows. Section 3.1 gives an overall presentation of concept of the *EEyeRobot* and a brief description of the environment where the robot operates and section 3.2 presents the robot's architecture: software and hardware. Afterwards, section 3.3 shows the scheme for visual motion perception that is proposed in this thesis. Finally, section 3.4 presents the most important conclusions of this chapter.

3.1 Introduction

Motion analysis is a challenging problem in computer vision and robotics because the perception and the interpretation of motion are fundamental requirements for the correct operation of several mobile robotic applications like, the UAVs (unmanned aerial vehicles) [91, 96]. Usually, the research-line of visual motion detection and analysis for moving observers follows one of the three approaches: organize the background into mosaics [12, 89, 91]; modify background subtraction methods [92]; or apply optical flow

[1] Some portions of this chapter appeared in [19].

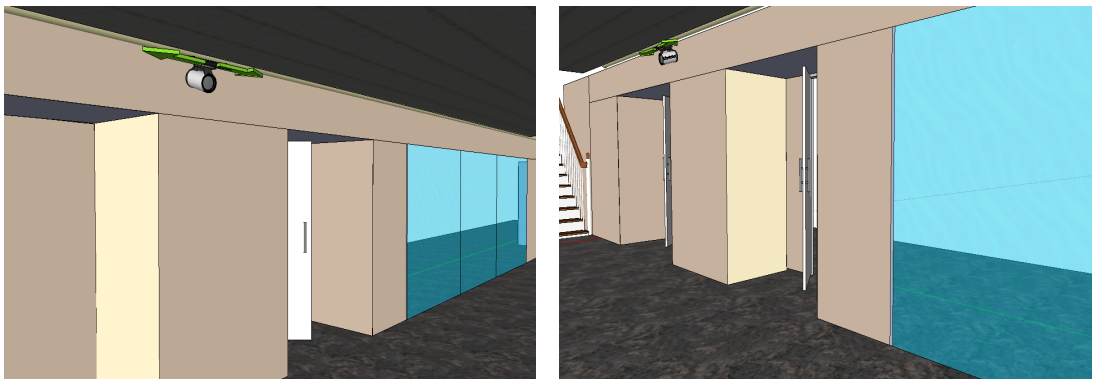
and geometrical models [18, 87]. On the other hand, conventional surveillance tasks are mainly performed with multiple static cameras, Figs. 3.1(a) and 3.1(b). Current research focuses on cooperative video networks and multiple sensor control [36]. Installation and calibration methods for multiple cameras have a high development cost and it is difficult to implement in a large environment. In addition, several calibration methods have been designed to reduce redundant cameras because sensor deployments have a large economic impact. A good configuration of sensors should be selected to cover the entire area with the minimum number of sensors by taking into consideration the time cycle of computer vision algorithms and blind regions. Moreover, conventional Closed-Circuit TeleVision (CCTV) systems have problems concerning the cooperation between sensors, for instance, synchronization, objects correspondences and communications [36]. All of these aspects make traditional security applications very unpractical for some large scale environments.

(a) train station. (b) CCTV control room.

Figure 3.1: 3.1(a) - train station with a high number of surveillance cameras. 3.1(b) - represents a common CCTV control room. A large number of cameras increases the complexity of the autonomous analysis and interpretation of certain events on the environment.

For this reason, the thesis presents an innovative mobile robotic system designed for active surveillance operations. The robot is named *EEyeRobot* and it is presented during this chapter. The main objective of this robotic application is to create a surveillance mobile system that autonomously detects and follows abnormal activities: in patrols or remotely-operated. The robot has several advantages when compared to conventional systems (mainly composed by multiple and static cameras). For instance, it enhances the security since blind spots are virtually eliminated, it induces a psychological effect against potential criminal activities (intimidation factor) and its navigation is not influenced by external factors that could damage the system, see Figs. 3.2(a) and 3.2(b). The robot may

also be used in other areas besides the surveillance of domestic environments, such as, quality control, supermarkets, security (for access restriction and face recognition) and sports.



(a) (b)

Figure 3.2: Concept of the *EEyeRobot* in a virtual scenario - Department of Electrical and Computer Engineering of the Faculty of Engineering of the University of Porto.

Figures 3.2(a) and 3.2(b) show the *EEyeRobot* concept in a virtual scenario; however, a real robotic prototype was designed, developed and installed in this real environment. The environment is depicted in Fig. 3.3 and it is a long corridor with several homogeneous regions (absence of texture). The scene is poorly illuminated and have doors that influence the local brightness. In addition, three glass walls cause photometric effects: a swimming pool reflects natural light through the middle glass wall since the room beyond the glass has several windows. These effects appear during the afternoon and turn the motion detection more challenging since illumination changes and sudden reflections make the visual analysis even more difficult. The environment is a realistic case of study with several uncontrolled illumination issues.

Computer vision is a challenging research field for small mobile robots because of the vehicle itself. The first issue is related to the limited space that is available onboard for the deployment of computer units and sensors. Other issue is related to the power consumption that enforces the autonomy of such robots. This limits the computational capability which has a direct influence in the performance of navigation and sensing procedures. In addition, most of these procedures cannot achieve the real-time constraint that is imposed by mobile systems without resorting to specialized computers. Specialized hardware is usually employed for the onboard processing of vision-algorithms since it fulfills the demand for real-time. However, these computer devices cannot be used in all situations because of the small size of vehicles or the higher consumption of energy that reduces the autonomy of robots. Thus, embedded processing units to provide a real-time

Figure 3.3: The environment where the *EEyeRobot* performs the surveillance activity. It is a long and strait corridor with five doors and tree glass walls.

capability for complex algorithms of vision computing are expense and energy-inefficient for an everyday use. Therefore, another option is to stream image sequences over the network and for a device with higher computational power. This external device sends the result back to the robot which makes it possible to decrease the computation capability of the embedded computer unit without affecting the demand for real-time. This approach can be applied if a network infrastructure is available on the environment where the robot operates, which is true for most of indoor environments. This last approach was preferred for the *EEyeRobot* and, therefore, this chapter presents a distributed software architecture that perceives and controls the activity of the mobile robot.

In this way, the contributions of this chapter include:

1. A novel robotic application for active surveillance called, the *EEyeRobot* . This robot is able to autonomously detect external motions while moving and using a monocular visual system;

2. A software architecture for controlling the robot based on a distributed computation which is formed by the onboard and the station applications;

3. An efficient scheme for streaming image sequences, and to be used by robotic applications with low computational power: achieves a frame transfer ratio of 25fps for 640×480 images in a domestic IEEE 802.11n network;

4. An architecture for visual motion perception. It has two operating modes that are triggered according to the movement of the robot: static perception and dynamic perception.

3.2 Robotic platform

The *EEyeRobot* uses a monocular camera to acquire information about the environment, it reports security issues and autonomously navigates along the rail with a visual motion detection capability (observations in motion). The robot could have several mounted sensors but this research focuses on the odometry and the monocular camera. These two sensors make possible the robot to gather sensor information in a spatial context: it is capable of detect and analyze external moving objects. Moreover, the navigation of the *EEyeRobot* is simple because it is not influenced by the presence of obstacles that could damage the robot. The first technological challenge of the robot is related to the limited space that is available onboard for the deployment of computer units and sensors. Another challenge is related to the power consumption that enforces the autonomy of the robot because it limits the computational capability. Both issues have a direct influence on the performance of navigation and sensing procedures. The solution for the *EEyeRobot* is to stream image sequences over the network and to a device with higher computational power. This external device sends the result back to the robot which decreases the computational power of the embedded computer unit while maintaining the demand for real-time. The hardware architecture of the robot can be seen with more detail in Fig. 3.8.

Figures 3.4(a) and 3.4(b) show the *EEyeRobot* prototype. In Fig. 3.4(a), the rail is placed in a corridor at the Department of Electrical and Computer Engineering of the Faculty of Engineering of the University of Porto. The rail framework provides a good solution to monitor corridors, medium or large retail outlets and distribution centers. This research focuses on the indoor surveillance environment depicted in Fig. 3.3; however, the robot can be used in different contexts and purposes. Nevertheless, it is a small robotic system based on vision computing and, therefore, complex and time demanding estimation are conducted outside the vehicle. In this way, the software architecture of the robot is formed by the onboard and the station applications, Fig. 3.6(a) and 3.6(b). This distributed architecture makes it possible to reduce the energy consumption and satisfies the requirements for real-time that otherwise would be difficult to achieve.

At first sight, the robot must be able to perceive and to interpret its surrounding environment in order to act properly according to the abnormal situations that it is facing. Figure 3.5(a) shows that a good perceptual system makes possible the robot to conduct

(a) (b)

Figure 3.4: Concept of the *EEyeRobot* in a real scenario - Department of Electrical and Computer Engineering of the Faculty of Engineering of the University of Porto. 3.4(a) gives a perspective below the robot where a rail on the ceiling allows the robot to move stealthy along the scene, increasing the flexibility and coverage of the surveillance. 3.4(b) shows the robot in surveillance operations (with zoom in).

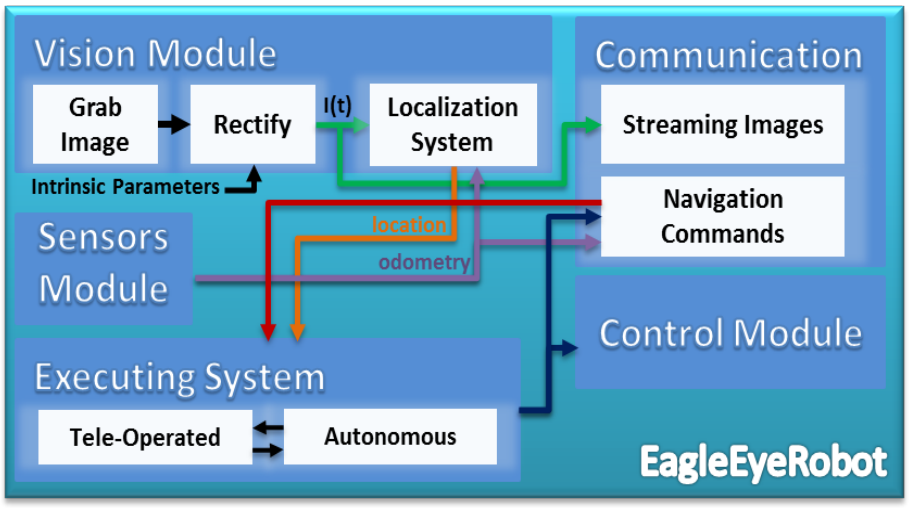
(a) (b)

Figure 3.5: The 3.5(a) depicts the interaction scheme between the environment and the robot. The robotic prototype is performing active surveillance in 3.5(b).

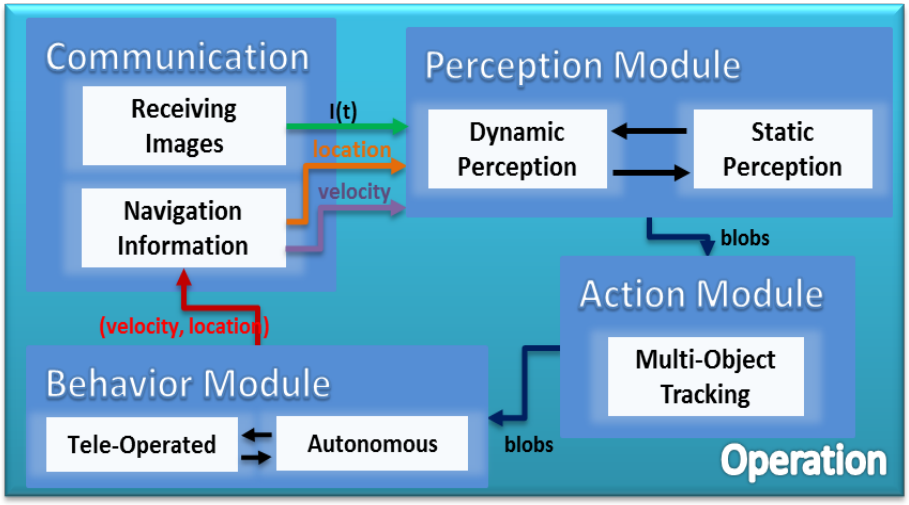
surveillance operations since the module provides the analysis of the environment. The module of motion perception is studied in detail but the action module is out of scope of this research. Although, the action module can be performed by conventional motion tracking techniques: to generate motion references for the robot according to the stimulus of the environment. A simple but effective tracking technique is briefly introduced in section 3.3.3, please consult the research [19] for additional information.

3.2.1 Distributed software architecture

The software architecture of the *EEyeRobot* is formed by two applications: embedded application and operating application. The first application runs in the mobile robot while the second runs inside an operator station.



(a) embedded application.



(b) operating application.

Figure 3.6: Software architecture of the *EEyeRobot* - 3.6(a) and 3.6(b) are the diagrams for the embedded and operating station, respectively.

3.2.1.1 Embedded application

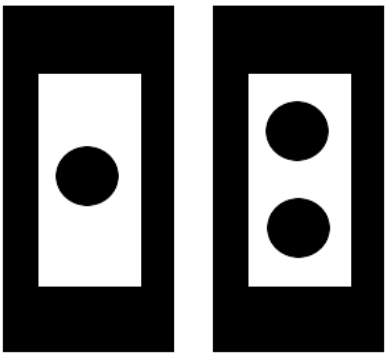
Figure 3.6(a) depicts the functional diagram of the application that is embedded in the robot and it is responsible for gathering information about sensors, for instance, encoders

and camera, and for controlling the movement of the vehicle. Internally, the application is divided into several modules: vision, sensors, control, communications and the executing system. The sensors module processes the data received from the sensors while the vision module grabs and rectifies the frame. This compensates radial and tangential distortions. In addition, the same module localizes the vehicle by detecting artificial landmarks that are placed in the environment along the rail.

A global localization method based on artificial landmarks is used to retrieve the location of the robot. In this approach, artificial landmarks are placed on the walls of the environment, and at positions that are known in advance. Thus, the localization of the vehicle is accomplished by extracting its relative position. Localization based on artificial landmarks are commonly used in industrial and domestic environments [117] due to its simplicity and effectiveness during the estimation of the robot's location. This type of approach is suitable since it is a fast procedure and does not require large computational resources. Unlike other robotic applications [117, 118], the robot presented in this thesis does not have a tight accuracy requirement for the estimation of its position and so, only a small number of markers is needed for the environment (the odometry gives an estimate of the robot's the position during its navigation, which is updated when the localization system detects a landmark). Figures 3.7(a) and 3.7(b) are examples of the landmarks used to localize the robot. They are quite simple however, they have a good contrast relatively to the environment and enable the computation of the distance between the camera and each landmark (also known as depth).

After the rectification of the lens' distortions, the frame is encoded[2] and sent over a 802.11 network through the communication module (for video streaming). The scheme for streaming image sequences that was designed for robotic applications with limited computer capabilities is described in [24]. Additionally, the communication module handles with the transmission and reception of navigation data. The *EEyeRobot* has two behaviors: autonomous and remotely-operated. A set of navigation procedures controls the movement of the robot according to what is desired and configured in the operating station. The executing system receives the navigation information, and according to the behavior and the action that are desired for the robot, it sends commands for the module that controls the movement the robot along the rail. Moreover, a set of low-level navigation procedures was implemented in the robot to create a certain level of autonomy from the operator station and to avoid possible sabotage acts. These procedures ensure that the robot is always in a safe state: communication faults, end of rail and motion detection based on external sensors.

[2]The JPEG-encoding is used to reduce the communication data and the time spent during the communications (real-time requirements).



(a) marker M1. (b) marker M2.

Figure 3.7: Localization markers.

3.2.1.2 Operating application

An external application is responsible for remote operating the robot and for understanding the visual information obtained during the autonomous behavior, Fig. 3.6(b). The application has the following modules: communication, behavior, perception and action. The first module communicates with the robot over the wireless network. It receives the image sequences and the navigation information, and sends commands to the robot. The behavior module setups the control mode of the *EEyeRobot* (autonomously-operated or remotely-operated). Surveillance-oriented features are available on the *EEyeRobot* software and incorporate the detection of an intrusion: automatically sends an e-mail over the Internet with the current image. In addition, it resorts to external sensors of the robot to detect abnormal situations of the environment, for instance, flames, smoke, liquefied petroleum gas, butane, propane, methane, alcohol hydrogen and natural gas. For an autonomous operation, the robot conducts a pre-defined patrol over the environment and evaluates the possibility of an intrusion. The visual detection of an intrusion changes the action of the robot because it will try to follow the source of the abnormal activity.

The perception module is the most important in terms of scientific relevance since it is formed by several algorithms of vision computing: spatio-temporal filtering, perception, segmentation and analysis of motion. This module has two working modes that are triggered when the robot is standstill or it is moving: static and dynamic perception, respectively. The static perception mode detects and extracts motion information based on conventional motion analysis techniques, namely, background subtraction. On the other hand, the dynamic perception mode extracts and analyses motion information when the robot is moving along the rail and based on dense flow fields. Finally, the action module receives a set of blobs that describes motion profiles of potential intrusions; and these

profiles are used by a multi-object tracking technique that was proposed in [19].

3.2.2 Hardware architecture

Figure 3.8 represents the hardware structure of the robotic solution presented in this thesis. An operating station makes it possible to monitoring the robot's activity since it processes the cognitive behavior of section 3.3: the navigation procedures are embedded in the control unit of the robot and the high level algorithms are implemented in the operator station. A wireless network enables the communication between the control unit of the robot and the control station: video streaming, navigation commands and information about the robot's status.

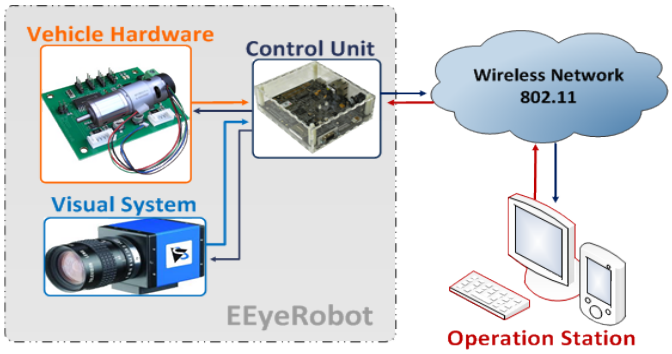


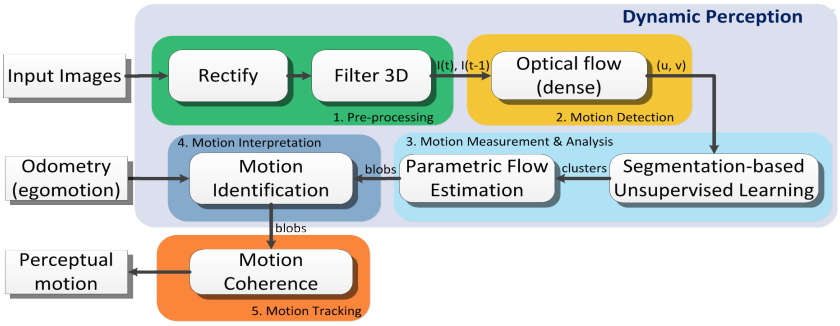
Figure 3.8: The hardware diagram of the *EEyeRobot* .

3.3 Architecture for visual motion perception

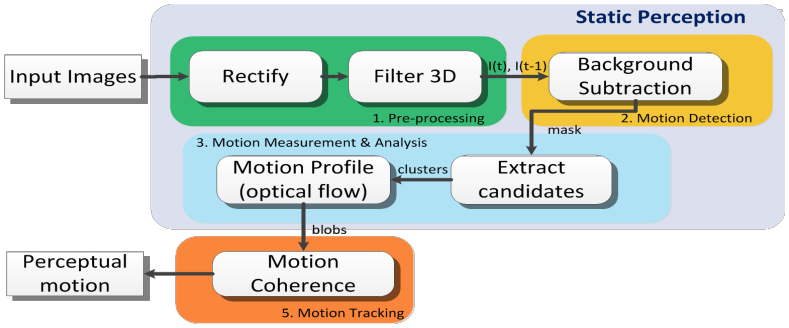
The technical development of the module for visual motion perception of the *EEyeRobot* was conducted with two distinct modes: dynamic and static visual perception. Scientifically, this thesis is focused in the dynamic perception mode whose overview is given in section 3.3.1; in addition, the static perception mode is presented in section 3.3.2, and the technique for tracking motion that is used in the action module is introduced in section 3.3.3.

3.3.1 Dynamic visual perception

The dynamic perception mode of the *EEyeRobot* is depicted in Fig, 3.9(a). As it can be confirmed, it is an extensive and complex set of sub-algorithms. The entire process is divided into several hierarchical layers that represent a motion perception architecture for moving observers: detection, measurement and cognitive layer. The dynamic perception



(a)



(b)

Figure 3.9: The architecture of motion perception - 3.9(a) and 3.9(b) are diagrams for the dynamic and static visual motion perception, respectively.

mode is presented in the next chapters of this thesis: the researching works [4, 20, 22] have demonstrated interesting results for the vision techniques that were developed especially for this mode. Figure 3.9(a) shows a logical diagram for the entire visual motion perception for a moving observer. A spatiotemporal filter [20] enhances the quality of image sequences without smoothing the object's edges since the temporal contribution is small (see chapter 4). The next stage is related to the perception and segmentation of motion. The optical flow technique [4] presented in chapter 5 uses sequences of images and obtains dense flow fields. Then, regions of the image with different motion characteristics can be extracted from these flow fields. This extraction is based on a segmentation technique [22], whose challenges are discussed in chapter 6. The *EEyeRobot* moves along a rail which makes it possible to use the odometry information with the results of the segmentation in order to understand what kind of external moving objects are currently present in the environment. This means that, the egomotion of the robot is identified from the segmentation results in a manner that makes it possible to infer about the regions of the image that are associated to external moving objects (number, direction and magnitude of the visual motion). The identification of the egomotion is performed by resorting to the odometry and the depth of the scene (estimated by the localization system since the posi-

tion and size of the landmarks are known in advance). The mathematical formulation for the identification of the egomotion follows the fundamentals presented by *H.C Longuet-Higgins and K. Prazdny (1980)* [88] and it was already discussed in the previous chapter. After compensating the egomotion, blobs are described by a motion profile characterized by a statistical and parametric affine model. The temporal coherence of these blobs is evaluated by the action module that is briefly presented in section 3.3.3. It makes the behavior module that controls the robot substantially more robust to the presence of outliers (blobs) and enhances the quality of motion analysis. The dynamic architecture that is proposed for the moving robot is focused on the analysis and segmentation of different types of motion (parametric flow estimation) based on dense optical flow fields.

3.3.2 Static visual perception

Figure 3.9(b) shows a diagram that describes the visual perception of the *EEyeRobot* when it assumes a static positioning. The first step is similar to the architecture of the dynamic perception; however, subsequent phases (motion detection, measurement and analysis) are quite different because when a static observer captures the scene, every spatial and temporal variation represents part of the moving object (neglecting illumination changes and noise). Therefore, motion detection is conducted by a background subtraction technique based on Mixture of Gaussians (MOG) [99]. The segmentation of moving objects is performed through a subtraction that defines the classification of each pixel between foreground and background. This segmentation result is used to extract a set of blobs candidates using features, for instance, the size and geometrical connectivity. Then, the motion profile of the blob is computed from a dense estimation of the optical flow field (limited to the region of interest - flow signature). This flow signature is formed by mean and standard deviation of flow vector represented in Polar space, and is computed using a RANSAC (RANdom SAmple Consensus) approach. The tracking method of the fifth phase (motion tracking) uses flow signatures to track multi-objects. Therefore, the presented architecture for motion perception with a static observer is more similar to conventional surveillance applications because images are obtained by a fixed camera placed on the environment.

3.3.3 Action module

The final stage of the architecture for motion perception is tracking the moving objects. A diagram of the action module is depicted in Fig. 3.10 and the proposed technique is named MTMP (Multi-Tracking of Motion Profiles) [19]. The method computes motion

profile of the moving objects, and resorts to multi-Kalman filters. This ensures a multi-tracking capability and a high computational efficiency. A Kalman filter tracks a single blob (called tracker) which is defined in terms of its states, motion model of constant velocity and measurement (centroid positioning) equations. The entire process is divided into three phases: association, actualization of trackers and prediction/correction.

Figure 3.10: Diagram of the multi-object tracking method.

The method resorts to three type of trackers, namely, young, active and older. Each tracker has associated one blob (motion profile) and a temperature value. The idea beyond the temperature is similar to the simulated annealing [26]; however, it is used to avoid the influence of outliers since only trackers with a well-established temperature (active trackers) are considered by the high level procedures (in this case, the behavior module of the *EEyeRobot*). A new tracker is created with a small temperature when the motion profile of all blobs does not match to the existing trackers. A tracker improves its solution causing an increasing of temperature whenever a detected blob has a similar motion profile and geometrical features, for instance, centroid position and size. Trackers having the higher temperature values are considered as active trackers; however, they are converted to older trackers if their temperature is reduced to below a threshold value. Finally, a older tracker is reconverted to active when the temperature rises again, or it disappears if a minimum value is reached. The first stage of the tracking method is the association phase since it analyses the current set of observed blobs, and finds trackers with similar motion profiles. The distance between motion profiles are computed using a similarity measurement and a similarity matrix (distance in terms of motion between trackers to blobs). This phase assumes that the observation is multivariate and normally distributed, and the feature vector has two dimensions since it is formed by the flow vector in Polar coordinates $\mathbf{w}^p$. Hence, the difference between blob and tracker, c_s and c_r, can be measured by a Mahalanobis squared distance of samples. The similarity between both is considered in terms of a normalized difference of the mean vectors, and is presented in detail in section 6.3.3.2. The association operation is an iterative process that takes into account the similarity of motion profiles, and the geometrical (Euclidean) distance of the size and centroid positions. This process is stopped when the dissimilarity value

is high. Next, trackers are updated according to the previous association. This rises the temperature for the trackers with new blobs (motion profile is also updated) and decreases the temperature for the remaining. In addition, trackers are classified in young, active or older according to their temperature value. For the final stage, the Kalman filter is updated according to the centroid position or is predicted for the non-updated trackers. Therefore, this tracking method returns the perceptual motion which is the set of active trackers, and is characterized by the confidence level, geometric features and motion profiles.

3.4 Final considerations

This chapter presented a novel autonomous mobile robot that performs active surveillance. The major advantage of the *EEyeRobot* relatively to conventional systems is its ability to perform surveillance procedures without crowding the environment with cameras. The architecture that controls the robot is a distributed application and the perceptual architecture of the robot has two operating modes that are triggered according to the vehicle's motion: static perception and dynamic perception. Motion analysis with moving observers is a challenging problem in computer vision and robotics. Therefore, this research is focused in the dynamic perception and, as it is possible to confirm, the proposed architecture is an extensive and highly complex set of sub-algorithms. The entire process is divided into several hierarchical layers that represent: the detection, the measurement and the cognitive layer. This chapter proposes a suitable architecture for motion analysis based on moving observations. In addition, an object tracking is briefly introduced (since it is out of the scope of this thesis). The method is quite simple and it goal is to prove that the estimation of motion profile provides a relevant information for complex motion analysis techniques, especially, for tracking moving objects. The technique exhibits a good computational performance since it takes less than 5 milliseconds to compute.

In short, an innovative robot is designed, developed and subsequently used to validate the methods presented in this scientific work. The *EEyeRobot* is a small robotic application based on vision computing and, therefore, complex and time demanding computations are conducted outside the robot. This reduces the energy consumption and satisfies the requirements for real-time, that otherwise would be difficult to achieve. Low-level navigation procedures were implemented on the embedded application in order to create a certain kind of autonomy from the operating station and to ensure that the robot is always in a safe state: communication faults, end of the rail and the detection of movement through motion detection sensors to avoid possible sabotage acts.

Chapter 4

The Robust Bilateral and Temporal Filter

Over the last few decades, surveillance applications have been an extremely useful tool to prevent dangerous situations and to identify abnormal activities. Although, the majority of surveillance videos are often subjected to different noises that corrupt structured patterns and fine edges. This makes the image processing methods even more difficult, for instance, object detection, motion segmentation, tracking, identification and recognition of humans.

This chapter proposes a novel filtering technique named Robust BilateraL and Temporal (RBLT), which relies on spatial and temporal evolution of sequences to conduct the filtering process while preserving relevant image information. A pixel value is estimated using a robust combination between the spatial characteristics of the pixel's neighborhood and its own temporal evolution. Thus, robust statics concepts and temporal correlation between consecutive images are incorporated together which results in a reliable and configurable filter formulation that makes it possible to reconstruct highly dynamic and degraded image sequences.

The filtering is evaluated using qualitative judgments and several assessment metrics, for different Gaussian and Salt-Pepper noise conditions. Extensive experiments considering videos obtained by stationary and non-stationary cameras prove that the proposed technique is capable of filtering corrupted sequences. The distorted images obtained by the proposed method achieve a better perceptual quality when compared to the bilateral filter and the spatiotemporal versions of the Gaussian average and the median filter.

The chapter[1] is organized as follows. Section 4.1.1 presents a brief review of the latest spatiotemporal filtering techniques. Sections 4.1.2 and 4.1.3 introduce the Gaussian

[1] Some portions of this chapter appeared in [20].

and bilateral formulation, respectively. Section 4.2 presents the proposed RBLT filtering technique in detail: section 4.2.1 shows the formulation based on robust estimation methods and section 4.2.2 describes the incorporation of the temporal contribution. Section 4.3 presents the experimental results of the RBLT: the proposed method is compared to state-of-the-art video denoising techniques in section 4.3.2. Later, a set of experiments is conducted in order to evaluate the performance of the RBLT under realistic surveillance videos: section 4.3.3 shows the filtering results of videos captured by a stationary surveillance application; section 4.3.4 presents the filtering results for videos obtained by a mobile robotic-based surveillance system. Finally, the major conclusions of this research are presented in section 4.4.

4.1 Introduction

Automated surveillance systems usually employ stationary sensors to monitor the environment; however, the number of research works that propose surveillance applications based on non-stationary cameras is increasing. The presence of noise in videos affects subsequent image processing phases, such as three-dimensional reconstruction, registration, classification of objects, motion segmentation and analysis, tracking, identification and recognition of humans. Thus, denoising is an extremely important pre-processing phase that is used to improve the perceptual appearance of images; however, a trade-off between noise reduction and data preservation is important to enhance the characteristics of images that are relevant for high level algorithms.

Despite recent improvements in the color filter array [119, 120], realistic surveillance sequences often have a low signal-to-noise ratio (SNR) [37] since they are commonly corrupted with noise that can be approximated by Gaussian and Salt-and-Pepper noise [121]. This chapter presents an image denoising technique, called Robust Bilateral and Temporal filter (RBLT) that meets the visual requirements of a surveillance system based on a mobile robot. This name was chosen because the technique follows and enhances the bilateral filter (BL) formulation of Tomasi and Manduchi [122]. The original formulation [122] is completely reformulated using robust estimation principles, for instance, non-quadratic estimators are incorporated into the filter formulation. This makes the filter more robust to the presence of outliers (usually, noise components) while preserving structural information of the image sequence. These two aspects are the major requirements for a denoising technique used in robotic-based surveillance systems. In addition, a temporal component is also incorporated into the filter formulation, which increases the filter's ability to remove strong noise components. The major advantage of using spatial

and temporal information in videos is the possibility of achieving filtering performances that otherwise would hardly be obtained. This means that temporal information can be used to infer the noise component that corrupts the current image if there is a reliable correlation for the brightness evolution of pixels between consecutive images. Therefore, the RBLT measures the reliability of temporal information of the brightness evolution and uses spatial information about the neighbors, which are weighted using non-quadratic norms to evaluate the similarity and to estimate the free-noise value of each pixel.

Contributions of this chapter include:

1. An innovative spatiotemporal filtering technique that can be used by stationary and non-stationary surveillances or robotic applications;
2. A measurement of the photometric similarity based on robust error norms;
3. A temporal filtering component based on the *temporal coherence* assumption with a self-evaluation mechanism to detect and treat violations of this assumption;
4. A filtering technique with a performance less influenced by outliers and by the type of noise that corrupts the sequence;
5. Filtering with a better trade-off between noise reduction and data preservation which is especially recommended for denoising images with low SNR. Thus, the filter does not create ghosts or strange artifacts in the denoised image that compromise the segmentation process;
6. A filtering technique that enhances videos corrupted by Gaussian and Salt-and-Pepper noise, in a very competitive manner when compared to other state-of-the-art techniques (especially designed for each type of noise);
7. Extensive qualitative and quantitative evaluation by considering several baseline filters.

The experimental results include the analysis of the proposed denoising technique in several contexts: a comparison to state-of-the-art methods, and an evaluation and discussion of the behavior of the RBLT in real and practical surveillance applications. Therefore, important conclusions are obtained about the usefulness of the filter for different types of videos and noise. In previous experiments, the bilateral filter and the spatiotemporal versions of the Gaussian average and the median filter are considered baseline methods. The performance of the RBLT method is evaluated using sequences corrupted by Gaussian and Salt-Pepper noise. The quality of distorted images is validated using subjective visualizations and several objective assessment metrics, namely, the root mean square error

(RMSE), the signal-to-noise ratio (SNR), and especially the peak signal-to-noise ratio (PSNR) and the Structural Similarity (SSIM). Experimental considerations indicate that filtering corrupted sequences using the "robustification" of the temporal correlation between consecutive images is computationally rewarding and represents an alternative to the state-of-the-art techniques. The filtering properties and edge preserving capabilities of the proposed filter can lead to a potential accuracy enhancement of motion segmentation and contribute to future developments in automated surveillance applications.

4.1.1 Related works

In the literature, many researchers have proposed denoising methods based on the average filtering, median filtering [6, 123], non-linear diffusion [124], non-linear total variation [125, 126], non-local means filters [127], wavelet [128, 129], multi-scale filtering [128, 130], morphological operators [131, 132] and bilateral filtering.

The bilateral filtering is commonly used in computer vision due to its denoising properties and ability to preserve data. Several bilateral-based formulations have been proposed [133, 130, 134, 123] over the last few years. In fact, an extension of the bilateral filter based on multi-resolution with wavelet transform sub-band mixing is proposed by *Zeinab and Yasser (2011)* [128]. Their technique achieves better results than the BL filter. The improvement was $0.1029dB$ (SNR), $1.1292dB$ (PSNR) and 3.3809 (mean squared error - MSE) for a Gaussian noise with a standard deviation equals to five. *Shyam Anand and Sahambi (2012)* [129] use the undecimated wavelet transform to obtain the coefficients. Then, a bilateral filter is applied to the transformed approximation in order to preserve relevant edges and to remove the noise. The results obtained by the technique are visually acceptable since the denoising level appears to be high. However, their research fails at performing a qualitative comparison between other filters and, therefore, it is difficult to quantify the improvement of the proposed approach. *Liu et al. (2006)* [135] have demonstrated that the domain parameter of the bilateral filter should be adjusted to $1.95\sigma_n$ in order to yield more satisfying results, where σ_n is the noise level. An effectiveness evaluation of the bilateral denoising filter is presented in [136]. The major contributions of this work are the experimental results since they prove that the bilateral filter is able to suppress noise without smoothing high resolution details. The bilateral performance is evaluated in 3D reconstructions. Visually, the BL has a denoising and reconstruction capability which is highly recommended in computer vision.

A direct implementation of the bilateral filter consists of two loops over all the pixels [137]. The complexity of this algorithm is $O(|\Omega|^2)$, where $|\Omega|$ is the number of pixels. Some authors tried to increase the performance of the BL filter. For example, Paris and

Durand (2006) [134] express the BL filter in a higher-dimensional space, named bilateral grid. The gray-level image is represented in a volumetric data structure that downsamples the data. Thus, a close solution is obtained in a discretized space-color grid because the bilateral filter is approximated to a 3D Gaussian convolution applied to the grid, followed by a tri-linearly interpolation and normalization of the pixels. Thus, the denoised image depends on the subsampling grid phase, which means that the discretization causes loss of precision particularly in high dynamic range images. *Weiss (2006)* [123] proposes a histogram-based implementation for the bilateral filter. The work describes an efficient way of computing the histogram of the neighborhood of a pixel by exploiting the similarity with the histogram computed for the adjacent pixel. The fundamental concept is that the neighborhoods of two adjacent pixels largely overlap [137]. This approach exhibits a complexity of $O(|\Omega| \log r)$ where r is the (spatial) filter size r. The downside of this algorithm is that it is limited to rectangular spatial kernels and box filters (creating imperfect frequency responses). Later, the complexity of the BL filter was reduced to O(1) in [138] and [139]. *Porikli (2008)* [138] proposes three methods to compute the BL filter in constant time. The first uses integral histograms to avoid the redundant operations and interactive speed is achieved by quantizing the corrupted image using a small number of bins. Then, the bilateral filter is computed with box spatial and arbitrary range kernels. The second method reformulates the BL filter as a weighted sum of the spatial filtered responses of the powers of the input image. The final method computes the filter response using Taylor series of linear filters to approximate to the Gaussian range function up to the four order derivatives. The bilateral filter is divided into a number of constant time spatial filters in [139] and, thus, the filter can be implemented in parallel which makes it possible to achieve a real-time computation.

Spatiotemporal filters are a natural evolution of the image filters since videos can be contaminated by several types of noise, for instance, Gaussian noise, impulsive noise and quantization noise [121]. Different spatiotemporal approaches can be found in the literature [140, 121, 141, 6, 142] and they can be classified as: a pixel domain technique (the denoising is done by a weighted averaging) or a transform domain technique (the denoising is conducted in a different space representation followed by an inverse transformation that is performed in the end in order to convert the space back to the pixel domain).

The non-local means filter (NLM) is extended to image sequences in [143]. A 3D search window through the sequence makes it possible to compute the weights that measure the dissimilarity between the patch centered at the corrupted pixel and that centered at the contributing pixels. Thus, the method restores every noisy pixel by the weighted average of all pixel intensities in the neighborhood. A technique called wavelet-domain denoising filter based on spatiotemporal Gaussian scale mixture model (STGSM) was

presented by *Varghese and Wang (2010)* [5]. This method estimates the global motion since it considers that the local motion is unreliable and would significantly change the noise statistics. The motion compensation is applied to past and future images of the sequence. The wavelet is applied to the current image that divides the signal into several sub-bands (the coefficients are computed from a spatiotemporal neighborhood). Then, a spatiotemporal Gaussian scale mixture filtering technique (based on the Bayesian least square estimation) is employed and the inverse wavelet transform makes it possible to obtain the denoised image. The major drawback of the STGSM is that the denoising process of the current frame involves the adjacent past and future frames (unknown in live applications).

Dai et al. (2013) [121] propose the generalized multi-hypothesis motion compensated filter (GMHMCF). The method combines the concept of the time-recursive filter and non-recursive filter since the frames used by the GMHMCF consist of the denoised previous frames as well as the noisy future frames. The temporal correspondence between neighboring frames is established by a noise-robust motion estimation with a predefined motion vector regularization term to construct multiple temporal hypotheses. The denoised frame is obtained using a linear optimal estimator that aggregates all the associated blockwise estimates by a weighted average. The performance of this method for denoising color videos corrupted with Gaussian noise was superior comparative to other spatiotemporal filters, for example, the STGSM and the NLM. Finally, the *Nawal Benmoussat, Faouzi Belbachir and Beloufa Benamar (2007)* [144] propose a filtering scheme for video sequences. The filtering is conducted by computing the optical flow (block matching algorithm) to compensate the motion as a temporal prediction step of previous frames (already denoised). Therefore, the current noisy frame can be reconstructed using its previous neighbor frame and the motion vector field. The results show that the proposed filtering technique alleviates the computation load by about 80% comparative to the classical motion-compensated filtering (adaptive weighted averaging filter) and for a comparable visual image quality.

4.1.2 The Gaussian filter

The Gaussian filter is a low-pass filter that smooths the image by replacing the current value of the focal pixel ($\mathbf{x} = (x, y)$) with the weighted sum of their neighbors. *Spatial coherence* assumption (see definition in page 21) is considered for the filtering, which means that images usually vary slowly along space, for instance, neighbor pixels belonging to the same surface are likely to share similar values. The weighting function used to

denoising the image is based on a Gaussian distribution, where each near pixel is evaluated according to its distance from the focal pixel, Eq. 4.2. Thereby, pixels further away from the focal pixel will have a weaker contribution.

Thus, a filtered image is obtained by convolving the input image, $I_{src}(\mathbf{x})$, with a Gaussian kernel, $r_\sigma(\mathbf{x})$. Mathematically,

$$I_f(\mathbf{x}) = r_\sigma(\mathbf{x}) * I_{src}(\mathbf{x}) = \frac{\sum_{i,j=-\infty}^{+\infty} r_\sigma(i,j).I_{src}(\mathbf{x}_{ij})}{\sum_{i,j=-\infty}^{+\infty} r_\sigma(i,j)}, \quad \forall \mathbf{x} \in \Omega; \tag{4.1}$$

$$r_\sigma(i,j) = exp\left(-\frac{i^2+j^2}{2\sigma^2}\right), \tag{4.2}$$

where $\mathbf{x}_{ij} = (x-i, y-j)$ and σ is the standard deviation that specifies the distance over which the weights are significant. The Ω means the image domain formed by N-columns and M-rows. The denominator of Eq. 4.1 is a normalizing factor that converts the sum of weights into a proper average and, therefore, it maintains the pixel range.

The major disadvantage of the Gaussian filter is its inability to discriminate boundaries, since the *spatial coherence* assumption fails at the edges. This usually results in the loss of edge details which are consequently blurred by the low-pass filtering.

4.1.3 The Bilateral filter

The bilateral filter was presented by *Tomasi and Manduchi (1998)* [122] and it is a non-linear filter that extends the concept of Gaussian filter. The denoising is also computed as a weighted average of each pixel's neighborhood; however, it is more selective relative to the neighbors that are allowed to contribute to the weighted sum. Instead of considering only the geometric closeness, given by Eq. 4.2, each neighbor contribution is measured according to their geometric closeness and photometric similarity.

The photometric similarity function, often referred to as domain, is formally defined by equation 4.3, and measures the similarity intensity difference between the focal pixel and its neighbors. For instance, when the intensity difference between the focal pixel, $I_{src}(\mathbf{x})$, and a near pixel, $I_{src}(\mathbf{x}_{ij})$, is large, the function yields a small weight.

$$d_{\sigma_d}(\mathbf{x}_{ij}, \mathbf{x}) = exp\left(-\frac{\|I_{src}(\mathbf{x}_{ij}) - I_{src}(\mathbf{x})\|}{2\sigma_d^2}\right). \tag{4.3}$$

This selective incorporation of pixel intensities in a weighted sum is achieved by multiplying the geometric closeness function, r_{σ_r}, with a photometric similarity function, d_{σ_d},

see equation 4.4.

$$I_f(\mathbf{x}) = \frac{\sum_{i,j=-\infty}^{+\infty} d_{\sigma_d}(\mathbf{x}_{ij},\mathbf{x}).r_{\sigma_r}(i,j).I_{src}(\mathbf{x}_{ij})}{\sum_{i,j=-\infty}^{+\infty} d_{\sigma_d}(\mathbf{x}_{ij},\mathbf{x}).r_{\sigma_r}(i,j)}, \quad \forall \mathbf{x} \in \Omega. \tag{4.4}$$

Equation 4.4 shows that each pixel value is replaced by an average of similar and nearby pixel values [122]. In other words, image regions are smoothed by the pixel values that share similar values and belong to the same neighborhood.

Thus, the bilateral filter has two free parameters, σ_r and σ_d. These parameters make it possible to control the influence of the geometric distance and similarity properties of the neighbors, respectively. The domain parameter, σ_d, specifies the difference of intensity that belongs to the same object surface and, therefore, it enables to detect the presence of the edges. The last weight component depends on the local density distribution, turning the bilateral filter into a non-linear filter.

The major advantage of this filter is that it smooth the noise while maintaining the structural information of edges.

4.2 The Robust Bilateral and Temporal Filter (RBLT)

The estimation of the pixel value, performed by a common filter, may suffer from the presence of outliers. Consequently, the estimation may be poor and the noise subtraction unreliable because the formulation of the filter is incapable of distinguishing between inliers and outliers in images with a low signal-to-noise ratio (SNR).

For the Gaussian filter, texture information on the image is combined over the geometric space which causes a significant smoothness across boundaries. Incorporating a factor to evaluate the domain of each pixel's neighbor is not enough to originate a sufficiently robust estimation in the presence of outliers. In cases were the image has a strong noise component or if the noise cannot be accurately approximated by a Gaussian distribution, since it assumes that the noise is also an isotropic function, the bilateral filter leads to a poor performance, see section 4.3.

Therefore, this research extends the bilateral principle in two fronts:

1. Robust static techniques are used to define the domain function. It makes the filtering process more accurate because the formulation is robust to the presence of outliers, even for images with a low SNR and unknown noise distribution;

2. *Temporal coherence* assumption is used. It makes it possible to assume that a pixel brightness value at some coordinate position may be temporally correlated. This assumption is true if small temporal increments are considered between consecutive

images of the sequence. Therefore, the neighborhood concept used by the bilateral formulation [122] is temporally extended, which originates a new contribution factor that measures the temporal evolution of the pixel value.

◇ **Definition 12:** *Temporal coherence - embodies the assumption that temporally adjacent pixels are persistent in sequential time instants. In practice, it assumes that the results at the current frame are correlated to the ones observed at previous time instants. It is directly related to the notion of temporal persistence.*

4.2.1 Robust estimation

This section presents a brief description of robust estimation principles.

The major objective of a regression technique is to resolve the model parameters that produce the best fit for a region of n independent observations. For example, considering the following linear model it is possible to highlight that:

$$t_i = \beta_0 + \beta_1 z_{i1} + \beta_2 z_{i2} + \ldots + \beta_d z_{id} + \varepsilon_i,$$

$$\Leftrightarrow t_i = \mathbf{z}_i \beta + \varepsilon_i,\ \forall i \in \{1,2,\ldots,n\}, \tag{4.5}$$

where t_i is the unknown signal, $\mathbf{z}_i$ is the observations, β is the model arguments and ε_i corresponds to the noise. Given an estimator $\hat{\mathbf{a}} = (\hat{a}_0, \hat{a}_1, \ldots, \hat{a}_d)^T$ for β, the model can be fitted as:

$$\hat{t}_i = \mathbf{z}_i \hat{\mathbf{a}} + e_i,\ \forall i \in \{1,2,\ldots,n\}. \tag{4.6}$$

The residuals are given by $err_i = t_i - \hat{t}_i$, the objective function can be expressed using M-Estimators (*Maximum Likelihood Type Estimators*)[2]:

$$min_{\hat{\mathbf{a}}}\ \xi(\hat{\mathbf{a}}) = \sum_{i=1}^{n} \varphi(t_i - \mathbf{z}_i \hat{\mathbf{a}}). \tag{4.7}$$

A robust function $\varphi(err_i)$ decreases the influence of outliers by replacing the squared residuals by a robust error function. The term "robust" is related to the rejection of outliers by recovering the estimate that represents the majority of the data. The $\varphi(err_i)$ is a positive-definite, monotone (punishing large residuals) and symmetric function with a unique minimum at zero [146].

[2]M-Estimators is a family of extremum estimators in *robust statistics* that was generalized from the maximum likelihood estimator (MLE) since $\varphi(\mathbf{z}, \hat{\mathbf{a}}) = -logf(\mathbf{t}, \hat{\mathbf{a}})$ [145].

Type	$\varphi(err)$	$\psi(err)$	$\omega(err)$
Quadratic	err^2	$2err$	2
L1	$\lvert err\rvert$	$sgn(x)$	$\frac{1}{\lvert err\rvert}$
Tukey($\lvert err\rvert \leq \beta$)	$\frac{c^2}{6}\left[1-\left(1-\frac{err^2}{\beta^2}\right)^3\right]$	$err\left(1-\frac{err^2}{\beta^2}\right)^2$	$\left(1-\frac{err^2}{\beta^2}\right)^2$
Tukey($\lvert err\rvert > \beta$)	$\frac{c^2}{6}$	0	0
Lorentzian	$log\left(1+\frac{err^2}{2\beta^2}\right)$	$\frac{2err}{2\beta^2+err^2}$	$\frac{2}{2\beta^2+err^2}$
Geman-Mcclure	$\frac{err^2/2}{1+err^2}$	$\frac{err}{\left(1+err^2\right)^2}$	$\frac{1}{\left(1+err^2\right)^2}$
Charbonnier	$\beta\sqrt{1+\frac{err^2}{\beta^2}}-\beta$	$\frac{err}{\sqrt{1+\frac{err^2}{\beta^2}}}$	$\frac{1}{\sqrt{1+\frac{err^2}{\beta^2}}}$

Table 4.1: Examples of error norms for M-estimators. The robust functions are graphically depicted in Fig. 4.1(a) and the weight functions in Fig. 4.1(b). In this research, the last four norms (*L1*, *Tukey*, *Lorentzian*, *Geman-Mcclure* and *Charbonnier*) are called robust error norms.

Equation 4.7 may be solved by differentiating $\varphi(err_i)$ with regard to the argument $\hat{\mathbf{a}}$ and setting the partial derivatives equal to zero.

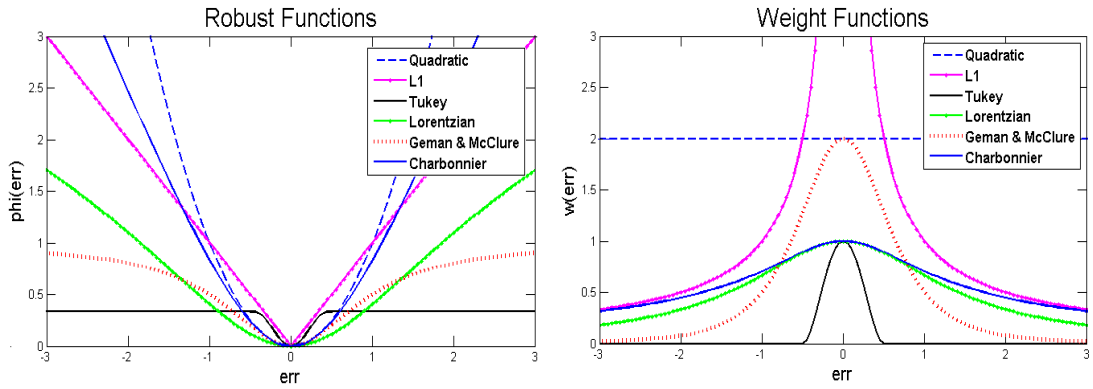
$$\frac{\partial}{\partial \hat{\mathbf{a}}}\sum_{i=1}^{n}\varphi(t_i-\mathbf{z}_i\hat{\mathbf{a}})=\sum_{i=1}^{n}\psi(t_i-\mathbf{z}_i\hat{\mathbf{a}})\mathbf{z}_i=\mathbf{0}, \tag{4.8}$$

where $\psi(err_i)$ is called influence function and is the derivative of $\varphi(err_i)$ with respect to its argument. The function characterizes the influence that a particular observation has on the solution. Finally, a weight function, $\omega(err_i)$, can be defined as:

$$\omega(t_i-\mathbf{z}_i\hat{\mathbf{a}})=\frac{\psi(t_i-\mathbf{z}_i\hat{\mathbf{a}})}{t_i-\mathbf{z}_i\hat{\mathbf{a}}}. \tag{4.9}$$

This $\omega(err_i)$ is useful for several minimization schemes, such as iterated reweighted least-squares, as it defines a weight contribution according to the residual value.

A comparison between the quadratic and robust error norms is depicted in Figs. 4.1(a) and 4.1(b). The robust error norms, presented in table 4.1, are well behaved with the exception of the L1 as it is not differentiable when $err = 0$. In addition, the *Lorentzian* norm is non-convex (negative logarithm) [147]. This may lead to a minimization procedure, usually performed by descending algorithms, to be stopped at local minimums (a unique solution is not guaranteed). The robust function has a saturation characteristic since as error increases the robust measure gets closer to a constant. This behavior can be examined by the influence function. It is especially evident for the *Geman-Mcclure* and *Tukey* robust norms that their measurements decrease the influence of outliers comparatively to



(a) (b)

Figure 4.1: Graphical representations of the robust functions in 4.1(a) and weight functions in 4.1(b) and for different error norms. These functions allow to analyze the behavior of each type of norm in the presence of outliers (points with a large residual value) and are mathematically presented in table 4.1.

the quadratic norm (see Fig. 4.1(a)). Fig. 4.1(b) clearly shows that the weight evolution for robust error norms is reduced as residual values increase, nevertheless, the weight evolution of the quadratic error norm maintains the same value. This robust behavior is extremely important to prevent estimations based on misleading data. Furthermore, this research focuses on the *Charbonnier* because it is a norm that performs well and makes it possible to weigh inliers and outliers according to the objectives of this research.

An important concept in robust statics is the breaking point which is the maximum proportion of outliers that each estimator can handle, for instance, an estimator based on mean has a breaking point of 0% (inliers and outliers have the same contribution for the estimation). Nevertheless, the median estimator can tolerate up to 50% of outliers and when this value is surpassed it cannot distinguish inliers and outliers.

The original formulation of the bilateral filter [122] relies on a Gaussian distribution to measure the photometric similarity. This provides an evaluation criterion that intends to decrease the contribution of the outliers, pixels with a large photometric distance, during the denoising procedure. Although, complications arise when the noise is not normally distributed, for instance, if it has heavier tails. The assumption that the noise can be expressed by a Gaussian rarely holds in applications based on computer vision since the surface luminance is not isotropic and there are specular highlights (ISO sensitivity).

Thus, the domain term of the bilateral formulation, presented in Eq. 4.4, is replaced by a robust error norm that is less affected by outlying neighbors. Following Eqs. 4.7 and

4.8, a robust bilateral formulation can be defined by:

$$min_{I(\mathbf{x})} \sum_{i,j=-\infty}^{+\infty} \varphi\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right), \quad \forall \mathbf{x} \in \Omega. \tag{4.10}$$

Assuming convexity of the robust formulation, which leads to a solution that converges toward to the minimum, Eq. 4.10 can be solved by differentiating the function and setting the derivative equal to zero, see Eq. 4.8.

$$\sum_{i,j=-\infty}^{+\infty} \psi\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right)\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right) = 0,$$

$$\Leftrightarrow \sum_{i,j=-\infty}^{+\infty} \psi(.)I(\mathbf{x}) - \sum_{i,j=-\infty}^{+\infty} \psi(.)I(\mathbf{x}_{ij}) = 0. \tag{4.11}$$

The focal pixel is a constant and, hence, Eq. 4.11 can be rewritten as follows:

$$I(\mathbf{x}) = \frac{\sum_{i,j=-\infty}^{+\infty} \psi\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right).I(\mathbf{x}_{ij})}{\sum_{i,j=-\infty}^{+\infty} \psi\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right)}. \tag{4.12}$$

Replacing the influence function by the weight function, $\omega(\mathbf{x}_{ij}, \mathbf{x}) = \omega\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right)$, the previous equation can be reformulated as:

$$I(\mathbf{x}) = \frac{\sum_{i,j=-\infty}^{+\infty} \omega(\mathbf{x}_{ij}, \mathbf{x}).\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right).I(\mathbf{x}_{ij})}{\sum_{i,j=-\infty}^{+\infty} \omega(\mathbf{x}_{ij}, \mathbf{x}).\left(I(\mathbf{x}) - I(\mathbf{x}_{ij})\right)}. \tag{4.13}$$

In Eq. 4.13, the weight function defines the photometric similarity of each neighbor; however, $I(\mathbf{x}) - I(\mathbf{x}_{ij})$ also measures the dissimilarity between the focal pixel and neighbor. Therefore, this last term is replaced by the geometric closeness between the focal pixel and each neighbor, $r_{\sigma_r}(i, j)$.

Finally, a robust bilateral formulation is obtained:

$$I_f(\mathbf{x}) = \frac{\sum_{i,j=-\infty}^{+\infty} \omega_d(\mathbf{x}_{ij}, \mathbf{x}).r_{\sigma_r}(i, j).I_{src}(\mathbf{x}_{ij}, t)}{\sum_{i,j=-\infty}^{+\infty} \omega_d(\mathbf{x}_{ij}, \mathbf{x}).r_{\sigma_r}(i, j)}, \quad \forall \mathbf{x} \in \Omega. \tag{4.14}$$

The weight function, $\omega_d(.)$, specifies the photometric similarity domain that is inversely proportional to the intensity difference between the focal pixel and the neighbor. It allows to exclude the contribution of the outliers from the weighted sum, independent of the type of noise that corrupts the image.

4.2.2 Temporal contribution

The robust bilateral and temporal (RBLT) filter is a temporal extension of the robust bilateral filter in Eq. 4.14, which facilitates denoising for a sequence of images with low SNR. It assumes temporal correlation for the pixel brightness value (*temporal coherence* assumption). Obviously, this is only valid for small temporal increments between consecutive images.

The neighborhood concept is, therefore, extended and as a result, spatial and temporal neighbors are incorporated during the weighting sum. A key difference between previous versions of the bilateral filter and the RBLT lies on its ability to integrate, in a robust way, the temporal evolution of the focal pixel in order to obtain a consistent brightness value for a sequence of consecutive images.

Thereby, the robust bilateral formulation is extended:

$$I_f(\mathbf{x}) = \frac{\sum_{i,j=-\infty}^{+\infty} \omega_d(\mathbf{x}_{ij},\mathbf{x}).r_{\sigma_r}(i,j).I_{src}(\mathbf{x}_{ij},t)}{2\sum_{i,j=-\infty}^{+\infty} \omega_d(\mathbf{x}_{ij},\mathbf{x}).r_{\sigma_r}(i,j)}$$

$$+\frac{\sum_{k=0}^{K} \omega_t(\mathbf{x}_k,\mathbf{x}).r_{\sigma_{rt}}(k).I_{src}(\mathbf{x},t-k)}{2\sum_{k=1}^{K} \omega_t(\mathbf{x}_k,\mathbf{x}).r_{\sigma_{rt}}(k)}, \quad \forall \mathbf{x} \in \Omega. \tag{4.15}$$

In addition to the spatial contribution of each neighbor, the formulation presented by Eq. 4.15 adds a temporal contribution. Temporal functions $\omega_t(.)$ and $r_{\sigma_{rt}}(.)$ are defined similarly to $\omega_d(.)$ and $r_{\sigma_r}(.)$, respectively. They intend to measure the photometric similarity and the associated temporal range for the focal pixel at different time delays, k. The temporal range function, $r_{\sigma_{rt}}(.)$, can be expressed using a Gaussian equation in order to give a higher weight to the focal pixel with a lower time delay. The idea is that focal pixel values closer in time are more likely to be temporally correlated. The temporal domain, $\omega_t(.)$, detects and treats violations of the *temporal coherence* assumption, preventing the smoothing across boundaries and making it possible to denoise images with low SNR, that is, a strong noise component.

Indeed, these two functions add a new contribution factor to Eq. 4.14, allowing one to measure both spatial and temporal consistency of the focal pixel.

4.3 Results

An extensive set of experiments was conducted as a part of this research. The experiments aimed at evaluating and analyzing the behavior of the proposed robust bilateral and temporal (RBLT) filtering technique comparative to the state-of-the-art techniques. Later,

the RBLT was compared to the original bilateral filter (BL) [122] and the spatiotemporal versions of Gaussian average filter and median filter for different testing conditions.

In these experiments, the original image (distortion-free or reference), $\hat{I}(\mathbf{x})$, is compared to the distorted image, $I(\mathbf{x})$, using several evaluation metrics. The distorted image is obtained by corrupting the original image with a distinct noise configuration and then, the image sample is filtered by each filter, individually. These experiments assess how the perceptual losses on image sequences have been corrupted with different combinations of Gaussian and Salt-Pepper noise. The assessment was performed using an objective (quantitative) and a subjective (qualitative) evaluation. Several objective metrics have been used in the literature, namely, the mean squared error (MSE), the root mean square error (RMSE), the signal-to-noise ratio (SNR) and the peak signal-to-noise ratio (PSNR) (the first two are represented in pixels and the last two are represented in decibels, dB) [121, 5, 148]. The previous metrics are simple to calculate and have a clear physical meaning; however, they are not as suitable to perceive visual quality. The perceptual quality assessment is accomplished using the structural similarity (SSIM) [148].

Finally, the performance of the RBLT filter is evaluated using surveillance image sequences obtained by a mobile robot. These videos represent highly dynamic scenes and the experiments demonstrate the behavior of the RBLT filter under real testing conditions.

The RBLT filter is implemented in C++ using the common OpenCV library[3]. The bilateral filter, used in this research is an implementation of the Tomasi and Manduchi [122] filtering technique and it is a standard function of the OpenCV. Videos of some experiments can be found in `http://paginas.fe.up.pt/~dee10015/_rblt.htm`.

4.3.1 Quality assessment

The quality assessment intends to analyze the visual degradation of the image which results from distortions made by the acquisition, transmission, processing, compression and storage of digital image sequences [148]. The assessment can be performed using an objective (quantitative) and subjective (qualitative) evaluation [149]. However, this research focuses on evaluating the filter using objective quality metrics.

Several metrics have been used in the literature and, for that reason, the most often used are briefly presented in this section. The simplest and most widely used quality metrics are the mean squared error (MSE) and the root mean square error (RMSE).

$$MSE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[I(\mathbf{x}_{ij}) - \hat{I}(\mathbf{x}_{ij})\right]^2. \tag{4.16}$$

[3] All experiments were conducted using the version 2.4.3 of the OpenCV.

$$RMSE = \sqrt{MSE}. \tag{4.17}$$

The signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) are represented in decibels (dB). The first measures the ratio of signal and noise power. The second measures the ratio between the maximum amplitude of a signal[4] and the power of corrupting noise.

$$SNR_{dB} = 10\log_{10}\left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{I}(\mathbf{x}_{ij})^2}{NM\ MSE}\right). \tag{4.18}$$

$$PSNR_{dB} = 10\log_{10}\left(\frac{(2^{nbits}-1)^2}{MSE}\right). \tag{4.19}$$

A higher PSNR value leads to a higher image quality and, therefore, as the PSNR value is closer to infinity the MSE gets closer zero. Furthermore, small PSNR values means that there are higher differences between the reference and the distorted image.

The previous metrics are simple to calculate and have a clear physical meaning; however, they are not as suitable to perceive visual quality. The perceptual quality assessment is accomplished using the structural similarity (SSIM) [148]. The metric calculates the similarity between two images by separating similarity evaluation into three comparisons: luminance, contrast and structure.

$$SSIM = \frac{(2\mu_r\mu_d + c_1)(2\sigma_{rd} + c_2)}{(\mu_r^2 + \mu_d^2 + c_1)(\sigma_r^2 + \sigma_d^2 + c_2)}, \tag{4.20}$$

where μ_r and μ_d are the luminance average of $\hat{I}(\mathbf{x})$ and $I(\mathbf{x})$, respectively. σ_r and σ_d are the standard deviation of the signal contrast and σ_{rd} is the covariance of both images that defines the correlation coefficient of the structure comparison. The c_1 and c_2 are constants that avoid the instability of the denominator. The positive values of the SSIM are within $[0,1]$. The value is 0 is there is no correlation between both images and 1 when both are identical.

4.3.2 Comparison to state-of-the-art techniques

The proposed spatiotemporal filter is tested with a variety of frequently used color video sequences[5] contaminated by Gaussian and Salt-and-Pepper noise, for instance, all chan-

[4] The $(2^{nbits}-1)$ is the digitalization maximum range that can be used to represent a signal, for instance, representing the image with 8bits means a range of 255.

[5] All videos are in a YCrCb 4:2:0 format and available online: `http://media.xiph.org/video/derf/`

Table 4.2: Gaussian noise - Average PSNR results for "Miss America" and "Salesman" sequences. Results of recent video denoising techniques are reported in [5] (the values in bold depict the best performance).

				MissA.	(PSNR)			
σ_G	RBLT	Wiener2D	Wiener3D	WRSTF	IFSM	VBM3D	2DGSM	STGSM
10	38.90	34.51	36.95	37.82	37.52	40.29	38.52	**40.58**
15	36.90	31.64	35.60	36.17	35.41	39.30	37.14	**39.40**
20	35.59	29.56	34.06	34.79	33.86	**38.54**	36.14	38.50
50	29.83	21.76	23.39	NA	29.79	33.39	30.49	**31.62**
100	**23.26**	12.84	13.27	NA	22.49	22.81	22.16	22.55
				Salesman	(PSNR)			
σ_G	RBLT	Wiener2D	Wiener3D	WRSTF	IFSM	VBM3D	2DGSM	STGSM
10	33.70	31.97	29.59	35.54	34.22	**38.33**	33.80	38.04
15	32.36	29.51	29.30	33.56	31.85	**36.60**	31.73	36.03
20	31.26	27.80	28.88	32.00	30.22	**35.12**	30.28	34.62
50	26.79	21.31	22.92	NA	25.40	**28.49**	24.95	26.87
100	**22.16**	13.09	13.50	NA	20.78	21.39	20.32	20.87

nels of the free-noise image (of the sequence) are corrupted. The colored version of the filtering technique is obtained by applying the grayscaled version of the method to each channel of the corrupted images, which is represented on the YCrCb space because this space is much more decorrelated with little dependency between different components [121].

Two objective metrics, namely, the PSNR and the SSIM, were used to provide quantitative quality evaluations of the denoising results since they are employed in most research [5]. Some results presented in this section are reported in [5] and [6]. They provide a reliable baseline that makes it possible to compare the proposed RBLT filter with state-of-the-art techniques using the same testing conditions.

4.3.2.1 Gaussian noise

In this experiment, the color videos are corrupted with Gaussian noise with different standard deviations. The "Miss America" and "Salesman" sequences of the QCIF resolution (176×144 pixels) were tested. Both sequences are characterized by a slow irregular motion. Tables 4.2 and 4.3 present the average PSNR and SSIM, respectively. State-of-the-art techniques for denoising videos degraded with Gaussian noise are the following: wavelet-domain reliability-based spatiotemporal filtering (WRSTF) [150], inter-frame statistical modeling (IFSM) [151], block matching and 3D filtering (VBM3D) [152], Gaussian scale mixture (2DGSM) [153] and the spatiotemporal Gaussian scale mixture (STGSM) [5].

Table 4.3: Gaussian noise - Average SSIM results for "Miss America" and "Salesman" sequences. Results of recent video denoising techniques are reported in [5].

				MissA.	(SSIM)			
σ_G	RBLT	Wiener2D	Wiener3D	WRSTF	IFSM	VBM3D	2DGSM	STGSM
10	**0.953**	0.818	0.908	0.908	0.904	0.947	0.936	0.952
15	0.933	0.704	0.868	0.877	0.857	0.939	0.922	**0.943**
20	0.915	0.602	0.808	0.846	0.812	0.933	0.913	**0.936**
50	0.782	0.214	0.286	NA	0.780	**0.905**	0.874	0.892
100	0.633	0.044	0.040	NA	0.604	0.789	0.809	**0.823**
				Salesman	(SSIM)			
σ_G	RBLT	Wiener2D	Wiener3D	WRSTF	IFSM	VBM3D	2DGSM	STGSM
10	**0.962**	0.859	0.839	0.932	0.904	0.960	0.909	0.960
15	0.932	0.778	0.818	0.901	0.851	**0.945**	0.865	0.941
20	0.910	0.704	0.785	0.868	0.801	**0.925**	0.825	0.923
50	**0.798**	0.340	0.425	NA	0.609	0.771	0.611	0.727
100	**0.626**	0.074	0.066	NA	0.488	0.538	0.464	0.496

As confirmed in table 4.3, the performance of the RBLT is on the top-3 of the filtering techniques, for sequences corrupted with low and moderate noise levels, $\sigma_G = \{10, 15, 20\}$. However, it reached the best performance for sequences corrupted with high noise level (table 4.2 and $\sigma_G = 100$). Thus, the denoising capability of RBLT filter is competitive relative to other state-of-the-art techniques, especially when the SSIM assessment metric is used to observe the perceptual quality and if the sequences are corrupted with strong noise.

4.3.2.2 Salt and Pepper noise

The video color sequences "Miss America" and "Flowers"(352×288) were corrupted with Salt-and-Pepper noise of different intensities and in an independent way for each channel. The "Flowers" sequence is characterized by a fast translational motion, different textures and high diversity in color distributions.

State-of-the-art techniques [6] for denoising videos degraded with Salt-and-Pepper noise are the following: 3D fuzzy directional (3DFD) [6], 3D median (3DM), 3D vector median (3DVM), vector directional K-nearest neighbor with vector median (VDKNNVM) [154], generalized vector directional (GVD) [155], adaptive vector directional α-trimmed median (AVTM) [156], α-trimmed mean (ATM) [157] and the K-means nearest neighbor (KMNN) [157].

Table 4.4 compares the performance of the RBLT filter to state-of-the-art techniques (that deal with the Salt-and-Pepper noise). Once again, the proposed technique is the

[6]The "3D" information was omitted in some techniques to simplify the notation.

Table 4.4: Salt-Pepper noise - Average PSNR results for sequences "Miss America" and "Flowers". Results of recent video denoising techniques are reported in [6] (the values in bold depict the best performance).

					MissA.	(PSNR)			
%	RBLT	3DFD	3DM	3DVM	VDKNNVM	GVD	AVTM	ATM	KMNN
0	41.31	**48.54**	35.32	35.03	34.21	33.49	37.74	35.43	40.41
5	38.04	**39.59**	35.12	34.86	33.48	33.76	36.97	35.22	37.21
10	36.02	**36.61**	34.80	34.58	32.91	33.79	36.18	34.88	33.34
15	34.68	34.33	34.36	34.18	32.37	33.70	**35.38**	34.42	30.09
20	33.56	32.23	33.72	33.58	31.60	33.31	**34.47**	33.79	27.38
30	31.51	28.26	31.94	31.81	28.48	31.61	**32.33**	32.03	23.16
40	**29.69**	24.52	29.15	28.84	23.90	27.70	29.14	29.03	20.01
					Flowers	(PSNR)			
%	RBLT	3DFD	3DM	3DVM	VDKNNVM	GVD	AVTM	ATM	KMNN
0	29.80	30.53	27.04	26.96	26.15	25.61	27.53	27.06	**33.13**
5	28.77	29.52	26.83	26.78	25.77	25.56	27.25	26.85	**31.30**
10	28.12	28.61	26.54	26.52	25.45	25.46	26.93	26.58	**28.95**
15	27.31	**27.76**	26.22	26.20	25.11	25.29	26.57	26.27	26.63
20	26.49	**26.93**	25.83	25.80	24.57	24.84	26.14	25.87	24.49
30	24.95	**25.04**	24.77	24.69	23.17	23.24	24.99	24.79	20.86

best spatiotemporal filter for denoising the sequence "Miss America" corrupted with 40% of noise, the difference between the RBLT and the 3DM is about $0.54dB$. For other noise levels, the RBLT is usually on the top-3. Focusing on the sequence "Flowers", the RBLT is also on the top-3. However, the PSNR difference between the RBLT with the 3DFD is about $0.09dB$ for the trial with 30% of noise, instead of $0.52dB$ for the same noise condition on the sequence "Miss America". This means that the performance of the proposed filter is substantially better for higher noise levels and for dynamic video sequences. Thus, it is unfortunate that there are no results in the literature for higher levels of Salt-and-Pepper noise.

The major advantage of the proposed filter is that it deals with Gaussian and Salt-and-Pepper noise in a very competitive manner when compared to other state-of-the-art techniques (especially designed for each type of noise). In addition, the RBLT filter can be implemented following the latest research on the bilateral filter and, therefore, it can be computationally efficient.

4.3.3 Denoising surveillance sequences with reference

Two surveillance videos obtained by a stationary and high definition (HD) camera with a 4mm focal lens are corrupted with Gaussian and Salt-Pepper noise. The aim of these experiments is to evaluate and compare the responses of the proposed RBLT filter with

(a) (b)

Figure 4.2: Distortion-free images: 4.2(a) represents the frame at 13 seconds of the I1 and 4.2(b) is frame at 7.5 seconds of the I2 sequence.

several baseline filters: the spatial bilateral filter and the spatiotemporal Gaussian average and median filter.

The color videos used in these experiments are sequences of images with size 640 $\times$ 480 and they are obtained by defining a region of interest in the original HD videos.[7].

1. Sequence "I1" - Two persons are moving in the hall. The first person is entering from the left and followed by a second person that enters from the right. This testing sequence starts in the 11 seconds of the original video and finishes at the 17.5 seconds.

2. Sequence "I2" - A blue truck moving between two parking lots. This sequence finishes at the 21 seconds of the original video.

The filters' parameters were enhanced for the $\sigma_G = 25$ and 20% (Gaussian and Salt-Pepper experiment, respectively).

4.3.3.1 Case: Gaussian noise

Figures 4.4(d) and 4.5(d) represent two test scenarios under different noise conditions. They aim is to provide a visual idea of the performance of the RBLT filter, that is, qualitative judgments.

Figure 4.2(a) and 4.3(a) shows the distortion-free image and the image corrupted with Gaussian noise ($\sigma_G = 50$) for the sequence I1. In this trial, it is possible to visualize the noise reduction of the BL, Gaussian average, median and RBLT filters, Figs. 4.4(a),

[7]The original surveillance videos are courtesy of CCTV Camera Pros. The 1920 $\times$ 1080 videos are available in `http://www.cctvcamerapros.com/HD-CCTV-DVR-s/621.htm`.

(a) (b)

Figure 4.3: Noisy images: The I1 and I2 sequences corrupted by a Gaussian noise with a standard deviation of $\sigma_G = 50$ and $\sigma_G = 40$, respectively.

4.4(b), 4.4(c) and 4.4(d). The static part of the scene is clearly represented in all the images, especially, in the distorted average image. However, the dynamic part of the scene (the person) in the distorted spatiotemporal images suffers from some issues related with their temporal contribution. As expected, the average and the median filter introduce artifacts, usually, called as ghosts, into the denoised image that corrupt the moving objects (this is more evident in Figs. 4.5(b) and 4.5(c) of the sequence I2). These ghosts are problematic for high level image processing techniques, for instance, motion segmentation or tracking. They reduce the applicability of such type of filter. Figure 4.4(d) shows that the dynamic part is more clearly represented on the RBLT-filtered image as the distorted average and median image since the "robustification" proposed by the RBLT filter makes possible to detect the outliers that characterize the noise and avoids the incorporation of ghosts in sequences. In addition, the RBLT filter has a higher denoising capability when comparing to the BL filter, for instance, Figs. 4.4(a) and 4.4(d).

Other experiment which has a Gaussian noise component of $\sigma_G = 40$ that corrupts the I2 sequence is shown in Figs. 4.2(b) and 4.3(b). The result is similar to the previous experiment since it depicts a pronounced noise in the distorted BL image, Fig. 4.5(a). The BL filter is known by its ability to denoise a corrupted image; however, the distorted BL image shows a noise component that is higher than the spatiotemporal filters. As will be seen later in this section, the filtering performance that is achieved by the BL filter is closer to the RBLT filter when the noise component is approximated by a Gaussian distribution with a relatively small σ_G. Otherwise, the behavior of the BL filter is significantly compromised and the RBLT filtering will perform much better. Visually, the RBLT maintains the fundamental structure of the image sequences (Fig. 4.5(d)). It makes possible a substantial noise reduction while the structure of moving objects of the noise-free sequence is represented without being substantially smoothed. For instance, it is possible to

(a) (b)

(c) (d)

Figure 4.4: Filtering results for the I1 sequence: BL - 4.4(a), Gaussian - 4.4(b), median - 4.4(c) and RBLT filtering - 4.4(d). It represents the frame at 13 seconds of the I1 sequence which is corrupted by a Gaussian noise with a standard deviation $\sigma_G = 50$.

(a) (b)

(c) (d)

Figure 4.5: Filtering results for the I2 sequence: BL - 4.5(a), Gaussian - 4.5(b), median - 4.5(c) and RBLT filtering - 4.5(d). It represents the frame at 7.5 seconds of the I2 sequence which is corrupted by a Gaussian noise with a standard deviation $\sigma_G = 40$.

identify the truck's edges in the RBLT-filtered image while it is difficult to extract these borders in the distorted average and median images, see Figs. 4.4(b) and 4.5(c). Thus, the RBLT-filtered image displays a better performance when compared to other filters.

Subsequent to the qualitative judgment, the BL, average, median and RBLT filters are evaluated using the assessment metrics, presented in section 4.3.1. At first, the temporal evolution of the filters' performance is analyzed using the I1 sequence (the results were similar for the I2). Figures. 4.6(a), 4.6(b), 4.6(c) and 4.6(d) show graphical representations of the filtering accuracy for several metrics. After the fourth frame, the results of the RBLT filter (red line) are significantly better comparative to other filters. This result was already expected since the RBLT uses robust temporal information of previous frames to denoise the current image frame. Therefore, it is possible to infer that the RBLT filter has a starting delay time which is the duration of time needed by the filter to obtain reliable temporal information. This starting effect can be seen in Fig. 4.6(a) to 4.6(d) and its duration is only a couple of consecutive frames. The RBLT filter has the best performance during the time evolution of the I1 sequence which is followed by the Gaussian average, the median and the BL filter. On average, the RMSE difference between the RBLT and average filter is usually more than 4 pixels, Fig. 4.6(a). According to the SNR and PSNR graphics, Figs. 4.6(b) and 4.6(c), the RBLT filter is normally 3 dB more accurate than the average filter. The performance of the filters are well depicted by the SSIM metric and presented in Fig. 4.6(d). As confirmed, the difference between the RBLT and the best baseline filtering technique is almost 0.12 and the SSIM index of the RBLT result is higher than 0.45 for the entire experiment that corrupts the I1 sequence with Gaussian noise ($\sigma_G = 50$).

A final experiment considering the Gaussian noise is conducted in order to analyze the robustness of each filter under different noise conditions, for instance, the filtering accuracy is evaluated for several standard deviations. The RBLT (red circle line), BL (green dashed line), Gaussian average (yellow line) and median (cyan dot line) filtering behaviors are compared for the I1 sequence. These results were obtained for the time instants portrayed in Fig. 4.2(a), but for various conditions. Figures 4.7(a), 4.7(b), 4.7(c) and 4.7(d) are graphical representations of the performance achieved by each filter using RMSE, SNR, PSNR and SSIM assessment metrics, respectively.

Obviously, increasing the standard deviation leads to a worse performance for all the filters. Although, the distorted RBLT images have higher quality relatively to the BL images (considering SSIM metric), especially, when standard deviation values are higher. For the $\sigma_G = 30$, the RMSE difference of the RBLT is about 14 point below the BL for the I1 sequence, see Fig. 4.7(a), and 7 point below the average filter. The PSNR profiles are similar to the SNR without the an offset, Figs. 4.7(b) and 4.7(c). For $\sigma_G = 30$, the SNR

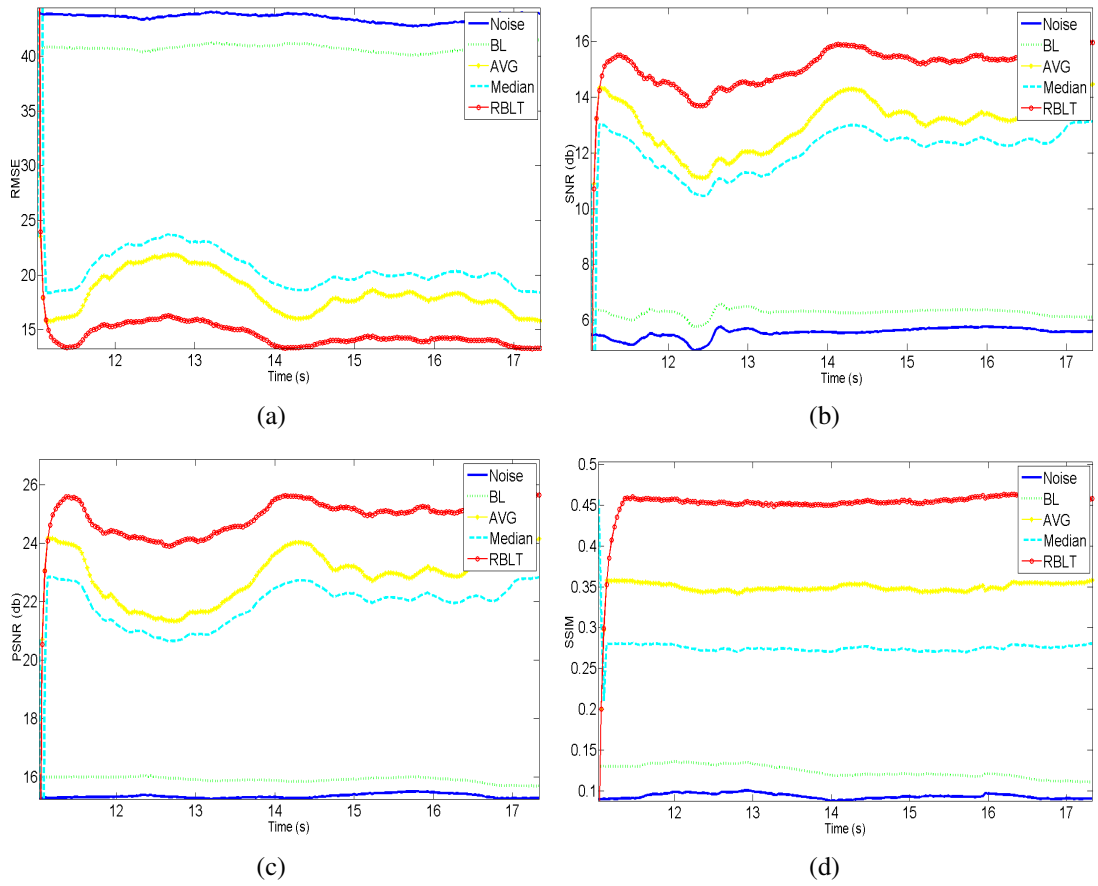


Figure 4.6: Performance evolution over time (in seconds) of the noise reference (dark blue line), bilateral (green dashed line), Gaussian average (yellow line), median (cyan dot line) and RBLT (red circle line) filter. Each filtering process was conducted for the I1 sequence and under a Gaussian noise with $\sigma_G = 50$. Figures 4.6(a), 4.6(b), 4.6(c) and 4.6(d) are graphical representations of the performance evolution using RMSE, SNR, PSNR and SSIM assessment metrics.

and the PSNR difference between the RBLT and the average filter is $4dB$ in both metrics. For the same noise condition, the SSIM index of the RBLT is about 0.69 whereas the BL is 0.22, the median is 0.48 and the average is 0.55. Thus, the SSIM difference between the RBLT and the best baseline filter is 0.14, approximately. SSIM is the most suitable metric to observe the perceptual quality of the image and, therefore, the SSIM evaluation of the RBLT performance indicates that it performs better in all the experiments, Fig. 4.7(d).

In these experiments, the parameters were enhanced for $\sigma_G = 25$ to make possible a reliable analysis of the behavior of each filter under several noise conditions. Consequently, the distorted BL images present an accuracy that is slightly better than the RBLT images for a standard deviation lower than 15 and considering RMSE, SNR and PSNR metrics (Figs. 4.7(a) to 4.7(c)). Actually, the configurations of the spatiotemporal filters that were used in these experiments are not suitable for images withing small noise since they do not need a strong temporal contribution (their performance is slightly lower than the reference). According to the RMSE, SNR and PSNR metrics, the BL performs better for smaller σ_G, although the SSIM indicate that the RBLT presents a higher accuracy independently of the σ_G. As it can be noticed, the difference is not significant and confirms what already was said, namely, that the BL presents an acceptable performance in situations where the image has a noise that can be reliably characterized by a Gaussian distribution with smaller standard deviations.

The RBLT filter incorporates the advantages of the BL formulation and yet it can filter images with a noise component that is much more intense. An ordinary filter that incorporates temporal information, usually, smooths the edges (regions with large gradient information); however, the proposed RBLT technique correctly detects the presence of boundaries, for instance, the red lights of the truck are maintained from the distortion-free image to the distorted result. For this extensive set of experiments based on Gaussian noise, it is possible to conclude that the RBLT filter performs similarly to the BL for smaller standard deviations; however, it performs much better as σ_G increases.

4.3.3.2 Case: Salt and Pepper noise

The RBLT filter is evaluated under the Salt-Pepper noise, Figs. 4.8(a) and 4.8(b), and using a test methodology similar to the Gaussian case, described in the previous section.

A qualitative judgment for a 50 percentage of noise can be visualized in Figs. 4.9(a), 4.9(b), 4.9(c) and 4.9(d). It is possible to verify in these images that the BL is not capable of filtering the corrupted images since the BL distorted image still has a strong noise component. Unlike the Gaussian case, the BL distorted images are still very corrupted. The BL filter presents some issues when it comes to filtering images with anisotropic

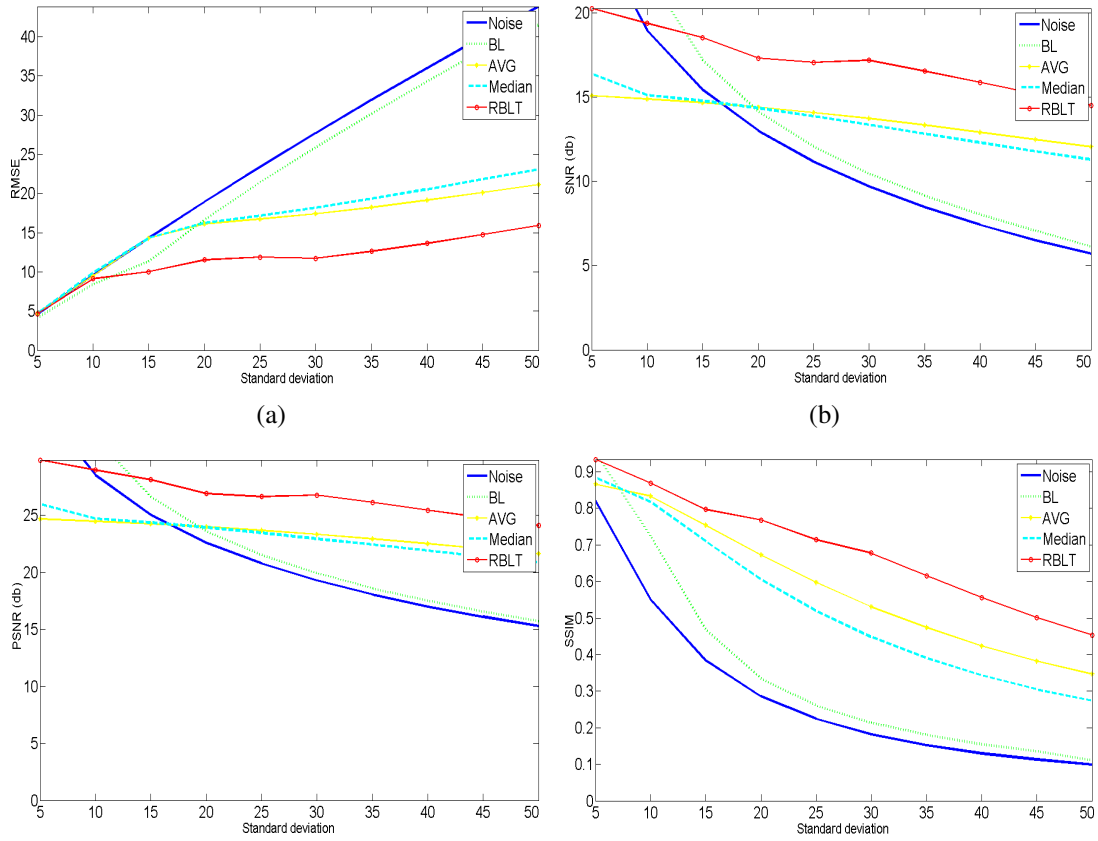
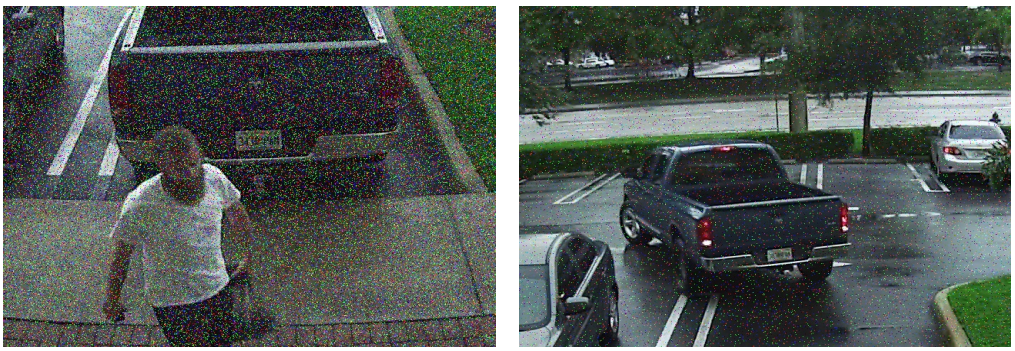


Figure 4.7: Performance evolution as a function of the standard deviation. The noise reference (dark blue line), bilateral (green dashed line), Gaussian average (yellow line), median (cyan dot line) and RBLT (red circle line) filtering were conducted on the I1 sequence and under a Gaussian noise. Figures 4.7(a), 4.7(b), 4.7(c) and 4.7(d) are graphical representations of the performance evolution using RMSE, SNR, PSNR and SSIM assessment metrics. These graphs represent the filtering results for the I1 sequence instant of Fig. 4.2(a).



(a) (b)

Figure 4.8: Noisy images: The I1 and I2 sequences corrupted by a Salt-Pepper noise with a percentage of 50% and 10%, respectively.

(a) (b)

(c) (d)

Figure 4.9: Filtering results for the I1 sequence: BL - 4.9(a), Gaussian - 4.9(b), median - 4.9(c) and RBLT filtering - 4.9(d). It represents the frame at 7.5 seconds of the I1 sequence which is corrupted by a Salt-Pepper noise with a percentage of 50%.

noise (in a density field context), simulated by the Salt-Pepper noise. On the contrary, the performance of the RBLT filter is substantially better since a large noise component is removed from the sequence while the moving structures remain intact. Therefore, the RBLT distorted image is less degraded and the resulting image has less background noise because the temporal information is by itself enough to remove the noise from static and dynamic regions. The parameters of the spatiotemporal median filter were enhanced in order to match their performance with the RBLT filter. As it can be noticed in Fig. 4.9(c), the background scene is very clear because the temporal contribution is strong. However, the dynamic structure of the scene is distorted with ghosts, for instance, the moving person of the distortion-free image is transformed in two moving persons. For this reason, the behavior of the median filter does not satisfy our objectives since the dynamic objects of the scene will be unreliable if the scene is captured by a camera mounted in a mobile robot.

The assessment results of the I2 sequence filtering are presented in Figs. 4.10(a),

4.10(b), 4.10(c) and 4.10(d). Analyzing these graphs, it is possible to infer that the performance of the RBLT is overwhelmingly better than the baseline filters.

After their short starting time, the RBLT filtering reaches a RMSE that is 35 points below the BL filter, 9 points below the average filter and 4 points below the median filter. In addition, the SNR and the PSNR indexes of the proposed filtering technique are $12dB$ and $22dB$ higher, respectively, and the SSIM is 0.55 higher over the entire sequence. Also, the SSIM difference between the RBLT and the BL filter is about 0.37, Fig. 4.10(d). For some instants and considering this last evaluation metric, the performance of the median filter is similar to the RBLT filter because the video was obtained from a stationary surveillance system and, therefore, the majority of the scene is static. If the video is obtained from a moving camera it is expected that the performance of the RBLT filter will remains approximately the same and the median filter will be substantially worse. From a qualitative evaluation point of view, the performance of the median filter is poor because it introduces ghosts. In contrast to the performance of the conventional bilateral and Gaussian average filter, the RBLT presents a remarkable signal reconstruction capability for images corrupted with Salt-Pepper noise.

Increasing the percentage of pixels that are degraded, the filtering process will be more complex and difficult. Therefore, the performance evolution of the filters as the percentage change from 5% to 50% is studied.

The assessment metrics prove the denoising complexity when sequences are affected by a strong noise component since the performance of filters decreases as the number of degraded pixels increases, see Figs. 4.11(a), 4.11(b), 4.11(c) and 4.11(d). The BL and the spatiotemporal filter have distinct performances. For 25% of noise, the RBLT filtering leads to a RMSE of 15 points, which is 6 points lower than median filter, 11 points lower than the average filter and 25 points lower than the BL filter, see Fig. 4.11(a). This represents a large difference between the performances, which is also depicted by other graphics. Figures 4.11(b) and 4.11(c) show that this difference is $3.5dB$ for the SNR and $3dB$ for the PSNR (relative to the median filter). In the same noise condition, the SSIM evaluation of the RBLT performance indicates that it performs similar to the median filter and better for the other baseline filters, Fig. 4.11(d). When 50% of the pixels are corrupted, the SSIM index of the RBLT is 0.23 higher than the Gaussian average filter and 0.4 higher than the BL.

4.3.4 Denoising surveillance sequences without reference

A final set of experiments was conducted in order to demonstrate the ability of the RBLT to filtering highly dynamic sequences. The scene is captured by a camera mounted in a

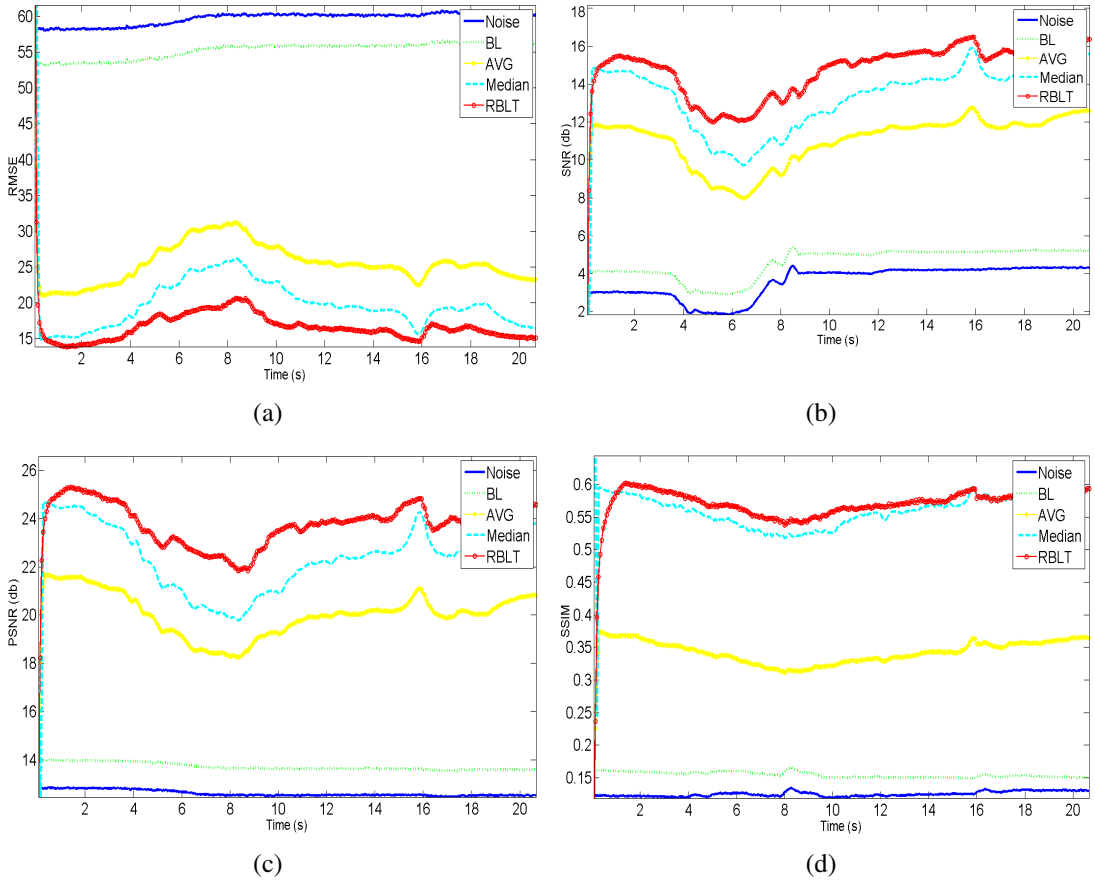


Figure 4.10: Performance evolution over time (in seconds) of the noise reference (dark blue line), bilateral (green dashed line), Gaussian average (yellow line), median (cyan dot line) and RBLT (red circle line) filter. Each filtering was conducted on the I2 sequence and under a Salt-Pepper noise with 50 % degradation. Figures 4.10(a), 4.10(b), 4.10(c) and 4.10(d) are graphical representations of the performance evolution using RMSE, SNR, PSNR and SSIM assessment metrics.

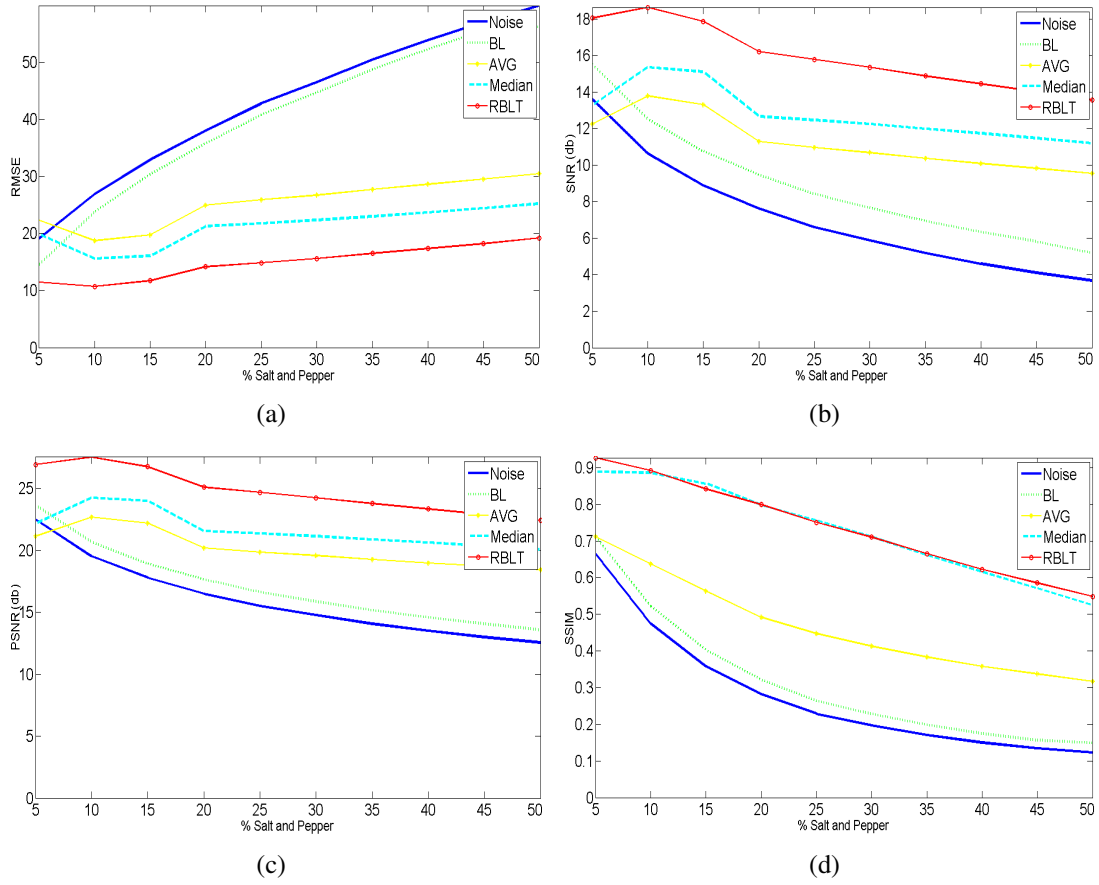
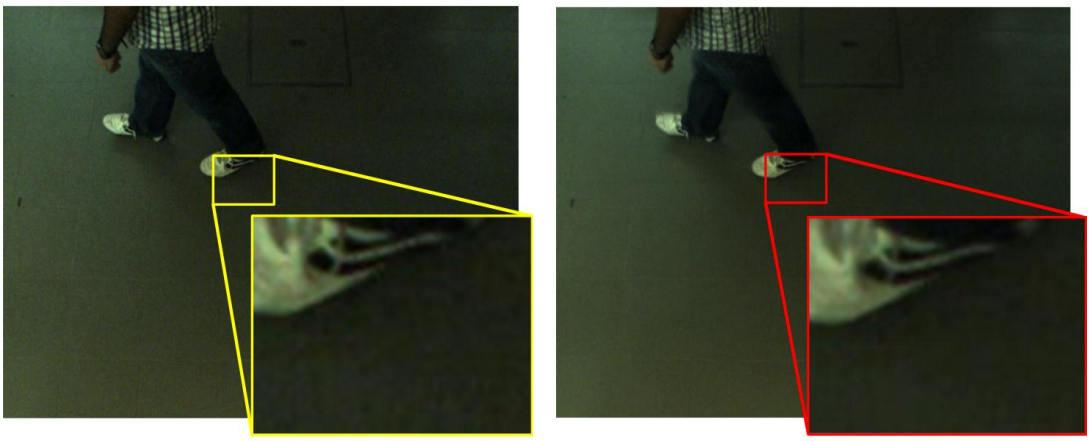


Figure 4.11: Performance evolution as a function of the percentage of pixels that are corrupted. The noise reference (dark blue line), bilateral (green dashed line), Gaussian average (yellow line), median (cyan dot line) and RBLT (red circle line) filtering were conducted on the I2 sequence and under a Salt-Pepper noise. Figures 4.11(a), 4.11(b), 4.11(c) and 4.11(d) are graphical representations of the performance evolution using RMSE, SNR, PSNR and SSIM assessment metrics. These graphs represent the filtering results for the I2 sequence instant of Fig. 4.2(b).



(a) (b)

Figure 4.12: Surveillance sequence of a moving person and obtained by a mobile robot. 4.12(a) depicts the image captured by a "TheImagingSource DFK 21AU04" camera with a 4mm focal lens. 4.12(b) represents the RBLT-filtered image.

mobile robot. Therefore, the resulting videos incorporate the egomotion and the movement of objects. The previous sections have proved the inability of the Gaussian average filter and the median filter to suppress the noise in dynamic videos. Thus, the RBLT filter is applied to these videos and the results are presented in Figs. 4.12(b) and 4.13.

The motion segmentation with the observer in motion is a complex problem and segmentation techniques have great difficulty in dealing with the existence of additional noise, especially, the optical flow methods. The original videos have an evident noise component, see Fig. 4.12(a) and the bottom side of Fig. 4.13. the RBLT-filtered image (top side of the Fig. 4.13) is substantially better since a large noise component is removed from the sequence while the moving structures remain intact, for instance, the borders and the edges. It is possible to see in Fig. 4.12(b) that the sneaker of the person is correctly represented and the background pixels are less corrupted by noise. Therefore, the RBLT filter denoise the corrupted image and it does not create ghosts or other artifacts that may compromise the high level processing techniques for instance, the motion segmentation.

In conclusion, the RBLT filtering technique has an interesting ability to reconstruct video sequences, even when sequences are corrupted with a strong noise component. This ability is a consequence of the significant improvements proposed by this chapter. The RBLT filter extends the BL formulation by introducing robust error norms to detect the presence of outliers and uses the temporal evolution of the pixel to infer about its current value.

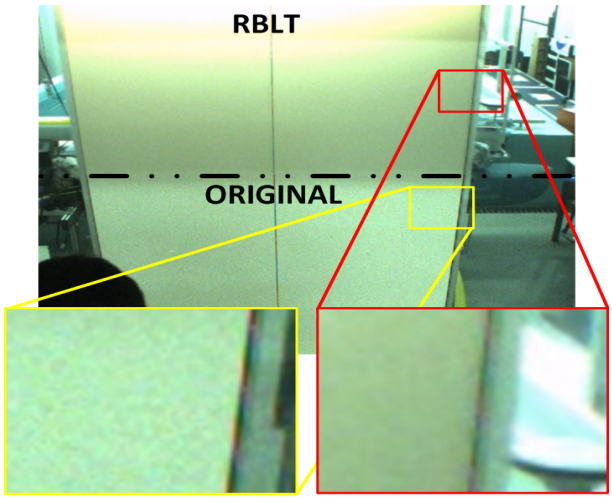


Figure 4.13: Image for comparing the original image and the result obtained by the RBLT filter. The 640 $\times$ 480 image is split in two, the bottom side is the original image and the top half is the RBLT-filtered image.

It was proved in Section 4.3.2 that these extensions lead to a filtering process that is robust, reliable and independent of the type and intensity of noise that corrupts the sequence. The experiments conducted in section 4.3 prove that it is possible to reconstruct the degraded image sequences while maintaining important structures (textures and gradients), that is, without a significant perceptual image loss. The denoising capability of the RBLT technique was similar for the standard experiments of section 4.3.2 and the realistic experiments of sections 4.3.3 and 4.3.4. For instance, PSNR indexes of the RBLT for sequences corrupted with Salt-Pepper noise (10% and 30%), were about $28dB$ and $24dB$ in table 4.4 (”Flowers”) and, $27dB$ and $24dB$ in Fig. 4.11(c). For sequences corrupted with Gaussian noise and $\sigma_G = 50$, PSNR indexes of the RBLT were $26dB$ and $24dB$, for the sequence ”Salesman” (table 4.2) and the Fig. 4.7(c), respectively.

4.4 Final considerations

This chapter has proposed a novel filtering technique which is called Robust BilateraL and Temporal (RBLT). The RBLT extends the formulation of the bilateral filter (BL) in order to enhance its ability to reconstruct degraded image sequences. Unlike other bilateral versions, the proposed technique uses temporal information about the image sequence to guide the filtering process.

The RBLT formulation combines two major advantages that reduce the noise component and make the image sequences clearer, namely, by incorporating a temporal context and by making the spatial and temporal neighbors more robust. It assumes a temporal

correlation for the pixel brightness over time (*temporal coherence* assumption) which is valid if temporal increments between consecutive images are small. In addition, robust error norms are used to decrease the influence of outliers during the estimation of the pixel value, for instance, when the process is less affected by outlying neighbors. Therefore, the denoising capability of the RBLT comes from their ability to estimate the pixel value as a robust combination between the spatial and temporal characteristics of pixel's neighborhood. This also makes the filter more independent of the noise's characteristics.

The proposed filter was compared to state-of-the-art methods and the performance of the RBLT was also evaluated in realistic scenarios. The experiments conducted prove that it is possible to reduce the noise component even in images having low SNR, that is, extremely degraded image sequences. In short, the chapter presents a reliable and robust filtering method that is highly recommended for surveillance sequences and for vision-based applications that resort to moving cameras, for instance, mobile robotic systems.

Chapter 5

The HybridTree Optical Flow Technique

In recent years, several optical flow techniques have been proposed. Even thought they are relatively accurate, most of them are too computationally demanding for autonomous robots and mobile systems, due to the limited computer resources that characterize most of these applications.

This thesis proposes an innovative and efficient dense optical flow architecture. The designed technique captures and combines the advantages of the local and global differential optical flow methods with a hierarchical and tree-based structure, achieving a surprising balance between computational effort and flow performance. The proposed *HybridTree* method is able to identify the intrinsic nature of the motion by performing two consecutive operations: *expectation* and *sensing*. A quadtree-based scheme and descriptive properties of the image are retrieved during the *expectation* phase. This makes it possible to divide the image into regions that may have different motions. In the sensing operation, the properties of regions are used by a hybrid and hierarchical optical flow structure to estimate the flow field.

The chapter[1] is organized as follows. Section 5.1 presents a brief introduction. Section 5.2 presents the *HybridTree* optical flow. An overview of the full structure of the proposed technique is provided in section 5.2.1 and afterwards, both operation phases are described in detail in sections 5.2.2 and 5.2.3. Section 5.3 presents the experimental results: section 5.3.1 shows the *expectation* performance and characterizes the behavior of both splitting and merging operation; and section 5.3.2 presents the results of the *sensing* phase, that is, modern implementations of the Lucas-Kanade (LK) [69], Horn-Schunck (HS) [68] and Combined Local-Global (CLG) [21] are compared with the *HybridTree*.

[1] Some portions of this chapter appeared in [4, 23].

The section 5.4 presents a set of dense flow fields that were obtained using the proposed *HybridTree* method in a realistic surveillance scenario. Finally, section 5.5 presents the most important conclusions of this chapter.

5.1 Introduction

Mobile robotic applications have certain problems related to visual perception and interpretation of the dynamic scene: unmanned aerial vehicles, unmanned surface vehicles and unmanned ground vehicles. In these applications, suitable motion detection is crucial to feed the high level processes with relevant information in order to define external interactions, navigation procedures and to increase the autonomy of the robots. Optical flow techniques provide excellent information about the motion vector of each pixel; however, they are still computationally too expensive for a real-time application without specialized hardware [158].

In the past few years, a high number of optical flow methods have been proposed however, the large majority focuses on the accuracy only. These methods use more elaborate and robust techniques (multiscale, iterative, nonquadratic and non-convex error functionals). In fact, state-of-the-art methods make possible to obtain very accurate estimations, but this accuracy is achieved at the expense of an exacerbated complexity as processing times increase immensely even using specialized hardware, for instance, GPUs (Graphics Processing Units). Optical flow techniques were not widely used before this last decade in real-time robotic applications due to the lack of robustness and that still happens today because they are computationally expensive.

The work presented by this chapter focuses on motion detection and extraction using a dense optical flow technique, designed for a robotic application with a vision system and limited computer resources. The proposed technique mimics the human motion detection based on different layers of visual details. The technique uses an innovative sensing structure composed of distinct perceptive levels defined by a hierarchical tree-based configuration, as will be discussed later. No other optical flow technique has been found in the literature that uses similar high level information on the image to perform a guided optical flow estimation.

The contributions of this chapter include:

1. An assisted optical flow estimator, that combines local and global differential methods using cognitive information;

2. A hierarchical method to guide the flow estimation process, enabling an optical flow enhancement while preserving the computational time requirements;

3. An efficient method to decompose the image into exclusive regions based on similarity properties: temporal differencing, texture, brightness and color;

4. Promoting efficient optical flow techniques for robotic and surveillance applications;

5. An extensive qualitative and quantitative evaluation.

Two operational steps, *expectation* and *sensing*, form the *HybridTree* method. The *expectation* decomposes the current image into regions with distinctive properties. A tree-based data structure, named quadtree, analyzes the image using descriptive properties to infer about the partitioning. Properties such as texture gradients [1], dominant color and spatial information reflect the similarity characteristics between different regions. A hierarchical and hybrid optical flow technique is used by the *sensing* operation combining modern implementations of the local and global differential optical flow methods. This hybrid approach blends the advantages of those techniques in a symbiotic and hierarchical topology.

The most classical optical flow formulations, such as, [69] and [68], can achieve respectable optical flow estimations when implemented with appropriate modern practices [23]. These practices play an important role to the accuracy of state-of-the-art methods; however, some of them are computational demanding, for instance, the warping procedure, non-quadratic and non-convex penalty functions (see [23]). They usually require special programming techniques and hardware to estimate the flow field in less than a couple of seconds, namely, multi-threading architectures and GPU. This strong computational effort means that the approaches are not very appropriate for the current robotics system.

Even thought they are not the most accurate flow formulations to date, classical optical flow methods are sometimes used in surveillance [159, 84] and robotic applications [160, 161], as they are more computationally efficient. However, their flow estimation is not very reliable. Some modern practices that improve the performance of these differential optical flow methods will be presented in this research.

The combination of local and global approaches benefits from the exclusive advantages of each method, for instance, local methods are robust to noise [21] unlike global methods. Furthermore, global methods make it possible to perform a dense estimation of the optical flow and the smoothness term propagates the flow in textureless regions (areas without significant gradient information), making them suitable for some kinds of images, such as, images containing many homogeneous areas. A related work includes [21], which use energy functionals to formulate the Combined Local-Global (CLG) method.

The 2D energy term combines the local robustness to noise offered by Lucas-Kanade with the regularized smoothness of the global Horn-Schunck. Although, this method's performance is good, the minimization of the energy functional requires a long time for the computation process, even when the successive over relaxation method (the method with fast convergence rate) is used to solve the Euler-Lagrange equations.

The developed architecture has a natural predisposition to detect and identify motion regions since the cognitive information (object boundary, texture and temporal evolution) assigns the most suitable optical flow technique and parameter configuration that fits each hierarchical level and processing region (parameter enhancement). By taking into account the expected motion properties of each region, the high level information makes it possible to identify regions where the computation effort should be increased. Therefore, the novel optical flow architecture proposed in this thesis differs from most methods because firstly it decomposes the image and then retrieves cognitive features on the spatial distribution of the expected motion in order to feed the multi-resolution hybrid optical flow procedure. This makes it suitable for different real-time applications.

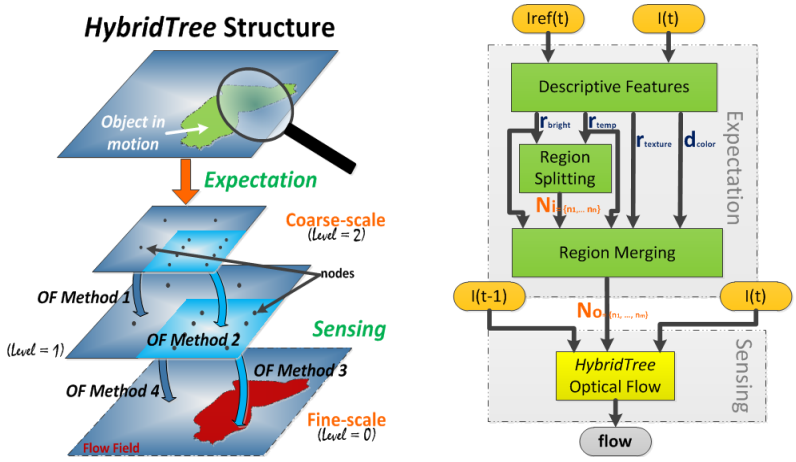
Experimental results include the comparisons of the proposed technique with modern versions of hierarchical classical Lucas-Kanade (LK) [69] and Horn-Schunck (HS) [68] optical flow techniques, as well as, the formulation presented by [21] which is called 2D-CLG (combined local-global). Experimental considerations prove that modeling the image into a hierarchical tree-based structure of cognitive information is computational rewarding and represents an alternative to state-of-art techniques based on "brute force" optimization.

5.2 HybridTree optical flow

5.2.1 Introduction

The proposed technique has two major and distinct phases: *expectation* and *sensing* (see Fig. 5.1(b)).

Briefly, the *expectation* phase evaluates the current image and identifies regions that potentially retain different characteristics and motions (like, homogeneous regions). The main objective is to guide the *sensing* phase during the optical flow estimation process. Different optical flow methods can be combined and allocated to each image region, avoiding the "brute force" computation. For instance, the *expectancy* controls the processing time in regions that require more effort to achieve the desired performance. In robotic applications, the proposed method can only focus on regions that apparently have the most complex and relevant motions. Figure 5.1(a) presents the major concept behind



(a) The concept. (b) Overview diagram.

Figure 5.1: 5.1(a) depicts the basic concept presented in this chapter. Several generic optical flow methods can be combined in a multi-scale approach in order to exploit the advantages of each approach according to the expected type of motion. This concept makes it possible to enhance the parameters. 5.1(b) depicts the overall structure and the relations between different stages of the method. The expectancy stage infers about the motion boundaries (region splitting and merging techniques based on texture, color and brightness properties). The sensing estimates the optical flow using a hierarchical and hybrid structure that incorporates the information on the boundaries.

the technique. The approach aims to balance the computational efficiency and the performance of the flow field, obtained at the finest level. The technique is highly configurable to meet the requirements imposed by generic applications, see Fig. 5.1(a). For instance, the technique can focus on relevant motion zones only, which decreases the time spent to compute the motion field, resulting in a significant advantage for robotic systems.

A hybrid optical flow technique is used in the *sensing* phase. This hybrid approach blends in a symbiotic and hierarchical scheme the advantages of both local and global techniques, LK and HS, to benefit from the exclusive advantages of each method. Figure 5.1(b) presents the overall structure of the proposed method, where $I(t)$[2] is the current two-dimensional colored image (time is denoted by the variable, t). $I_{ref}(t)$ is the reference image obtained from a temporal median filter based on a sliding window and is used to extract temporal information on the expected motion. For image sequences with only two frames, the reference image is $I(t-1)$.

[2]The image is represented in this thesis by $I(\mathbf{x},t)$, where $\mathbf{x} = (x,y)$ is the pixel spatial position. Whenever the context is clear the notation is simplified.

5.2.2 Expectation

An efficient splitting-merging technique is proposed for image partitioning. The process of dividing an image into regions must be consistent with spatial and temporal characteristics and it also should be aware of the computational effort spent during the entire process.

The *expectation* phase intends to:

- Improve the performance of optical flow estimations. The most suitable optical flow technique is used in each region according to its properties, enhancing the overall performance;

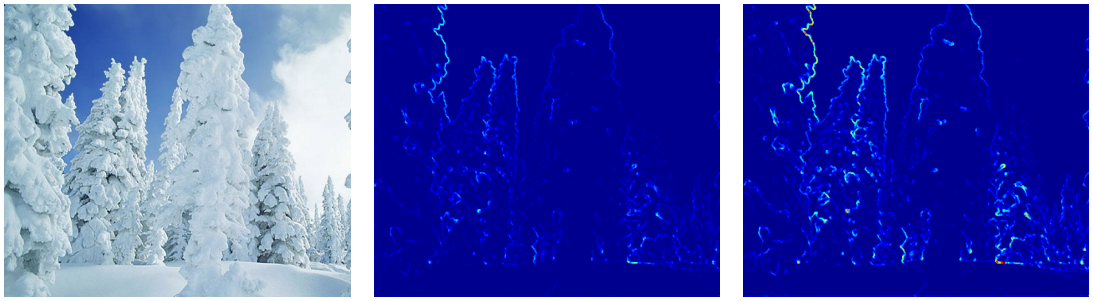
- Boost the computational efficiency by avoiding costly and intensive computations in large homogeneous regions. The aim is to enable real-time processing for limited computing systems.

Conventional methods for splitting and merging images deal with two major problems: when and how to divide a region? Although, common techniques intercalates merge and split operations [162] and these approaches are more time consuming. Both split-merge operations are applied separately and so it is desirable to have resulting regions with a regular shape for computational efficiency.

Differential optical flow methods resort to spatial and temporal derivatives to infer about the apparent vector motion of each pixel. Therefore, image decomposition should depict the spatial and temporal arrangement of the motion along a sequence of images. Descriptive properties such as brightness, texture gradients, colors and spatial information are used in this research to derive conclusions on the similarity between different regions. Hence, the splitting method proposed in this research uses the brightness magnitude gradient and the absolute temporal derivative to divide the image into many regions. Region merging uses the spatial information, dominant color, absolute temporal derivative and magnitude gradient for both brightness and texture in order to cluster consistent regions. Some of these features are commonly used in edge contour representations. For our purposes, the object's boundaries are helpful to infer about spatial characteristics as they depict: depth discontinuity, texture boundary, noise, reflection and illumination changes. Methods to detect edges can be found in [163, 164, 165].

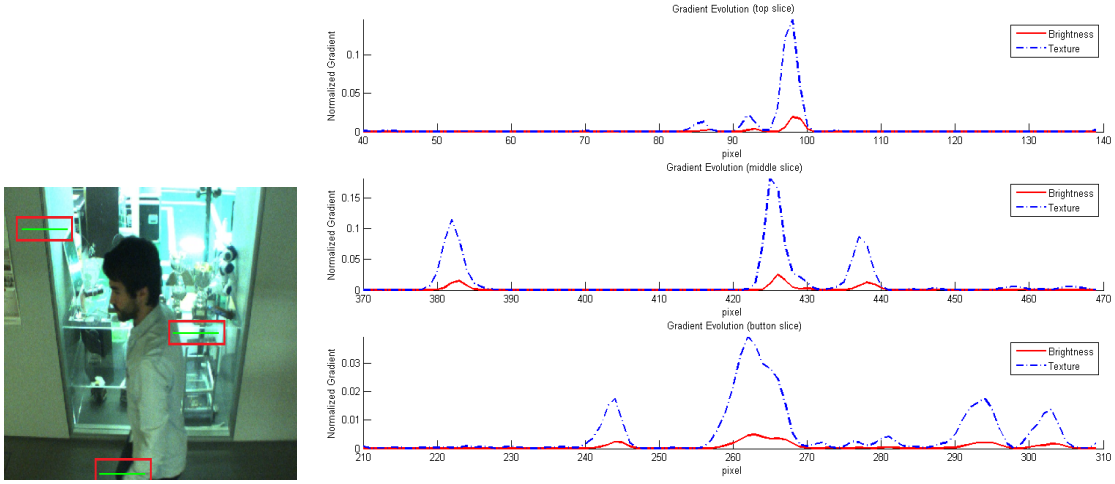
Figures 5.2(b), 5.2(c) and 5.3(b) compare the magnitude gradient for brightness[3] and texture. As it can be seen, texture magnitude gradient supports more information on the

[3]Firstly, the image is convolved with a Gaussian operator to remove noises and destabilizing high frequencies. Then, spatial derivatives are approximated by convolving the result with the fourth-order stencil $[1,-8,0,8,-1]/12h$.



(a) Snow environment. (b) Brightness magnitude gradient. (c) Texture magnitude gradient.

Figure 5.2: 5.2(a) is the Snow Image: 5.2(b) and 5.2(c) are the visual representation of the gradient magnitude for both brightness and texture [1] in a blue-scale representation. 5.2(c) shows that the contrast of texture gradient magnitude is greater than the brightness gradient at boundaries. For that reason, texture gradient magnitude provides better information on the spatial properties of the image, nevertheless, the texture gradient magnitude is two times more computationally demanding than the magnitude brightness gradient.



(a) Surveillance Image. (b) Normalized magnitude gradient.

Figure 5.3: 5.3(a) - Surveillance image with three green slice regions represented: top, middle and bottom. 5.3(b) - Graphical representation of the normalized gradient magnitude of both brightness and texture for each of slices.

spatial nature of the moving objects since it is the gradient within the multidimensional space [1]. Texture and brightness magnitude gradients have similar profiles when disregarding a scale factor (see Fig. 5.3(b)); however, the texture requires two or three times more computational effort, resulting in a major drawback. Thus, texture information is used by merging operations to analyze the spatial decomposition obtained by the splitting operation.

5.2.2.1 Region Splitting

A regular decomposition is performed using a tree-based data structure called quadtree. This technique is appealing from the computational point of view since it enables non-iterative inference procedures [166]. Initially, the image is represented by a parent node whereas the four quadrants are represented by four child nodes (in a predefined order). Therefore, this is a special type of tree where all the nodes are either parents or leafs (see Fig. 5.4).

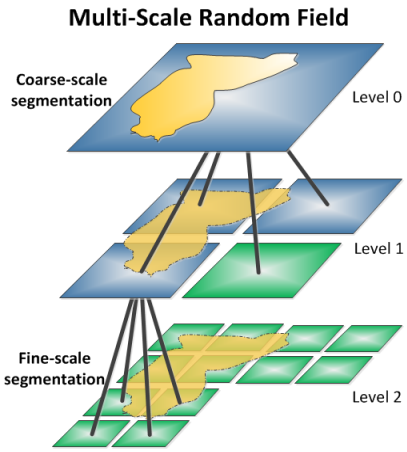


Figure 5.4: The quadtree structure is a coarse-to-fine tree with the ability to recursively divide the image into regions. Each region is represented by a node that can be a parent or leaf node (blue and green, respectively). It is an efficient multi-scale structure.

The quadtree is often used in image coding [167, 168] and video gaming [169] due to several advantages:

- Fast computations and low sensitivity to initial conditions (distracting local minima) and noise;
- Hierarchical graphs are manipulated as a whole. This leads to a unique statistical inference problem instead of a sequence of loosely related problems;

- The quadtree is very simple and yields in-scale causality properties enabling fast and noniterative inference procedures, similar to discrete and continuous Markov chain models [166].

The partition of the parent node into four children nodes is analyzed by a division criterion. To obtain the desired partitioning, several image's properties can be integrated into an objective function (forming the criterion) and used to infer if the node should be divided or not. Usually, a tree is completed when the division criterion cannot be satisfied by the current leaf nodes; however, due to time limitations, a maximum number of tree levels are also considered. In this research, the segmentation of the image into several regions must maintains spatial and motion coherences. The temporal requirement imposed by surveillance and robotic applications also requires the approach to be simplified and precludes the use of pure probabilistic methods as they are time consuming.

The division criterion is formed by a discriminative function based on two descriptive properties (features): brightness magnitude gradient and absolute temporal derivative.

$$y(n_i, X) = w(n_i)^T X + w_0, \tag{5.1}$$

where $w(n_i)$ is the weight vector that determines the orientation of the decision surface (hyperplanes) for the node n_i, w_0 is the bias that determines the location of the decision surface and X is the input vector. In a two class problem (parent or leaf), an input vector X_A is assigned to class C_1 if $y(.,X_A) \geq 0$ and to class C_2 otherwise [170]. For each node, the input vector is $X = (N_{bright}, N_{temp})^T$, where N_{bright} and N_{temp} are the number of occurrences for which the brightness magnitude gradient and absolute temporal derivative are higher than a given threshold (thr_{bright} and thr_{temp}, respectively). The weighted vector is $w(n_i) = \frac{1}{area(n_i)}(\alpha, \beta)$, where α and β are both real with $\alpha + \beta = 1$ (α controls the weight of the brightness feature and β controls the weight of the temporal feature). The value of these control parameters makes it possible to adjust the performance of the *HybridTree* method: the accuracy of the flow estimation or the computational demands of the robotic application. These parameters are set up in section 5.3. Although, the value of α should be higher than β because, in this way, the splitting process is more influenced by the spatial derivative (for instance, edges and corners), which usually have better information about the apparent motion.

Descriptive features capture the motion expectation during a sequence of images and are used to control the overall computational efficiency. The main objective of the splitting method is to guide the *sensing* phase along the optical flow computation by detecting large homogeneous regions and regions without apparent motion. Figure 5.5 shows the result of the splitting method using two consecutive frames of a rally sequence. As it can be

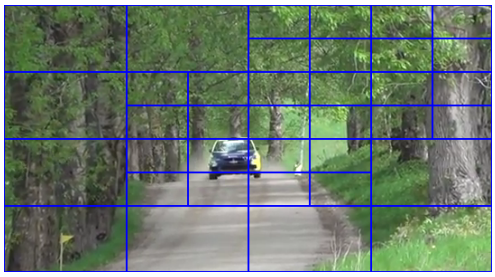


Figure 5.5: Rally sequence - image of a rally car moving, dust, bushes and tree leaves. Splitting method - The discriminative function based on temporal derivative and brightness gradient allows to divide the image into distinct regions.

seen, the method is successful by gradually dividing the image regions. Even without a merging operation and focusing on the size of the nodes, it is possible to infer that the most significant movement occurs in the central and upper right areas of the image. The resulting nodes with small areas represent zones with higher motion expectation (spatial and temporal combination) and nodes with large areas mean that the expectation is lower, that is, regions without apparent motion. From the optical flow point of view, large cells represent regions of smooth motion while smaller regions represent motion discontinuities or regions with higher texture.

The splitting method based on the quadtree structure induces non-stationarity in space, which means that the distribution of leafs depends on the correlation between ancestor nodes. This may result in a blocky looking image segmentation, Fig. 5.5. Several techniques have been proposed to prevent these undesired effects. This study uses a merging phase to increase the regions' coherence, enhancing the computational efficiency and the performance of the optical flow, as will be clearly described later.

5.2.2.2 Region Merging

This operation eliminates spurious regions by merging adjacent nodes. The technique, applied to the leaf nodes, gradually merges regions that belong to the same context (object and motion), see Figs. 5.6(a) and 5.6(b). The neighborhood of each node is evaluated using a similarity criterion that enforces the descriptive features: dominant color, absolute temporal derivative and magnitude gradient for both brightness and texture. The similarity criterion comprises relevant characteristics to assess if two nodes can be merged. The previous splitting operation uses less information than the current merging phase. This intentional procedure aims to create a more relaxed criterion during the region splitting in order to detect and characterize motion discontinuities and to avoid heavy computations. Thereafter, the merging operation analyzes the discrete regions and evaluates the coherence between each region and its neighbors.

(a) (b)

Figure 5.6: Rally sequence (movement of a rally car and bushes). Merging Phase - 5.6(a) is the result of an intermediate merging stage (red). 5.6(b) is the final result of the image decomposition based on the splitting-merging method (green).

Two nodes are merged when two main conditions are satisfied, that is, when the similarity criterion is met and the neighbor regions have the same size. The similarity criterion requires a distance measurement to evaluate the success of the merging operation. Assuming the descriptive features as being multivariate and normally distributed, then the feature vector has four dimensions, $X \in \Re^4$, and the difference between two regions, n_1 and n_2, can be measured by a Mahalanobis squared distance of samples (following a Chi-square distribution).

The similarity between nodes is considered in terms of difference between both mean vectors [171], $\overline{X_i}$, and the covariance $\hat{\Sigma}$.

$$\hat{\Sigma} = \frac{\left[(area(n_1)-1)\hat{\Sigma}_1 + (area(n_2)-1)\hat{\Sigma}_2\right]}{(area(n_1)+area(n_2)-2)}, \tag{5.2}$$

where $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ are the sample covariance matrices of each node and $\hat{\Sigma}$ is the estimated common nonsingular covariance. This equation can be simplified since the areas (number of samples) of both nodes are equal.

Yielding the sample means $\overline{X_1}$ and $\overline{X_2}$:

$$\Lambda^2 = (\overline{X_1} - \overline{X_2})^T \hat{\Sigma}^{-1} (\overline{X_1} - \overline{X_2}), \tag{5.3}$$

$\hat{\Sigma}$ is positive-definitive and then Λ is a metric that evaluates the distance between two regions by considering the mean characteristics and confidence (represented by the covariance). $\overline{X} = \left[r_{bright}, r_{temp}, r_{texture}, d_{color}\right]^T$ is the mean vector, where r_x is a ratio that considers the number of values above a given threshold. The r_{bright} and r_{temp} have already been obtained from previous splitting operation, the $r_{texture}$ is computed similarly to brightness and temporal features, and the dominant color, d_{color}, is the mean color

calculated using the normalized hue-channel in a HSV (hue, saturation and value) color space (the hue-channel scale is normalized between 0 to 1).

A significance level, Q, in the Chi-square distribution with one degree of freedom, $\chi(.)$, allows to change the similarity level imposed during the merge evaluation. Neighbors with a similarity $\Lambda < \chi(Q)$ are merged, for instance, $Q = 5\%$. The merging operation can be seen as an iterative process where each region can only be clustered to a neighbor in the same iteration. When the similarity criterion is not met during an entire iteration or when a maximum number of iterations is reached, the merging operation is concluded.

Images with objects containing distinct motions usually lead to a balanced tree with many tree levels and, consequently, each node represents a small region. This results in a higher computational effort as the tree-based algorithm takes more time to converge; however, the resulting tree provides a meaningful guideline for the *sensing* phase. Nevertheless, if the significance level (Q) is not properly chosen, the practical advantages of the tree structure are partly or completely lost. The major goal of the proposed splitting-merging method is to detect images with large and textureless regions since it improves the computational efficiency and performance of the *sensing* operation (performing a guided optical flow estimation).

5.2.3 Sensing

The image decomposition occurred in a previous phase is used by the *sensing* operation to guide the flow estimation through a hierarchical combination of local and global differential optical flow methods, namely, Lucas-Kanade and Horn-Schunck.

Differential methods are widely used because they allow formulating efficient and reliable optical flow techniques. These formulations use several assumptions, namely, brightness consistency and temporal persistence, which lead to a well-known motion constraint:

$$\nabla I^T \cdot \mathbf{v} + I_t = 0, \tag{5.4}$$

where $\nabla I = (I_x, I_y)^T$ denotes spatial intensity gradient[4] and is computed using derivative operators. $\mathbf{v} \equiv (u, v)^T$ is the optical flow (horizontal and vertical velocities) and I_t denotes the temporal derivative at time t. Due to the brightness constraint, the temporal derivative will always cancel the inner product between the spatial gradient and flow vector. The two-dimensional motion constraint, Eq. 5.4, is the basis of the optical flow; however, there is only one equation for two unknown variables, which means that the

[4]This chapter follows the notation used in [72].

measurements are underconstrained and a unique solution cannot be obtained for a single pixel. This is known as aperture problem and represents the inability to measure or estimate the motion in directions or regions that do not exhibit distinguishable characteristics. In those situations, it is only possible to infer about the velocity component in the same direction of the spatial gradient (normal to the image edges).

Differential-based techniques use the spatio-temporal intensity derivatives to estimate optical flow since they are easy to compute; however, additional constraints are required to obtain a unique solution for the ill-posed problem defined by Eq. 5.4. These techniques can be local [69] or global [68], according to the neighbor concept used to estimate the optical flow. More recent approaches combine both concepts using energy functionals and iterative solvers, like Gauss-Seidel and Successive Over Relaxation to minimize the formulation [21, 82].

This research focuses on efficient techniques as core for the optical flow computation. Therefore, colored versions of LK and HS are recasted in order to increase the overall performance without compromising the efficiency, see section 5.3. Furthermore, a color image is represented by their channels, $I(\mathbf{x}) = (I_1(\mathbf{x}), I_2(\mathbf{x}), I_3(\mathbf{x}))$, where $I_i(\mathbf{x})$ denotes a single channel and $\mathbf{x} = (x, y, t)$.

5.2.3.1 Multichannel Lucas-Kanade

Local techniques assumes that surrounding pixels belong to the same surface and are likely to move together [172, 72] (spatial coherence assumption). Thus, several motion constraint equations are combined using the local neighbors to determine the flow that minimizes the sum of the constraints over the neighborhood. These neighbors introduce additional constraints, causing a well-stated and overconstrained system. Therefore, assuming that neighbor pixels moves coherently and share the same flow, the system can be solved for each pixel position by a standard least-squares regression (considering a quadratic error norm). This is the principle behind the Lucas-Kanade method [69], originally designed for information on brightness.

The local method used as part of this research is a multichannel version of LK.

$$min_{u,v}\, E_{LK} = \sum_{i=1}^{3} G_\sigma * \left[I_{ix}(\mathbf{x}).u + I_{iy}(\mathbf{x}).v + I_{it}(\mathbf{x}) \right]^2, \tag{5.5}$$

where I_{ix}, I_{iy} and I_{it} are spatial and temporal derivatives for a single channel, $i \in \{1,2,3\}$, and G_σ denote a Gaussian convolution kernel with deviation σ, which controls the contribution of the neighbors. Equation 5.5 can be written in a more compact

form using motion tensors. A structure tensor, T_{LK}, is obtained by coupling all the channels [82].

$$T_{LK}\left(I(\mathbf{x})\right) = \sum_{i=1}^{3} G_\sigma * \nabla_3 I_i(\mathbf{x}).\nabla_3 I_i(\mathbf{x})^T; \tag{5.6}$$

$$min_{\mathbf{w}}\, E_{LK} = \mathbf{w}^T T_{LK}\left(I(\mathbf{x})\right)\mathbf{w}, \tag{5.7}$$

where $\nabla_3 I_i(\mathbf{x}) = (I_{ix}(\mathbf{x}), I_{iy}(\mathbf{x}), I_{it}(\mathbf{x}))^T$ and $\mathbf{w} = (u, v, 1)^T$. A disadvantage of this technique is the size of the neighborhood, σ, as it is not enough to correctly estimate the flow caused by images with textureless regions. Applying the least-squares to Eq. 5.5:

$$\begin{bmatrix} \sum_{i=1}^{3} G_\sigma * I_{ix}^2 & \sum_{i=1}^{3} G_\sigma * I_{ix}I_{iy} \\ \sum_{i=1}^{3} G_\sigma * I_{ix}I_{iy} & \sum_{i=1}^{3} G_\sigma * I_{iy}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} =$$

$$= - \begin{bmatrix} \sum_{i=1}^{3} G_\sigma * I_{ix}I_{it} \\ \sum_{i=1}^{3} G_\sigma * I_{iy}I_{it} \end{bmatrix}. \tag{5.8}$$

Surely, the solution exists when its system matrix is invertible (nonsingular) which happens when it has rank 2, both eigenvalues are not zero and it can be solved using Cramer's rule. Textureless regions do not provide sufficient gradient information which leads to a singularity matrix, precluding to perform a reliable estimation. This is a major drawback for local methods as they do not assure a reliable estimation of the optical flow in the entire image (but a confidence measurement can be obtained based on the smallest eigenvalue [60]).

5.2.3.2 Multichannel Horn-Schunck

Global methods use a regularization term to avoid singularities. Horn-Schunck [68] was the first global method and it focuses on minimizing a quadratic error that is easy to solve (the function is convex). The smoothness term makes it possible to propagate the neighborhood information across large and uniform intensity regions. However, the method is very sensitive to noise and the average induced by the smoothness constraint blurs the flow across the motion boundary [72].

By recasting the HS, the data term is extended to a multichannel formulation by coupling all the channels into a motion tensor, T_{HS}. This approach follows the energy-based formulation presented in [82].

$$T_{HS}\left(I(\mathbf{x})\right) = \sum_{i=1}^{3} \nabla_3 I_i(\mathbf{x}).\nabla_3 I_i(\mathbf{x})^T; \tag{5.9}$$

Maintaining a quadratic penalizer, a colored version of the classic HS method can be reformulated as:

$$min_{\mathbf{w}}\, E_{HS} = \int_I \left(\mathbf{w}^T T_{HS}\left(I(\mathbf{x})\right) \mathbf{w} + \lambda \left|\nabla \mathbf{w}\right|^2 \right) dxdy, \tag{5.10}$$

where λ is the regularization constant or Lagrange multiplier, $\mathbf{w}^T T_{HS}\mathbf{w}$ is the square of the motion constraint, Eq. 2.5, and $|\nabla \mathbf{w}|^2 = |\nabla u|^2 + |\nabla v|^2$. The function presented by Eq. 5.10 includes two terms: the data conservation term which penalizes deviations from the motion constraint, and the smoothness term, also known as regularity term, which penalizes deviations from the smoothness flow assumption. This function can be minimized by Euler-Lagrange equations with Neumann boundary conditions, $\mathbf{n}^T \nabla u = 0$ and $\mathbf{n}^T \nabla v = 0$.

$$\breve{I}_x{}^2 u + \breve{I}_x \breve{I}_y v + \breve{I}_x \breve{I}_t - \lambda \Delta u = 0; \tag{5.11}$$

$$\breve{I}_x \breve{I}_y u + \breve{I}_y{}^2 v + \breve{I}_y \breve{I}_t - \lambda \Delta v = 0, \tag{5.12}$$

where $\breve{I}_j = I_{1j} + I_{2j} + I_{3j}$ with $j \in \{x, y, t\}$, and Δ denotes the spatial Laplace operator that is numerically approximated by finite differences (considering a rectangular grid spacing of h_x and h_y for x and y-directions). All the channels are equally important however, an additional weighted function can be applied in order to decrease the contribution of a channel while estimating of the optical flow. The major advantage of the HS is the fact that it includes smoothness term that enables a *filling-in effect* in locations without distinguishable characteristics, that is, homogeneous regions ($|\nabla I| \approx 0$). Even though the HS method propagates the neighborhood information across large uniform intensity regions (depending on λ) and its computation cost is higher than in local methods [173].

The CLG method [21] extends the LK and HS methods by considering a structure tensor that enables an energy-based formulation. As previously explained, the formulation leads to an iterative solver that minimizes the energy function. However, the computational cost involved is not affordable for the objectives of this research and, therefore, both approaches are combined in a different and yet reliable way.

5.2.3.3 Hybrid optical flow

The methods' formulations presented in the previous sections have two major concerns, namely, the fact that motion discontinuities remain untreated (quadratic penalizer are used) and the linearization caused by the first-order approximation of the Taylor expansion of the motion constraint is only valid for small optical flow values, such as subpixel

displacements. In addition, large displacements cause aliasing and multimodal energy functionals which can cause the minimization process to stop at local minimums [78].

To handle these situations, the proposed technique combines both methods in a hierarchical and tree-based structure that follows cognitive information about motion. Therefore, color versions of the LK and HS methods are embedded together into a multiresolution architecture with a refinement procedure between different scales. The optical flow constraint of Eq. 2.5 is changed to nonlinear formulation, $I(\mathbf{x}+\mathbf{w})-I(\mathbf{x})=0$, yielding Eqs. 5.13 and 5.14.

The architecture creates a pyramidal structure of downsampled images to deal with large motions, Fig. 5.1(a). The current flow is used at each pyramid level to warp the image at time $(t+1)$ towards the image at time t [72, 173]. The motion flow increments are obtained by minimizing energy functionals. Before downsampling the input images by a factor of $\tau \in (0,1)$, a low-pass Gaussian filter is applied with a standard deviation $\sqrt{2/4\tau}$ [82]. After this convolution, the images are sampled using the flow estimation from the coarser level and a bicubic interpolation.

$$min_{\delta \mathbf{w}_l} E_{LK} = G_\sigma * [I(\mathbf{x}+\mathbf{w}_l)-I(\mathbf{x})]^2; \tag{5.13}$$

$$min_{\delta \mathbf{w}_l} E_{HS} = \int_I \left([I(\mathbf{x}+\mathbf{w}_l)-I(\mathbf{x})]^2 + \lambda \left|\nabla(\mathbf{w}_l+\delta \mathbf{w}_l)\right|^2 \right) dxdy, \tag{5.14}$$

where $\mathbf{w}_l$ denotes the flow estimation and $\delta \mathbf{w}_l$ is the flow increment at pyramid level $l \in \{0,...,\#_{levels}-1\}$[5]. The energy functionals of Eqs. 5.13 and 5.14 are minimized with regard to $\delta \mathbf{w}_l$.

Writing $I_{x,l}$ and $I_{y,l}$ as spatial derivatives of $I(\mathbf{x}+\mathbf{w}_l)$ and using an approximation to the first order of the Taylor expansion, the non-linearity can be removed from the equations above and, thus, the temporal derivative $I_z = I(\mathbf{x}+\mathbf{w})-I(\mathbf{x})$ is expressed by:

$$I_{z,l+1} \cong I(\mathbf{x}+\mathbf{w}_l)-I(\mathbf{x})+I_{x,l}\delta u_l + I_{y,l}\delta v_l;$$

$$I_{z,l+1} = I_{x,l}\delta u_l + I_{y,l}\delta v_l + I_{z,l}. \tag{5.15}$$

This linearization makes it possible to reformulate Eqs. 5.13 and 5.14 using the tensor notation:

$$E_{LK} = \delta \mathbf{w}_l^T T_{LK}\left(I(\mathbf{x}+\mathbf{w}_l)\right)\delta \mathbf{w}_l; \tag{5.16}$$

[5]The $l = \#_{levels}-1$ is the pyramid level at coarsest resolution and l=0 is the pyramid level at full resolution.

$$E_{HS} = \int_I \left(\delta \mathbf{w}_l^T T_{HS} \left(I(\mathbf{x} + \mathbf{w}_l) \right) \delta \mathbf{w}_l + \lambda \left| \nabla (\mathbf{w}_l + \delta \mathbf{w}_l) \right|^2 \right) dxdy. \tag{5.17}$$

Therefore, minimizing of the energy functional in Eq. 5.16 is a straight forward process and Eq. 5.17 yields the following Euler-Lagrange equations:

$$\breve{I_{x,l}} \left[\breve{I_{x,l}} \delta u_l + \breve{I_{y,l}} \delta v_l + \breve{I_{z,l}} \right] - \lambda \Delta u_{l+1} = 0; \tag{5.18}$$

$$\breve{I_{y,l}} \left[\breve{I_{x,l}} \delta u_l + \breve{I_{y,l}} \delta v_l + \breve{I_{z,l}} \right] - \lambda \Delta v_{l+1} = 0, \tag{5.19}$$

where $u_{l+1} = u_l + \delta u_l$ and $v_{l+1} = v_l + \delta v_l$.

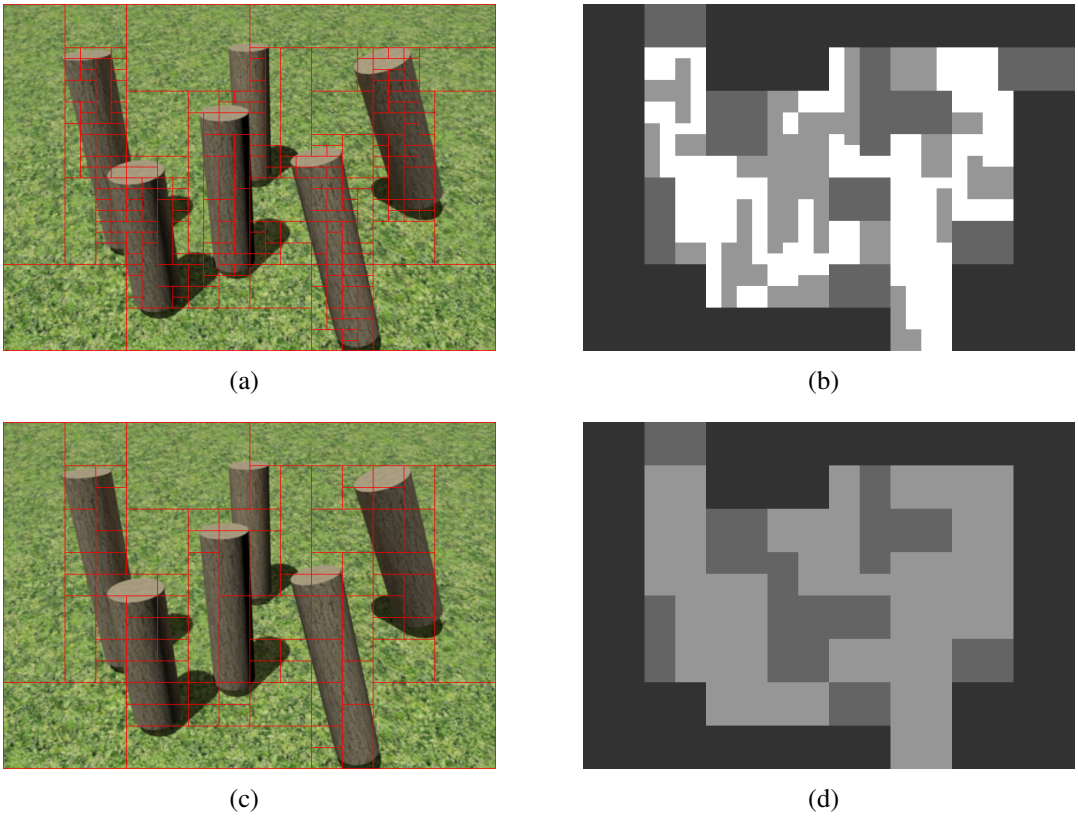
The optical flow at each pyramid level is divided into two variables, the flow of the coarser level and the unknown flow increment of the current pyramid level. The problem can be linearized since the flow increment is small as a result of the multi-resolution approach. This coarse-to-fine approach starts from a coarsest version of the original problem and refines the coarser estimation flow to warp and compensate the input image before the next finer level. The motion incremental estimation, $\delta \mathbf{w}_l$, is obtained between the partially registered images. The optical flow estimation, $\mathbf{w}_l$, is updated at the end of each level. Thus, the warping scheme implicitly removes the nonlinearity.

In order to discretize the Euler-Lagrange equations, spatial derivatives are approximated via a central finite differences by a fourth order approximation and using the stencil $[1,-8,0,8,-1]/12h$, where h is the grid size (considering a rectangular grid). Temporal image derivative is approximated by a two-point stencil $[-1,1]$ and the spatial flow derivatives are approximated by a second order stencil $[-1,0,1]/2h$.

LK and HS formulations use quadratic penalizers (see Eqs. 5.13 and 5.14). These make it possible to achieve a respectable performance and, at same time, they are computationally suitable for the efficiency requirements addressed by this research. However, motion discontinuities represent a major drawback for these formulations since they are not very robust in the presence of outliers, especially the HS because it is a global approach.

To deal with motion discontinuities more properly, the LK and HS are spatially combined during the optical flow estimation, creating a hybrid approach that combines the advantages of both methods. This symbiotic combination between methods is guided by the *expectancy* procedure. Each region is classified into five admissible classes according to its relative size: finest, fine, medium, coarse and coarsest. The classification of regions into several classes allows interpreting the image using high level information about the characteristics of the motion. The classification is performed considering the relative size

of each region because different region sizes incorporate different types of motions. For instance, the lower the size more local information a region has. Therefore, each class represents a different layer of visual details. This behavior is similar to the behavior of human beings as they resort to distinct perceptive levels to infer about the motion.



(a) (b)

(c) (d)

Figure 5.7: 5.7(a) and 5.7(c) depict the image decomposition of the "Blow" sequence [2] under different bias w_0 and similarity levels. 5.7(b) and 5.7(d) denote the classification of regions according to their sizes. The classification uses the relative parameters, 50%, 17%, 7% and 3.5% of the full image size. The five classes are visually represented in shades of gray, black represents the coarsest class and white the finest class. Figure 5.7(b) has 4 classes represented and Fig. 5.7(d) has only 3.

Selecting an appropriate bias and similarity level allow to classify regions considering different scopes (for instance, the scope in Fig. 5.7(a) is smaller than Fig. 5.7(c)). Obviously, the classification affects the estimation of the optical flow (efficiency and performance) and the classification parameters (relative sizes) must be adjusted according to the requirements of the application. Therefore, cognitive information is used to select a local or global methodology according to the spatial characteristics of each class. The local optical flow method performs better in small regions and the global method performs better in larger regions. This principle is applied to assign the optical flow technique to each class. For instance, the LK is assigned to the finest and fine classes, and the HS to

the medium, coarse and coarsest classes. The configuration parameters of the methods are adjusted between different classes. Therefore, the image decomposition allows the parameter to be enhanced, which means that the configuration can be properly set up for applications with different requirements. For instance, the neighborhood size for the LK version can be smaller for the finest regions and the smoothness term of the HS should be greater for the coarsest regions.

Usually, discontinuities cover only a small fraction of all pixels and, therefore, the hybrid approach uses local methods with small neighborhood to capture these discontinuities more efficiently, relatively to the non-quadratic penalizers. The filling-in effect with different strengths allows to propagate neighborhood information to large and textureless regions. Only optical flow differential formulations with quadratic penalizers are considered because they increase the overall computational efficiency of the *HybridTree* technique. This happens because the error function is convex and the minimization is simpler. Therefore, the hybrid approach incorporates the advantages of each type of technique, namely, the robustness under noise and the *filling-in effect*. This approach also enhances parameters taking into account the expected motion of each image region to assign the most suitable method and its parameter configuration.

5.3 Results

A comprehensive set of experiments were conducted as part of this work [6]. The experiments aims to analyze and understand the behavior of the proposed *HybridTree* technique, namely, the image decomposition and the optical flow architecture that combines local and global differential optical flow techniques. The first experiments focus on the *expectancy* operation. Then, the performance of the optical flow estimation is presented and analyzed in detail. The experiments conducted use personal images and two major databases (see [3][7] and [2][8]).

5.3.1 Expectation

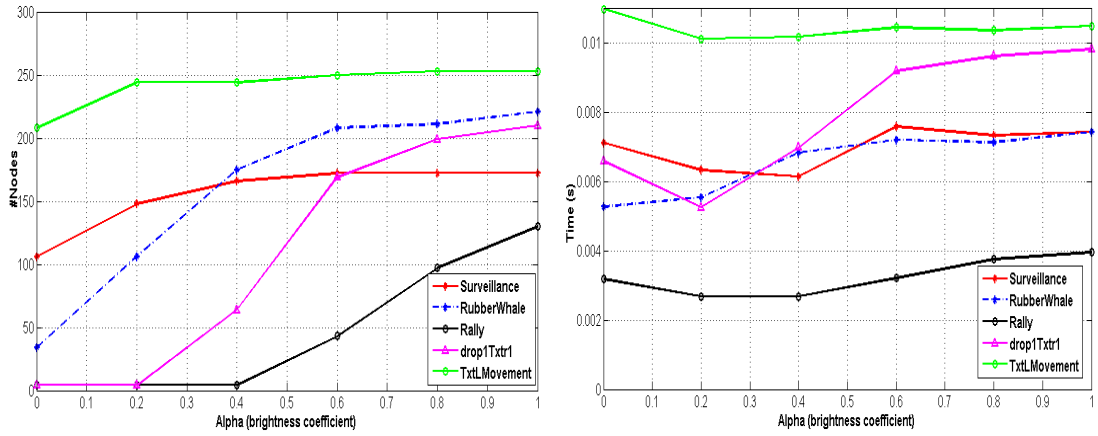
The *expectation* is evaluated by considering the splitting and merging operations individually. This way, the evaluation provides a baseline to characterize the behavior of the image decomposition. The characterization takes into consideration relevant properties,

[6] All the results were obtained with a I3-M350 2.2GHz and no parallel programing.
[7] http://vision.middlebury.edu/flow/eval/
[8] http://visual.cs.ucl.ac.uk/pubs/flowConfidence/supp/

such as the region's size, merging and splitting ratio, and computational time. The influence of different descriptive features (brightness, temporal, texture and color) is also studied. The tests were conducted under several different conditions and image sequences. Particularly, five different sequences are used to evaluate *expectation*: surveillance, rally, rubber whale [3], "drop1Txtr1" [2] and "TxtLMovement" [2]. The first three represent real environments and the last two represent virtual scenarios.

5.3.1.1 Splitting Operation

The bias factor (w_0) defines the decision boundary for the spitting procedure. It should be adjusted according to the minimum value of spatial and temporal displacements that are supposed to occur in the environment. Therefore, higher and positive bias means that the quadtree is only interested in greater displacements and the algorithm creates larger nodes.



(a) (b)

Figure 5.8: Splitting operation - 5.8(a) presents the number of nodes that were split and 5.8(b) presents the time cycle spent for different brightness coefficients (α). During these experiments, the temporal coefficient is set to $\beta = 1 - \alpha$ and the bias factor to $w_o = 0.12$, for all the sequences. This is a low bias factor, meaning that we are interested in a detailed tree (images are represented by many small regions). The results confirm that the brightness magnitude gradient is extremely relevant during the splitting operation, as it increases number of regions obtained.

Figures 5.8(a) and 5.8(b) analyze the contributions of the brightness and temporal features during the splitting process. The α and β coefficients control the influence of the brightness magnitude gradient and the absolute temporal derivative, respectively. During these experiments, the temporal coefficient was set to $\beta = 1 - \alpha$. When $\alpha = 0$, the brightness feature does not contribute to the splitting process, and the operation follows

the temporal information only. Moreover, when $\alpha = 1$ the operation only uses the brightness information and the temporal contribution is disregarded. Comparing both situations, it is possible to infer that the brightness feature seems to have a strong relevance for the splitting operation. For instance, the spatial context originates more nodes than the temporal context; however, they should both be combined in order to control the number of regions that are obtained. The coherence regions obtained at the end of the process guide the optical flow estimation and, therefore, the coefficients are extremely important. The balance between the number of nodes and the operation time required to decompose the image is controlled by α. In this research, the computational effort is the major criterion; however, the value of α must lead to coherent nodes. The $\alpha = 0.6$ is an acceptable value for the sequences tested since the overwhelming majority of sequences took less than 10ms and our surveillance sequence took 6.1ms to be decomposed.

In conclusion, splitting operations can be adjusted to focus on the spatial or temporal characteristics of the images sequences. Depending on the computational performance, it is possible to manage the time spent during the splitting procedure by adjusting the decision boundary (it influences the number of nodes and, consequently, the merging operation and the *sensing* phase as well).

5.3.1.2 Merging Operation

The behavior of the merging operation is studied by adding different descriptive features to the merging formulation. Then, the convergence times are analyzed for different image sequences and under different formulations. To remove the influence of the splitting operation, the experiments are conducted using a complete and balanced tree with 4 levels which results in 256 leaf nodes at the beginning.

Figures 5.9(a) and 5.9(b) study the convergence of merging over six iterations and considering several similarity levels (C-value), such that $c \in \{0.02, 0.03, ..., 0.12\}$.

Only the results for the surveillance sequence are presented since similar graphics were obtained for other sequences. Relaxing the level of similarity increases the merging rate and, in general, the area value stabilizes in the fourth iteration, which means the process converges. A lower similarity level leads to fewer nodes being merged and, consequently, the operation converges in a couple of iterations. Contrarily, regions with less similarity levels can be merged with a large C-value. Therefore, the convergence time decreases due to a strong decrease in the number of nodes by the first iteration. Hence, the average value of the area increases rapidly.

Figure 5.9(c) shows the time elapsed during the merging operation and over the iterations. Figure 5.9(d) presents the accumulative number of merges which occurs during

the six iterations and over time. Real sequences have a higher merging ratio, for instance the rally, surveillance and rubber whale. This result is also verified in other similarity levels. Virtual images intend to approximate real images using a controllable and artificial environment. Our result shows that, real test scenarios should not be ignored during the algorithm's evaluation, even with very good virtual images (as is the case presented) because many object properties are difficult to synthesize, such as the texture.

Images with less resolution lead to shorter time cycles. Both virtual and surveillance sequences have the same resolution; however, the convergence time for the surveillance is 12ms and instead of 19ms with the "drop1 Txtr1". The time cycle is also lower for the surveillance sequence. The operation took 12ms to converge; however, this value is substantially reduced when the splitting process uses a proper division criterion (the quadtree cannot be complete and balanced).

The merging formulation presented in this research uses four descriptive features. Several experiments have been conducted in order to evaluate the impact of each feature on the overall performance. Figures 5.10(a) and 5.10(b) present the accumulative number of the merged nodes over time, considering the "TxtLMovement" and Surveillance sequences, respectively. The method is analyzed by taking into account different merging formulations: four simple (with one feature only) and three complex (with more than one feature) formulations are considered in this evaluation. For simple brightness, temporal and color formulations, the results show that the number of nodes converges quickly because they not constrain the similarity between regions and it is simple to calculate each feature. Although computing texture requires more time and it is more stable because the behavior is quite similar for all test sequences. Figures 5.10(a) and 5.10(b) reinforce the results presented in Fig. 5.3(b), regarding the importance on texture. It can be seen that the texture magnitude gradient is very stable and converges after the third iteration; however, it takes a considerable amount of time to compute under real test sequences.

Complex formulations based on brightness and temporal features assume a very interesting behavior because they provide a number of merges close to the original (which uses the four descriptive features), but for a very short period of time (see Fig. 5.10(b)). This formulation can be used in applications that have a very restricted time requirements.

One last set of experiments was conducted in order to analyze this previous result better. In these experiments, the performance of the "brightness + temporal" formulation is compared to the original. Formulations based on brightness and temporal features induce a strong relaxation (more than 230 nodes are merged) and the operation converges during the fifth iteration (less than 3ms), see the continuous lines in Fig. 5.11(a).The merging operation based on the four descriptive features takes a longer time to converge since it restricts the similarity definition more; however, convergence happens during

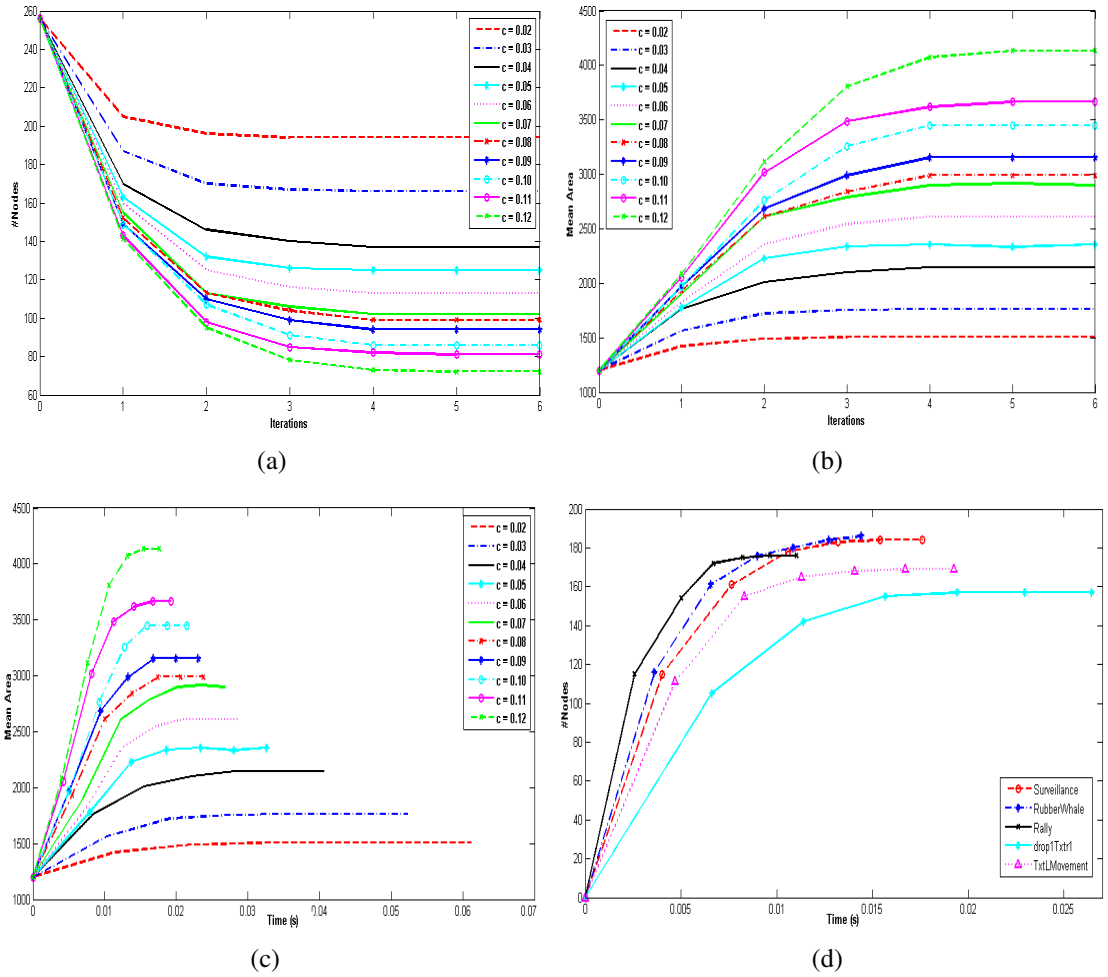
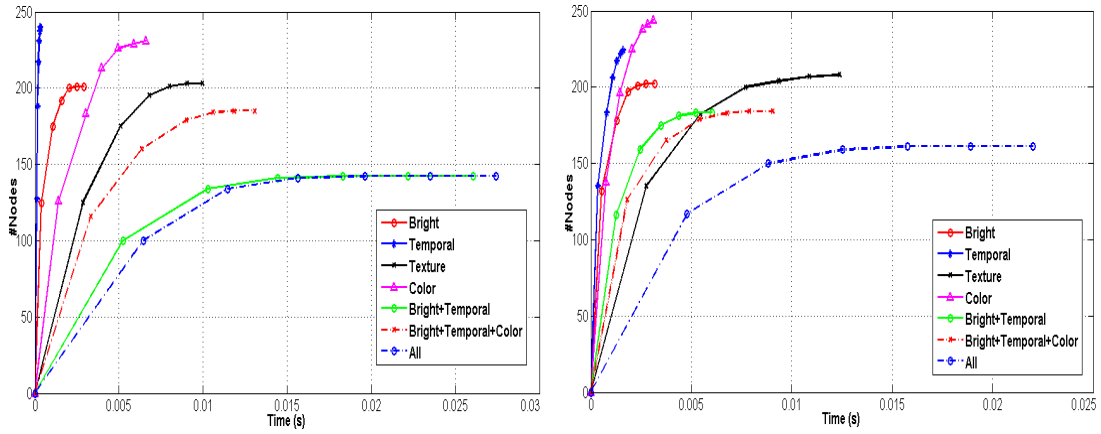


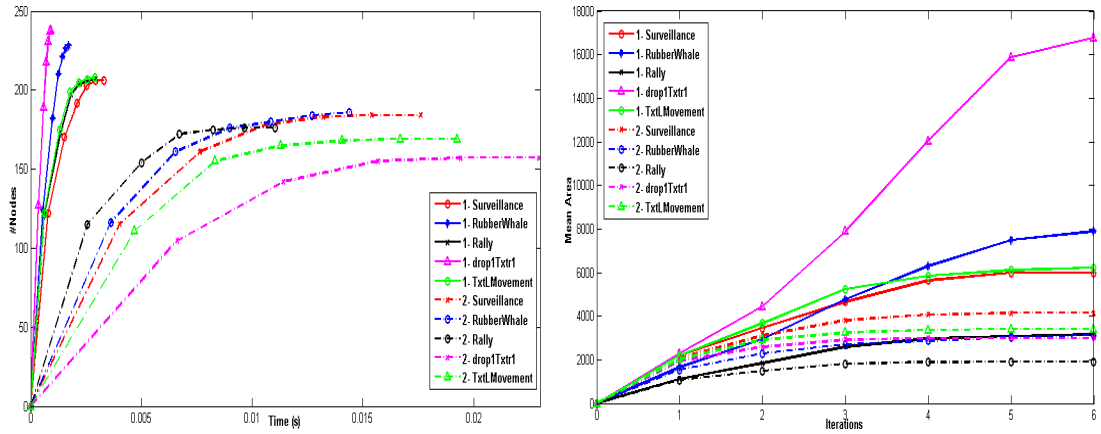
Figure 5.9: Merging operation for the surveillance sequence - 5.9(a) shows the evolution of the number of nodes during the iterations and for different similarity levels. 5.9(b) shows the mean area (in pixels) of the regions and 5.9(c) shows the time elapsed (in seconds). Finally, 5.9(d) compares the accumulative number of nodes that are reduced over time for different sequences and considering $c = 0.12$. As expected, increasing the similarity level leads to a higher number of nodes that are merged per iteration, which results in a higher convergence rate. In addition, the mean area increases during the merging operation and converges after 4 iterations.

the third iteration. Comparing Figs. 5.11(a) and 5.11(b) a clear advantage of merging based on brightness and temporal features is the time spent. The first formulation uses less information on the sequences, which allows lower time cycles. Although, in some applications this aspect can be very useful, the similarity level must be carefully chosen in order to avoid meaningless operations.



(a) (b)

Figure 5.10: The 5.10(a) and 5.10(b) present the accumulative number of merges over time when $c = 0.09$ for the "TxtLMovement" and Surveillance sequences, respectively. This experiment considers different formulations. All trials converge between the third (for complex formulations) and sixth iterations (for simple formulations). The original formulation (based on the four features and represented by a dashed blue line) is the trial with the lowest number of merges and the highest convergence time.



(a) (b)

Figure 5.11: Merging operation. Comparison between two merging formulations using $c = 0.12$. The index "1"- is the formulation based on brightness and temporal features, and the index "2"- is the original merging based on the four descriptive features. 5.11(a) compares the evolution of the number of nodes merged over time (in seconds) and 5.11(b) presents the average area of the regions. Although, the original formulation merges a lower number of regions, the regions obtained are more consistent and converge after the third iteration. Relaxing original merging allows to increase the number of merging nodes in a shorter period of time.

5.3.2 Sensing

The performance of the *HybridTree* technique is evaluated using a set of synthetic and real image sequences. By knowing the authentic velocity field, that is, the ground truth, it is possible to analyze the error. Although there is an amount of error measurements, in the literature, and the most used are the average Angular Error (AE) and the Endpoint Error (EPE). The AE was proposed by *Fleet and Jepson (1990)* [174] and used in *Barron et al. (1994)* [60]. It measures the error between the ground truth, $\mathbf{w}_g = (u_g, v_g, 1)^T$, and the estimated flow, $\mathbf{w}_e = (u_e, v_e, 1)^T$, by considering the unit vector normal for the velocity plane. In this case, the error is reported in degrees (see Eq. 5.20).

$$AE = arcos\left(\frac{\mathbf{w}_e^T.\mathbf{w}_t}{\|\mathbf{w}_e\|.\|\mathbf{w}_g\|}\right). \tag{5.20}$$

The simplest error measurement is probably the EPE as this is the square magnitude of the difference between the ground truth and the estimated flow. The EPE is useful to analyze the spatial structure of the errors [3], and can simply be viewed from a visual image, see Eq. 5.21.

$$EPE = \sqrt{(\|\mathbf{w}_e - \mathbf{w}_g\|)}. \tag{5.21}$$

In the literature, proposed optical flow techniques are often compared to the classical HS and LK formulations [60]. Comparing todays methods to the notable work conducted by *Barron et al. (1994)* [60] is not reasonable because it does not incorporate the most important modern practices, such as the hierarchical structure and the warping process.

Therefore, the results presented in this research aim to provide a reliable characterization of the performance and computational cost involved during the optical flow estimation for modern and standalone versions of LK and HS, presented in the previous section. In addition, the *HybridTree* (HY) is compared to another hybrid approach, named, CLG-2D [21]. The LK, HS and HY are implemented using similar schemes so that they can be compared more reliably. The CLG is implemented considering the information presented in [21]. The performance achieved by our implementation of the CLG method is very similar to that reported in [21], if not somewhat better (probably due to an improved combination of parameters). Extensive experiments were conducted using a large number of test sequences. All the techniques are implemented in C++ using the commonly used OpenCV library (version 2.4.3).

A parameter optimization method based on simulated annealing [26] is used to set up some parameters. The algorithm optimizes the parameters by considering the convergence time of the splitting and the merging operation (following the number of regions).

Table 5.1: Recommended values (they were experimentally obtained) for the parameters of the HybridTree optical flow technique.

Parameters:	Simulated Annealing	Value
w_0	No	0.05
α	Yes	$\alpha \in \{0.4, 0.8\}, \alpha = 0.6$
thr_{bright}	No	10
thr_{temp}	No	20
Q	Yes	$Q \in \{1\%, 10\%\}, Q = 3\%$
M_{iter}	No	3

The tuning method provides the parameters of the optical flow technique that enable a good performance; however, it is not possible to ensure that this is an optimal combination. Modern implementations of the HS and LK [23], and the CLG are considered baselines for the HY. Spatial derivatives are approximated using the fourth-order stencil, the temporal derivative is approximated by a simple two point stencil and a median filter is applied to intermediate flow values during the incremental optimization stages. The warping process is accomplished by a bicubic interpolation and the CLG uses the Charbonnier non-quadratic penalizer with a scaling parameter of 0.001. The image sequence is pre-smoothed by a Gaussian convolution that helps reduce noise and makes the image infinitely many times differentiable [82].

5.3.2.1 Optical flow estimation

Even thought the CLG and HY are both hybrid approaches, their schemes are completely different. Summarily, the CLG combines local and global methods in the same mathematical formulation. It resorts to non-quadratic and convex penalizers to deal with motion discontinuities. On the other hand, the HY method interprets the sequence of images and identifies areas with distinct motion features. This way, the information on the image is used to infer and to assign the technique that best suits each image region. The idea is to avoid non-quadratic penalizers so that the process is more computationally efficient. This research uses extended and standalone versions of LK and HS formulations, based on a hierarchy with a warping formulation (similar to CLG).

Figures 5.12(b), 5.12(e), 5.12(h), 5.12(k), 5.13(b) and 5.13(e) are the results obtained by the CLG method. Figures 5.12(c), 5.12(f), 5.12(i), 5.12(l), 5.13(c) and 5.13(f) show the results obtained by the HY method.

Comparing these results, it is possible to infer that the performance of the HY is generally better than CLG. This is easily be confirmed since the results of the HY are

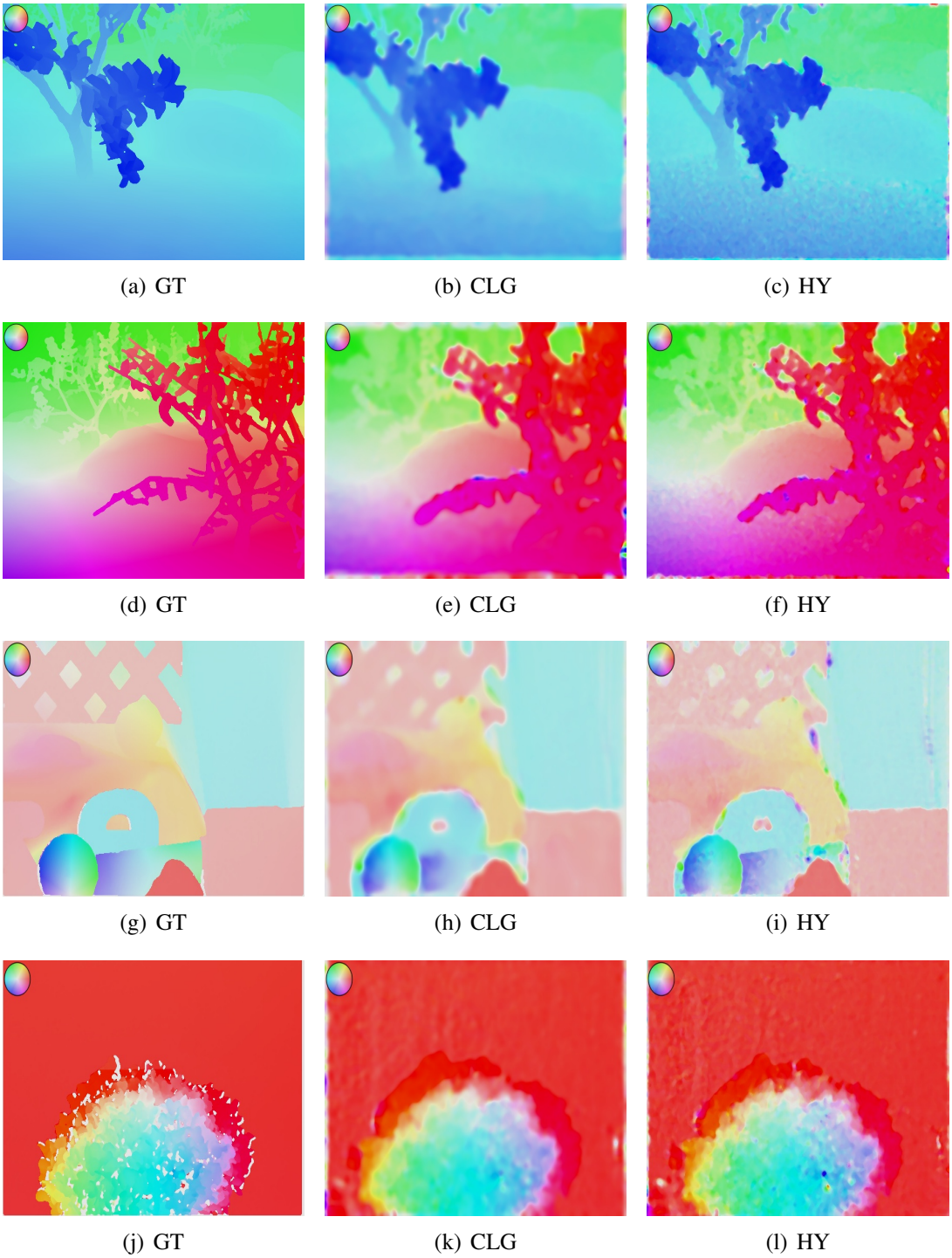
Table 5.2: Comparison between the *HybridTree* (HY), Combining Local and Global (CLG) and the colored versions of Lucas-Kanade (LK) and Horn-Schunck (HS). The performance of these methods are analyzed for several test sequences, considering full density and AAE - average angular error (°). *Dimetrodon*, *Grove2*, *Grove3*, *RubberWhale*, *Hydragea* and *Urban3* [3]. *Blow*, *Blow2* and *Drop1txtr1* [2].

Sequence	LK	HS	CLG	HY
Dimetrodon:	4.28°	6.13°	4.20°	**3.96**°
Grove2:	3.53°	3.79°	3.15°	**2.97**°
Grove3:	7.74°	8.13°	7.55°	**6.90**°
RubberWhale:	7.96°	9.06°	7.27°	**6.59**°
Hydragea:	5.85°	7.36°	5.87°	**3.90**°
Urban3:	6.30°	9.88°	5.66°	**5.22**°
Blow:	3.20°	2.32°	2.47°	**2.17**°
Blow2:	9.95°	9.27°	7.10°	**6.88**°
Drop1txtr1:	5.04°	5.03°	4.48°	**4.21**°

similar to those achieved by the CLG. However, they are more detailed in terms of motion discontinuities (see, for instance Fig. 5.12(c) and 5.12(f)). Even though the motion of the CLG is smoother, the regularization affects the objects' boundary in terms of data and smoothness even with non-quadratic penalizers. On the other hand, the HY results appear to be less smooth and, for this reason, this method captures the motion boundaries more effectively.

As it stands, the HY technique has a major disadvantage because it uses local methods in image regions that may not contain sufficient texture information to allow a proper estimation of the optical flow. For instance, focusing on the horizontal branch of the tree represented in Fig. 5.12(f), it is possible to see a small blue area that should be pink. This drawback can be improved by a filtering step after estimating the flow, for instance, applying a bilateral filter.

The resulting average angular error (AAE) measurements are listed in table 5.2 and the EPE measurements are presented in table 5.3. Both tables compare the performance achieved by the LK, HS, CLG and HY. The results were obtained using the optimization technique based on simulated annealing [26] to set up the parameters of each technique. As it can be seen, when implemented with modern techniques, classical methods perform satisfactorily for many cases. The performance gain achieved by a multiresolution approach (coarse-to-fine) combined with a warping process is significant when compared to classical formulations. The *HybridTree* technique improves the results obtained by the LK, HS and CLG. Adding cognitive information to the optical flow estimation makes it possible to increase the estimation performance. The HY does not incorporate the robusti-



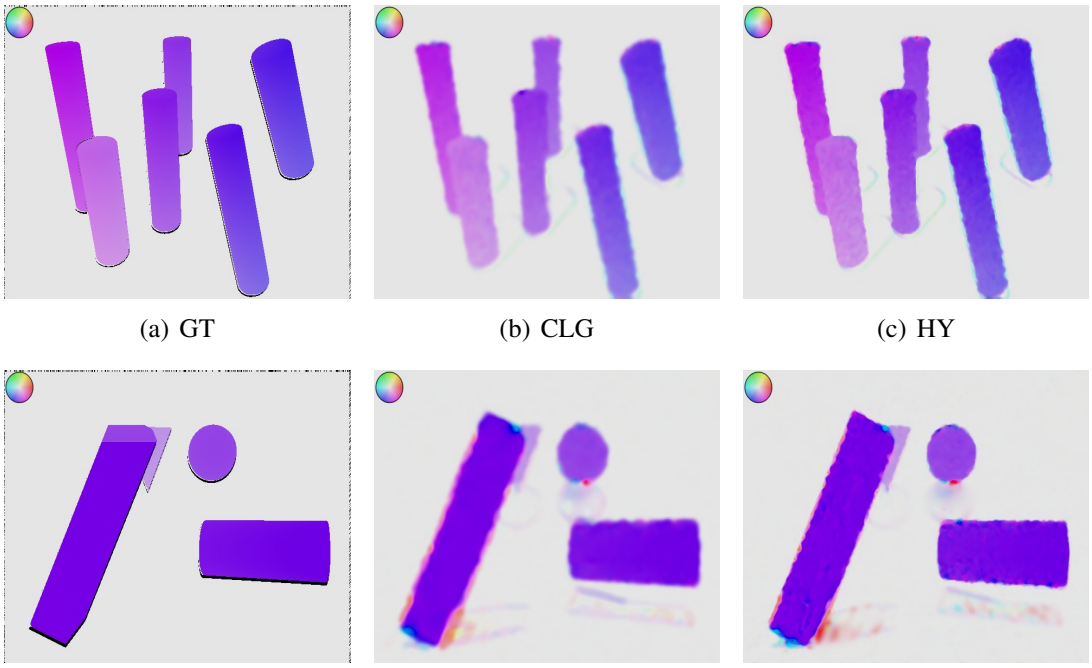
(a) GT (b) CLG (c) HY

(d) GT (e) CLG (f) HY

(g) GT (h) CLG (i) HY

(j) GT (k) CLG (l) HY

Figure 5.12: Results for some Middleburry sequences [3]. The first column is the ground truth (GT). The HSV color space is used to represent the direction (color) and magnitude (saturation) of the flow. The second color is the result obtained by the CLG. Finally, third column is the result obtained by the *HybridTree* optical flow. From top to button, the sequences are: *Grove2*, *Grove3*, *Rubberwhale* and *Hydrangea*.



(a) GT (b) CLG (c) HY

(d) GT (e) CLG (f) HY

Figure 5.13: Results for some synthetic sequences [2]. The first column is the ground truth (GT). The HSV color space is used to represent the direction (color) and magnitude (saturation) of the flow. The second color is the result obtained by the CLG. Finally, third column is the result obtained by the *HybridTree* optical flow. From top to bottom, the sequences are: *Blow* and *Drop1txtr1*.

Table 5.3: Comparison between the *HybridTree* (HY), Combining Local and Global (CLG) and the colored versions of Lucas-Kanade (LK) and Horn-Schunck (HS). The performance of these methods are analyzed for several test sequences, considering full density and EPE - average endpoint error (pixels). *Dimetrodon*, *Grove2*, *Grove3*, *RubberWhale*, *Hydragea* and *Urban3* [3]. *Blow*, *Blow2* and *Drop1txtr1* [2].

Sequence	LK	HS	CLG	HY
Dimetrodon:	**0.196**	0.290	0.222	0.202
Grove2:	0.260	0.292	0.234	**0.221**
Grove3:	0.959	0.952	0.910	**0.820**
RubberWhale:	0.267	0.341	0.217	**0.202**
Hydragea :	0.355	0.515	**0.351**	0.372
Urban3:	1.045	1.937	0.860	**0.850**
Blow:	0.159	0.121	0.130	**0.113**
Blow2:	1.415	1.337	0.901	**0.831**
Drop1txtr1:	0.238	0.254	0.207	**0.197**

Table 5.4: Comparison between the colored versions of the Lucas-Kanade(LK) and Horn-Schunck (HS), Combining Local and Global (CLG), and *HybridTree* (HY). The complexity of these methods are analyzed for several test sequences, considering full density and the computational time expressed by orders of magnitude relatively to HY. *Dimetrodon*, *Grove2*, *Grove3*, *RubberWhale*, *Hydragea* and *Urban3* [3]. *Blow*, *Blow2* and *Drop1txtr1* [2].

Sequence	LK	HS	CLG
Dimetrodon:	5.08	4.82	27.58
Grove2:	14.45	19.04	49.08
Grove3:	8.64	15.47	39.73
RubberWhale:	18.29	12.48	34.99
Hydragea:	1.58	3.22	7.12
Urban3:	7.64	7.75	21.37
Blow:	2.43	15.84	69.73
Blow2:	10.21	16.21	94.75
Drop1txtr1:	5.53	14.10	36.96

fication theory since non-quadratic penalizers are not used, and still the proposed method performs better than the CLG in all sequences tested, see tables 5.2 and 5.3.

5.3.2.2 Computational efficiency

The following experiment is studied with the efficiency of each technique. Although, the experiment tried to demonstrate the complexity of each method, it should not be seen as an absolute performance reference. The optical flow formulations of modern techniques increase the complexity of the estimation. When several complex assumptions are combined, for instance, non-linearity assumptions and non-quadratic and non-convex penalizers, the formulation obtained can lead to a poor and complex estimation procedure. This complexity leads to a computational cost that may compromise the usability of the methods for today's applications. The techniques were implemented and tested using a generic processing unit and without parallel processes. Table 5.4 presents the computational complexity expressed in order of magnitude relatively to the time duration of the HY estimation. For instance, the LK took 1.58 times more to process the *Hydragea* sequence comparatively to the HY. These results report to the experiments presented in tables 5.2 and 5.3.

Table 5.4 shows the complexity of each method, using the proposed technique as reference. The HY method is less complex than the LK, HS and CLG. More specifically, the complexity of the CLG is on average 42.36 times higher than the HY, while the LK is 8.20 and the HS is 12.10 times higher than HY. Thus, the optical flow estimation conducted

by the HY is more computationally efficient than the standalone versions of the LK and HS. This is a remarkable result because the HY technique resorts to similar LK and HS formulations. Even though this technique takes more time to interpret the image (using the *expectancy* operation), the computational gain made possible by the incorporation of structural information in the optical flow estimation is significant. Therefore, combining local and global methods using high level information is computationally rewarding and allows a better estimation.

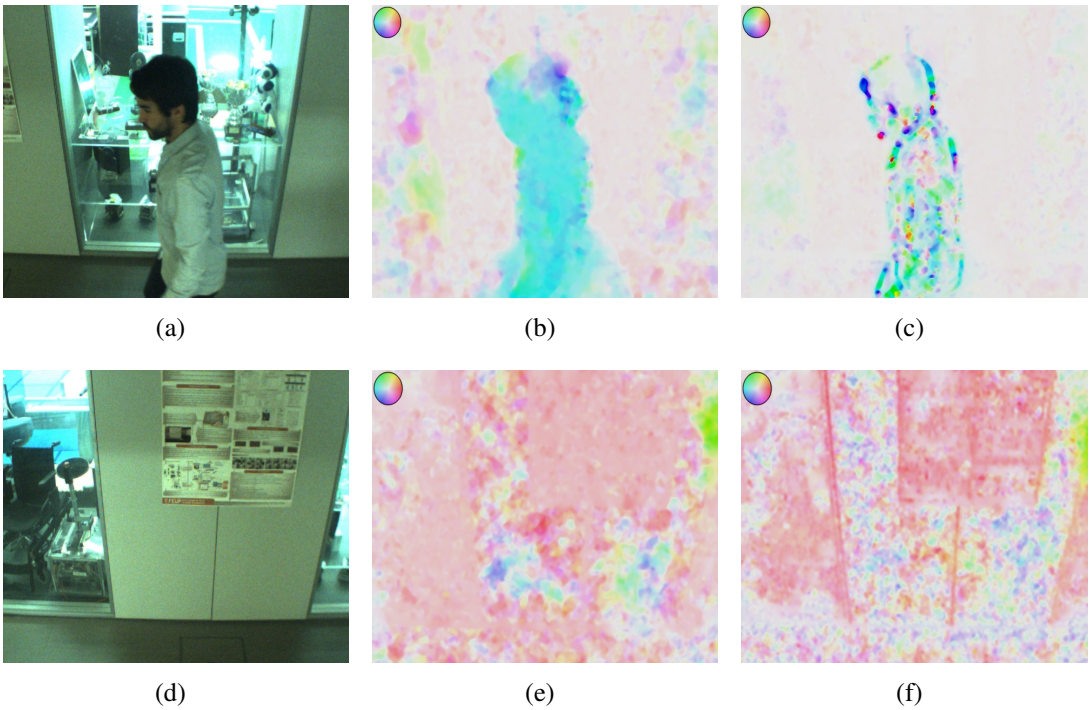
In addition to that, the HY computing is many times faster than the CLG. The CLG and HY achieve respectable performances. The HY was designed not to be a state-of-the-art method in terms of the flow estimation quality (low AAE and EPE), but to provide a reliable estimation with an acceptable computational complexity. Taking into account the results achieved by the HY technique, it is possible to conclude that incorporating high level information in optical flow estimations is a clear advantageous.

5.3.2.3 Motion perception using a robotic system

Finally, the HY technique was used in a real application. The robotic system with monocular vision and limited computer resources was considered in this experiment. The robot moves in a corridor with a constant velocity of $0.4m/s$ and the image resolution is 640×480, see Fig. 5.14(a). Images were captured by a DFK 21AU04 camera ("The Imaging-Source") with a 4mm focal lens.

The results of the HY under different strategies are presented in Figs. 5.14(a) to 5.14(f). Figures 5.14(b) and 5.14(e) show the result for an estimation based on the same combination of local and global strategies as presented in previous sections. Figures 5.14(c) and 5.14(f) are the results of the HY technique with a stronger local strategy (the neighborhood size is set to 3 pixels with four-connectivity and the global formulation is only used to classify the coarsest region). The *expectancy* operation took 23.3 milliseconds. The *sensing* operation estimated the optical flow in 148 milliseconds and 125 milliseconds, Figs. 5.14(b) and 5.14(e), respectively. In Fig. 5.14(c), the *expectancy* and the *sensing* operation took 14.2 and 82.8 milliseconds to compute and, the HybridTree method took 78 milliseconds to compute Fig. 5.14(f).

The egomotion of the robot is represented by the red and pink pixels, while the motion of the person is represented by the blue pixels. Images 5.14(b) and 5.14(e) have a strong noise component which affects the quality of the optical flow belonging to the egomotion. This happens because the corridor has large textureless regions without much velocity (the egomotion) and, therefore, the flow estimation can easily be misled by noise of the camera's sensor. As expected, Figs. 5.14(c) and 5.14(f) are not significantly influenced by



(a) (b) (c)

(d) (e) (f)

Figure 5.14: Comparison of the HY for different strategies (strong local and strong global). 5.14(a) - a person passes by the robot that moves in a different direction. 5.14(d) - the robot is moving alone. 5.14(b) and 5.14(e) are the optical flow field with the two finer classes being computed by the local formulation and the three coarser classes being computed by the global formulation. 5.14(c) and 5.14(f) represent the optical flow where four classes are computed by the local formulation and the coarsest class is computed by the global formulation.

noise because the configuration is more based on local scheme; however, some *filling-in effect* is lost. Therefore, according to the computational resources available, there should be a balance between optical flow performance and the efficiency.

5.4 The testing scenario: examples of dense flow fields

This section introduces the results of the HY method in a scenario where a mobile robotic system conducts active surveillance operations. The *EEyeRobot* is equipped with a fixed monocular camera and it moves along a rail. The rail was placed in a corridor at the Department of Electrical and Computer Engineering of the Faculty of Engineering of the University of Porto.

The main objective of this section is to provide a set of dense flow fields. These flow fields are used as reference for testing the techniques of motion analysis that are presented in chapter 6. Therefore, the performance of the HY method was evaluated for

several surveillance sequences and considering the multi-channel and single-channel approach. This makes it possible to compare the method for different formulations: in terms of visual quality and computational efficiency. All experiments were conducted using the *EEyeRobot* in a realistic surveillance scenario (similar to the previous subsection). Therefore, they depict real testing conditions which means that the visual system of the mobile robot is subjected to different light conditions, reflections, diffractions, shadows and ghost effects (due to grass walls) [20]. No filtering technique was previously applied to the sequences in order to maintain the reliability and repeatability of the experiments. The images have a resolution of 640×480 and were captured using a "The Imaging Source DFK 21AU04" camera with a 4mm focal lens. Examples of surveillance images and the respective flow fields can be seen in Figs. 5.15(a) to 5.17(l). These flow fields were obtained by the single and the multi-channel formulation of the HY method. The HSV color space is used to represent the direction (color) and magnitude (saturation) of the flow vectors.

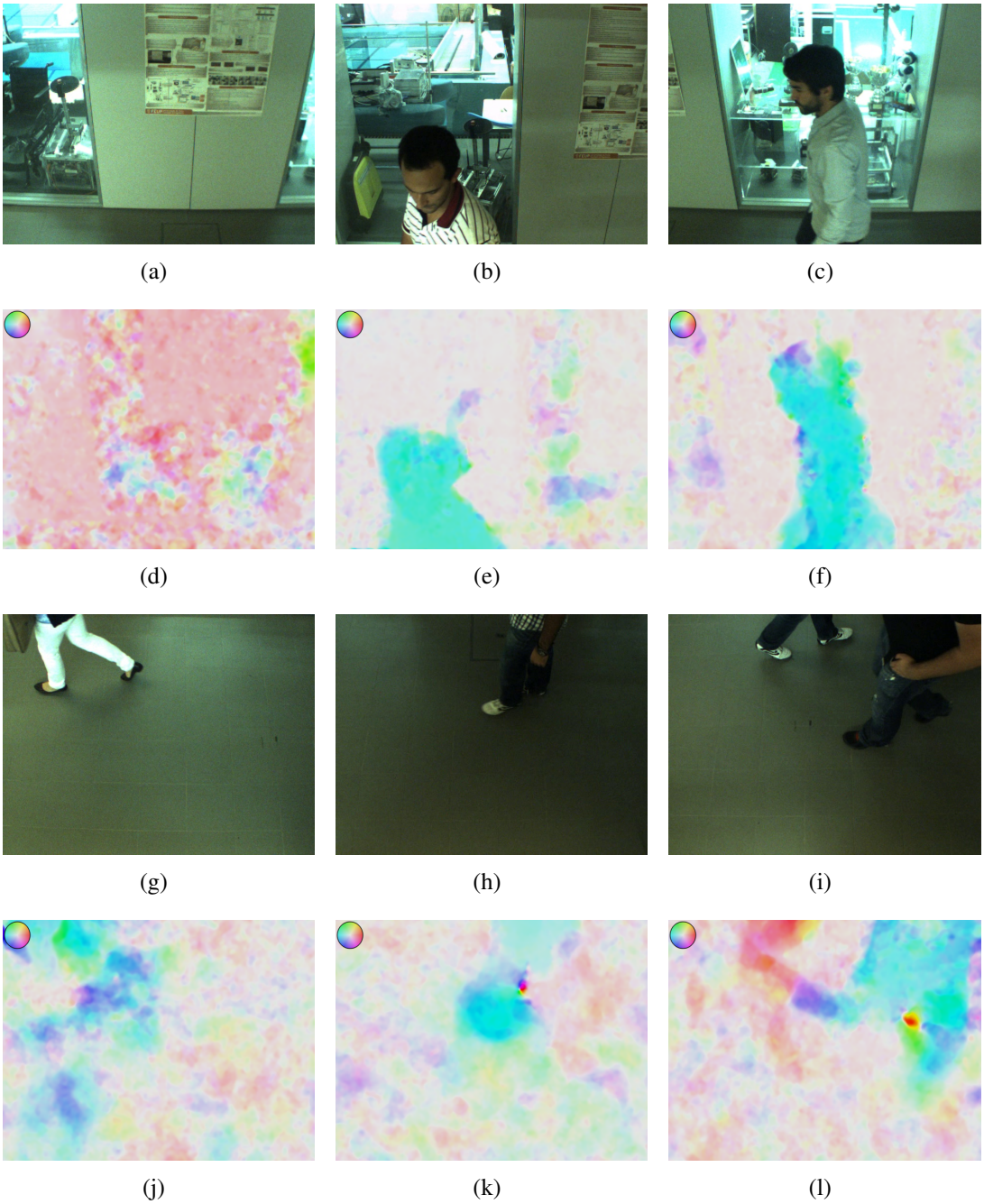
5.4.1 Estimation of the optical flow for a multi-channel formulation

Two different scenarios were tested during the trials: the robot views the faces or the bodies of the people. Figures 5.15(a), 5.15(b), 5.15(c), 5.15(g), 5.15(h) and 5.15(i) depict one image in the sequence that originates the respective flow field, Figs. 5.15(d), 5.15(e), 5.15(f), 5.15(j), 5.15(k) and 5.15(j). These flow fields were obtained by the multi-channel formulation of the HY method: three channels of each image sequence are considered during the estimation of the optical flow. In a flow field figure, the saturation (intensity) of the HSV color space represents the magnitude of each flow vector and it is obtained using the maximum magnitude of all vectors. Thus, one flow field should not be directly compared to another since its representation is relative. For instance, the flow field of Fig. 5.15(d) appears to have flow vectors with larger magnitude when compared to 5.15(e); however, this is not true because the last flow field contains a moving person characterized by a flow vector with higher magnitude, which reduces the representation scale of the remaining scene (red regions).

The estimation of these dense flow fields took on average 325 milliseconds to compute and using a multi-channel formulation with: 4 pyramidal levels with warping process.

5.4.2 Estimation of the optical flow for a single-channel formulation

Examples of surveillance images and the respective flow fields can be seen in Figs 5.16(a) to 5.17(l). The estimation of these dense flow fields took on average 120 milliseconds



(a) (b) (c)

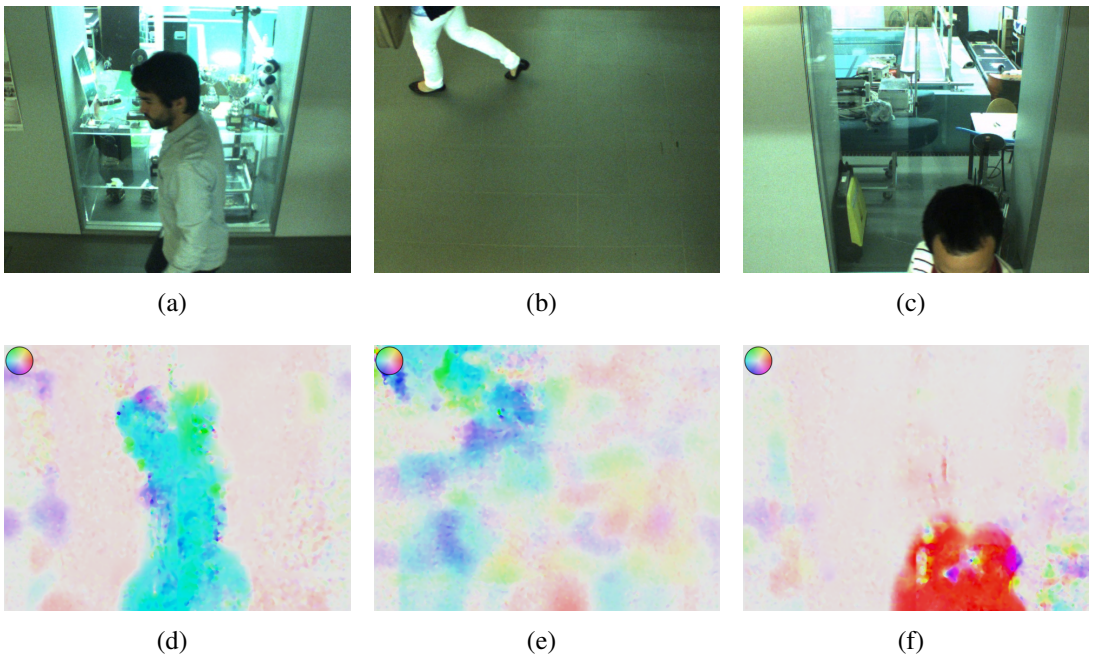
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 5.15: Multi-channel configuration - Examples of flow fields obtained from a dense optical flow technique with the *EEyeRobot* moving along the rails. One image of each sequence is presented in the first and third row. The corresponding flow field represented in the HSV color space (direction-color and magnitude-saturation) is presented in the second and fourth row. The caption is shown on the upper left side of the flow field images, Figs. 5.15(d), 5.15(e), 5.15(f), 5.15(j), 5.15(k) and 5.15(l).

to compute and using the single-channel formulation: 4 pyramidal levels with warping process.



(a) (b) (c)

(d) (e) (f)

Figure 5.16: Single-channel configuration - Examples of flow fields obtained from a dense optical flow technique [4] with the *EEyeRobot* moving along the rails. One image of each sequence is presented in the first row and the corresponding flow field is presented in the second row.

Visually, the results obtained from the single-channel approach are similar to the trials with a multi-channel formulation. As expected, the quality of the dense flow fields is lower when the HY method computes the optical flow using only the brightness of image sequences. Comparing Figs. 5.15(f) and 5.15(l) with Figs. 5.16(d) and 5.17(k) it is possible to confirm a reduction of the quality of the flow field since the presence of noise is evident in the last two flow fields. The noise is caused by several problems that have already been mentioned in this research, for instance, the aperture problem, reflections and the sensor noise. Although originating dense flow fields with a poor quality, the computation of the single-channel formulation is 2.7 times more computational efficient since it is usually conducted in 120 milliseconds. In this way, the HY method estimates the flow field in a short period of time by balancing the computational effort with the quality of the estimated optical flow.

The test sequences proposed in this section are used in further chapters of this thesis to compare different methods that were designed to interpret and analyze motion based on dense flow fields. Therefore, the HY method identifies motion properties about the sequence that are considered as high level information before estimating the optical flow.

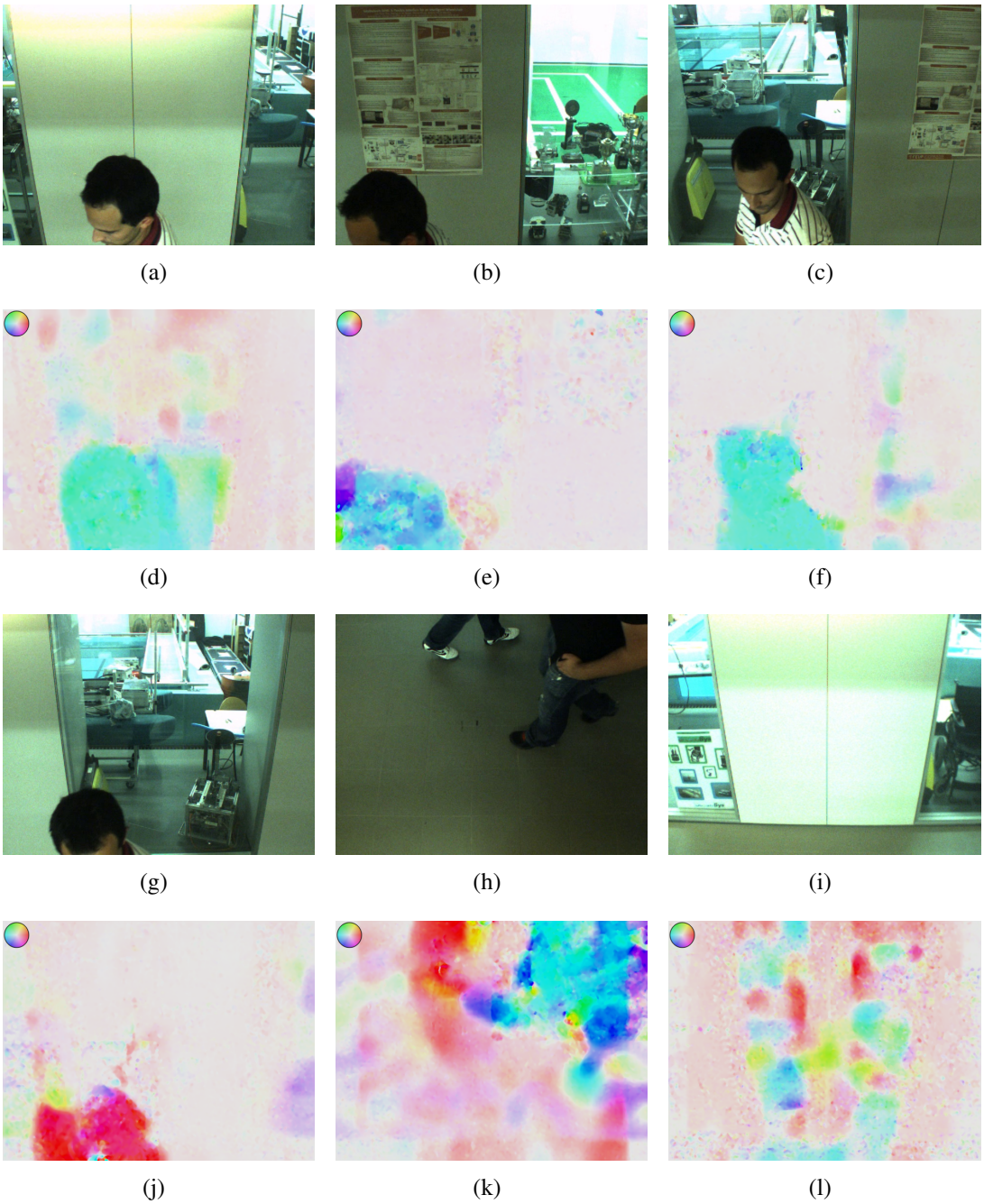


Figure 5.17: Single-channel configuration - Examples of flow fields obtained from a dense optical flow technique [4] with the *EEyeRobot* moving along the rails. One image of each sequence is presented in the first and third row. The corresponding flow field is presented in the second and fourth row.

After, local and global differential optical flow formulations are combined without resorting to non-quadratic penalizers. This affects the quality of the estimation but satisfies the real-time requirements of conventional robotic applications. The technique has a good computational performance; however, the presence of noise is evident in some flow fields. The noise is caused by several problems that are related to the sensor's noise, the aperture problem and the technique itself.

5.5 Final considerations

A novel and colored optical flow technique is proposed in this research which is called *HybridTree*. Unlike traditional methods, the proposed technique uses high level information on the image sequence to guide the optical flow estimation.

The *HybridTree* is formed by two operations, namely, *expectancy* and *sensing*. The technique interprets the sequence of images and identifies areas with distinct motion characteristics. This stage is called *expectancy*. This way, the information on the image is used to infer and assign the optical flow technique that best suits each image region. This stage is called *sensing*. A hybrid optical flow structure is used in the *sensing* operation. The hybrid approach blends in a symbiotic and hierarchical scheme the advantages of both local and global techniques, namely, the LK and the HS. The aim with this combination is to benefit from the exclusive advantages of each method, the robustness to noise and the *filling-in effect*, respectively. The goal of this architecture is to avoid non-quadratic penalizers in order to increase computational efficiency. In addition to guiding the *sensing* operation, the high level information provided by dividing the image into distinct regions enables to enhance the parameters of the optical flow technique assigned to each type of region.

The technique presented is designed not to be a state-of-the-art method in terms of the quality of the flow estimation, but to provide a reliable estimation with an acceptable computational complexity. However, some changes can be made in order to achieve more accurate estimations: "robustification" of the data and smoothness term as well as each color channel, the gradient constancy assumption and normalization of the data term, and photometric invariant color spaces.

The experiments conducted proved that incorporating high level information in the optical flow estimation is advantageous. Comparatively to another hybrid approach, the CLG method, the *HybridTree* achieves a better flow estimation and a higher computational efficiency. Therefore, the proposed optical flow technique meets the computational requirements of common robotic systems, as it can estimate the flow field in less than 150

milliseconds (for images with a resolution of 640×480), without specialized hardware or parallel programming. In short, a reliable and efficient method was developed as part of this study that makes it possible to perceive motion using limited computational applications, such as, robotic and surveillance systems. This method takes advantage of the most relevant and efficient improvements to create a balance between the real-time capability and the estimation performance.

Chapter 6

An Intelligent Segmentation of Dense Optical Flow Fields

The computational resources and the processing-time are two of the most critical aspects in motion analysis based on dense optical flow fields and for a new generation of robotic moving systems with real-time constraints. Therefore, this chapter proposes two non-parametric and block-wise techniques, namely, the Hybrid Hierarchical Optical Flow Segmentation (HHOFS) and the Hybrid Density-Based Optical Flow Segmentation (HDBOFS). Both methods are able to extract the moving objects by performing two consecutive operations: refining and collecting. During the refining phase, the flow field is decomposed in a set of clusters and based on descriptive motion properties. These properties are used in the collecting stage by a hierarchical or density-based scheme to merge the set of clusters that represent different motion models. The results obtained by both techniques have a blocky aspect and, therefore, this research proposes a motion analysis technique, called Wise Optical Flow Segmentation (WOFS). This method extracts all the moving objects at flow level and by performing two consecutive operations: evaluating and resetting. Descriptive motion properties of the flow field are retrieved in the evaluation phase and using the hybrid hierarchical optical flow segmentation, which provides high level information on the spatial segmentation of the flow field. In the resetting operation, these properties are used by a watershed-based approach to enhance the resulting clusters.

Moreover, the chapter presents a novel method to extract information about the number of moving objects using a polar representation of the dense optical flow fields. The model selection method is a Bayesian approach that balances the model's fitness and complexity since it combines the correlation of a histogram-based analysis with the decay ratio of the normalized entropy criterion.

This research evaluates the performance achieved by the proposed methods in a real and concrete surveillance situation. Therefore, the experiments conducted in a realistic environment and using qualitative/quantitative judgments have proved that the HHOFS and the WOFS are able to segment multiple moving objects in a short period of time and without using specialized computers. Hence, the proposed motion analysis techniques meet most of the robotic or surveillance requirements since they are less computationally demanding comparatively to other state-of-the-art methods.

The chapter[1] is organized as follows. An overall presentation of the achievements obtained by this chapter is introduced in section 6.1 while a model selection method is proposed in section 6.2. Section 6.3 shows the unsupervised clustering techniques: an overview of the EM and K-means is provided in sections 6.3.1 and 6.3.2, and two non-parametric techniques are described with detail in sections 6.3.3 and 6.3.4. Afterwards, section 6.4 presents a technique (WOFS) that segments motion of dense flow fields at flow level. Experimental achievements are presented in section 6.5. These experiments include the comparisons of the HHOFS, HDBOFS and WOFS with the EM and the K-means. The experiments were conducted using the *EEyeRobot* in a real surveillance scenario; and results demonstrate that the proposed techniques perform satisfactorily better than the baseline methods, and can be used as a tool for motion analysis in applications with limited resources. Finally, section 6.6 presents the most important conclusions of this chapter.

6.1 Introduction

The research work presented in this chapter studies the real-time motion analysis using dense optical flow fields and for a practical use in a mobile robot. In the scientific community, motion perception is one of the most relevant areas under discussion, existing several models to perform motion analysis in a variety of environments. However, most of the methods cannot achieve the real-time constraints imposed by mobile robots without specialized computers (discussed in chapter 2). In some cases, these computer devices cannot be used due to the small size of the vehicles or because they cause a higher consumption of energy which reduces the autonomy of such robots. Nowadays, there are pixel-wise techniques that have good results [114]; however, segmenting motion commonly takes more than a pair of seconds. Visual techniques for robotic solutions are computationally more efficient than techniques for other application fields, although the improvement is

[1] Some portions of this chapter appeared in [22].

usually done at the expense of using images with lower resolution and feature-based approaches. Motion segmentation is the process of dividing an image into different regions in a way that each region presents homogeneous motion characteristics. Therefore, the goal of this work is to segment different objects according to their motion coherence.

In this chapter, the estimation of dense flow fields is conducted by the *HybridTree* technique [4], that was especially designed for small robotic applications equipped with generic computers. This optical flow technique identifies motion properties which are considered as high level information about the sequence and originates flow fields in a short period of time. The computational resources and the processing-time are some of the most critical aspects for vision-based techniques applied to robotics. Usually, these applications tolerate some loss of accuracy in the algorithms to ensure a fast response [175].

This thesis proposes two block-wise techniques for unsupervised segmentation of dense flow fields: the Hybrid Hierarchical Optical Flow Segmentation (HHOFS) and the Hybrid Density-Based Optical Flow Segmentation (HDBOFS). These two techniques were designed for robotic applications with a vision system and limited computer resources. Two major and distinct phases form both methods, namely, *refining* and *collecting*. The *refining* stage decomposes the flow field in a set of distinctive clusters that represent image regions with different motion models and the *collecting* stage merges the set of clusters that were obtained in the previous phase (using a hierarchical scheme or a density-based scheme). This architecture reduces the computational requirements of the proposed methods (HHOFS and HDBOFS). The performance of the HHOFS and HDBOFS is compared to two baseline methods, called, K-means and Expectation-Maximization (EM). An extensive and interesting comparison between the parametric (K-means and EM) and the proposed non-parametric techniques (no assumptions about the distribution of the data) is discussed.

Because the parametric techniques require information about the number of clusters in the data, this chapter proposes a model selection algorithm that combines, using a Bayesian approach, the correlation of histogram-based analysis in the polar representation space with the decay ratio of the normalized entropy criterion. This method balances the model's fitness and complexity, providing a reliable estimation about the number of motion models described in dense flow fields.

Motion segmentation in surveillance operations can be appropriately performed without processing at flow level; however, this chapter presents a pixel-wise technique for segmenting dense flow fields, called Wise Optical Flow Segmentation (WOFS). This technique can be applied for situations where the blocky results of the HHOFS are inadequate. The WOFS meets the visual requirements of a surveillance system based on the mobile robot that was briefly presented in chapter 3. The technique has two major and

distinct phases: *evaluating* and *resetting*. The *evaluation* phase uses the polar representation (magnitude and phase) to identify regions of the flow field that retain different motion models. Basically, this phase is the HHOFS because it provides a set of advantages, for instance, is very intuitive to setup, is less affected by the visual artifacts of the environment and is computationally efficient (it takes 30 milliseconds to compute a 640×480 flow field). Descriptive motion properties obtained by evaluating dense flow fields are interpreted during the *resetting* phase as high level information of the moving objects. Thus, the main objective of the *evaluation* stage is to guide the pixel-wise segmentation. This research proposes the watershed algorithm for the *resetting* phase to avoid the appearance of spatially decorrelated blobs (noisy blobs) that belong to the same cluster. In a surveillance context, the result of a clustering procedure with several small blobs that represent the same moving object is not desirable. Therefore, information that is obtained in the *evaluation* phase can be used to initialize the watershed, which enhances the shape of moving objects in the final result. The watershed is an efficient technique that can be easily guided using the evaluation of the flow field, and it does not compromise the real-time requirements of mobile robotic systems.

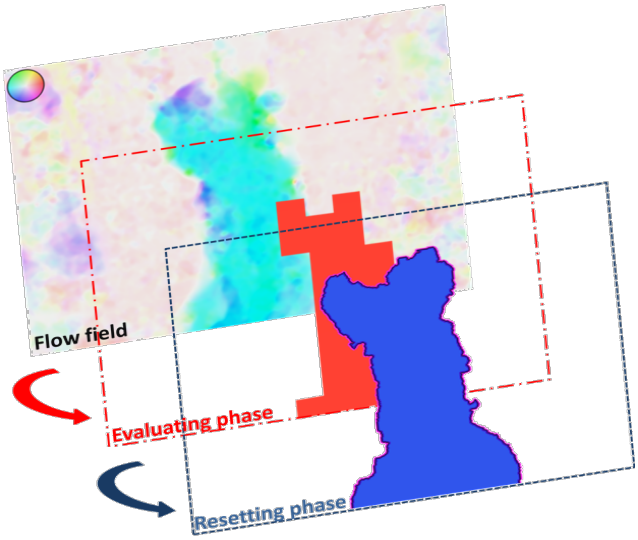


Figure 6.1: The two phases of the WOFS technique: *evaluating* and *resetting*.

The major advantage of using high level information for motion segmentation based on the spatial location of different moving objects is the ability to use an efficient technique to segment dense flow fields that are affected by noise (sensor noise and the visual artifacts of the environment) with the desirable robustness. Furthermore, the WOFS technique combines the advantages of a block-wise method (efficiency and robustness) and the watershed approach (pixel-wise segmentation and spatial correlation of clusters), which

makes it possible to achieve performances in terms of visual quality and computational efficiency that otherwise would hardly be obtained.

Overall, the contributions of this chapter include:

1. A study about an efficient motion analysis based on dense optical flow fields and for moving observers;

2. Novel motion analysis methods characterized by a reduced computational complexity: the HHOFS and the HDBOFS. The proposed architecture of Fig. 6.5 performs motion analysis for both methods, enabling a reliable segmentation while preserving the computational time requirements;

3. An efficient metric to decompose the optical flow field into exclusive regions based on similarity properties of motion;

4. An assisted technique for clustering dense flow fields that automatically extracts and combines cognitive information about distinct motions. This guided-based technique (WOFS) enhances the edges of moving objects (contours) and preserves the computational time requirements of robotics applications;

5. An efficient Bayesian model selection approach, called Bayesian Fusion of Histogram and Entropy (BFHE). This method provides information regarding the number of distinct moving objects present in dense flow field by combining the correlation of the phase and magnitude histogram analysis with the decay ratio of the normalized entropy criterion;

6. Extensive qualitative and quantitative evaluations of the proposed techniques and considering several baseline pixel-wise clustering techniques, namely, the K-means and the Expectation-Maximization;

7. A comparative study of several motion analysis methods under realistic working conditions (with moving observers).

Experimental considerations prove that modeling a motion analysis technique in a structure formed by two consecutive stages is computationally rewarding and represents an alternative to state-of-the-art techniques based on EM and K-means. The computational demands of the HHOFS and the HDBOFS are substantially lower than that of the EM and K-means that are conducted at flow level. The behavior of the proposed techniques can be adjusted according to specific characteristics of the application. For instance, motion segmentation in surveillance operations can be appropriately performed

without processing at flow level. Moreover, experimental considerations prove that combining high level information in a pixel-wise procedure is computationally rewarding and represents an alternative to other techniques [114, 112]. Therefore, the proposed WOFS is completely capable of perceiving and understanding different external motions using low computational resources.

6.2 Model selection

Dense flow fields can represent an unknown number of moving objects (K) that should be estimated previously to produce a more reliable segmentation. The number of clusters is a necessary parameter in some techniques, for instance, EM and K-means. The techniques that are proposed in this research do not need this parameter; however, the knowledge of this parameter is a clear advantage because makes it possible to guide and conclude the clustering process when the desired number of clusters is reached. The estimation of K is one of the most important problems for unsupervised machine learning and a huge effort related to this issue is being made by the scientific community. This research proposes a method to estimate the value of K, see Fig. 6.2. This model selection method is called Bayesian Fusion of Histogram and Entropy (BFHE) and combines the histogram analysis of the flow field in Polar coordinates with the highest decay ratio of the normalized entropy criterion. The hypotheses are combined using the naïve Bayesian formulation which generates a reliable estimation of K.

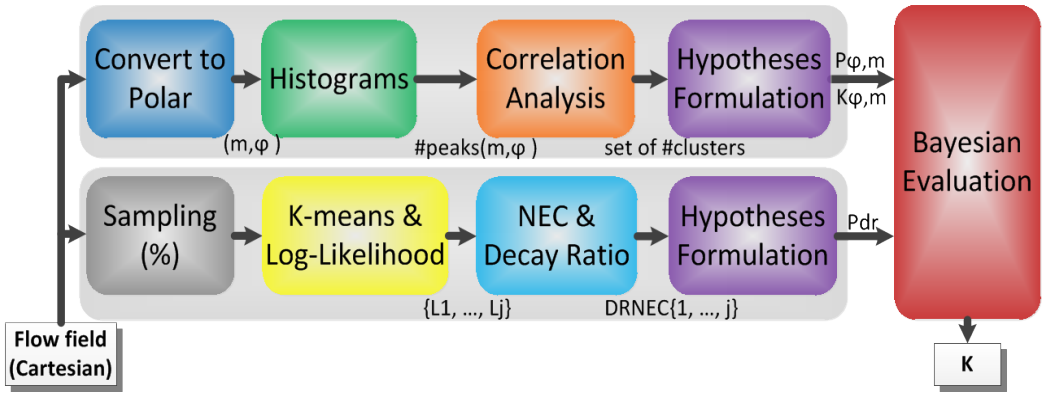


Figure 6.2: Detailed structure of the model selection method. It combines the histogram-based approach with the decay ratio of the normalized entropy criterion (NEC).

6.2.1 Feature space

Finding the most suitable feature space is one of the most important issues when segmenting data. A feature space is suitable when it is possible to identify more easily the number of clusters represented in the data. This chapter uses the representation of the flow vectors in the Polar space (magnitude and phase) due to several advantages that are briefly discussed. A flow vector is defined by $\mathbf{w} = (u, v)$ in Cartesian coordinates and $\mathbf{w}^p = (m, \psi)$ in Polar coordinates, where u and v are the horizontal and vertical flow velocity; and m and ψ are the magnitude and angle of the flow vector, respectively. Therefore, a single observation is $\mathbf{x} = (\mathbf{w}, \mathbf{w}^p, x, y)$, where (x, y) is the coordinate position.

Figures 6.3(a), 6.3(b), 6.3(c) and 6.3(d) show a two-dimensional histogram representation of the 5.16(f) and 5.17(k) flow fields (with two and three clusters). The histograms of these flow fields are represented in the Cartesian space, Figs. 6.3(a) and 6.3(b), while the histograms in the Polar space are represented in Figs. 6.3(c) and 6.3(d).

The histograms in Figs. 6.3(a) and 6.3(b) are bimodal; however, one modal is high and the other is very small. The detection of small modals is affected by the size of the bins and the noise of the data. Hence, the analysis of histograms in Cartesian coordinates is not very robust since Figs. 6.3(a) and 6.3(b) return a modal with high confidence and another with low confidence (two possible clusters). In addition to the bimodal behavior of the second histogram, the corresponding flow field is actually formed by three clusters and, thus, the number of clusters presents many uncertainties. On the contrary, the Polar representation of the same flow field originates histograms that reveal characteristics of distinct motions with more confidence. Figure 6.3(d) shows three main peaks at $\{(0,0),(10,0),(23,-\pi)\}$ and Fig. 6.3(c) shows two peaks (same direction but with different magnitude). In the case depicted by Fig. 6.3(d), these three clusters represent the components of the flow field that are associated with the egomotion of the robot and the movement of two external objects in the opposite direction.

Thus, the Polar representation makes the data more distributed over space which facilitates the analysis since the data is more modal. In more detail, Figs. 6.4(a) to 6.4(d) show one-dimensional histograms of the flow field 5.16(f) represented in Cartesian and Polar space. Figure 6.4(a) is bimodal: one modal is high and the other is very small. The detection of small modals will depend on the size of the bin and the noise of the data. Hence, the analysis of this histogram is not robust since it returns a modal with high confidence and the other with low confidence (two possible clusters). Figure 6.4(b) is an unimodal histogram, and thus the information obtained by the analysis of both histograms presents many uncertainties. The flow field represented in Polar coordinates originates histograms that reveal the characteristics of the dataset with more confidence. Figure 6.4(c) shows

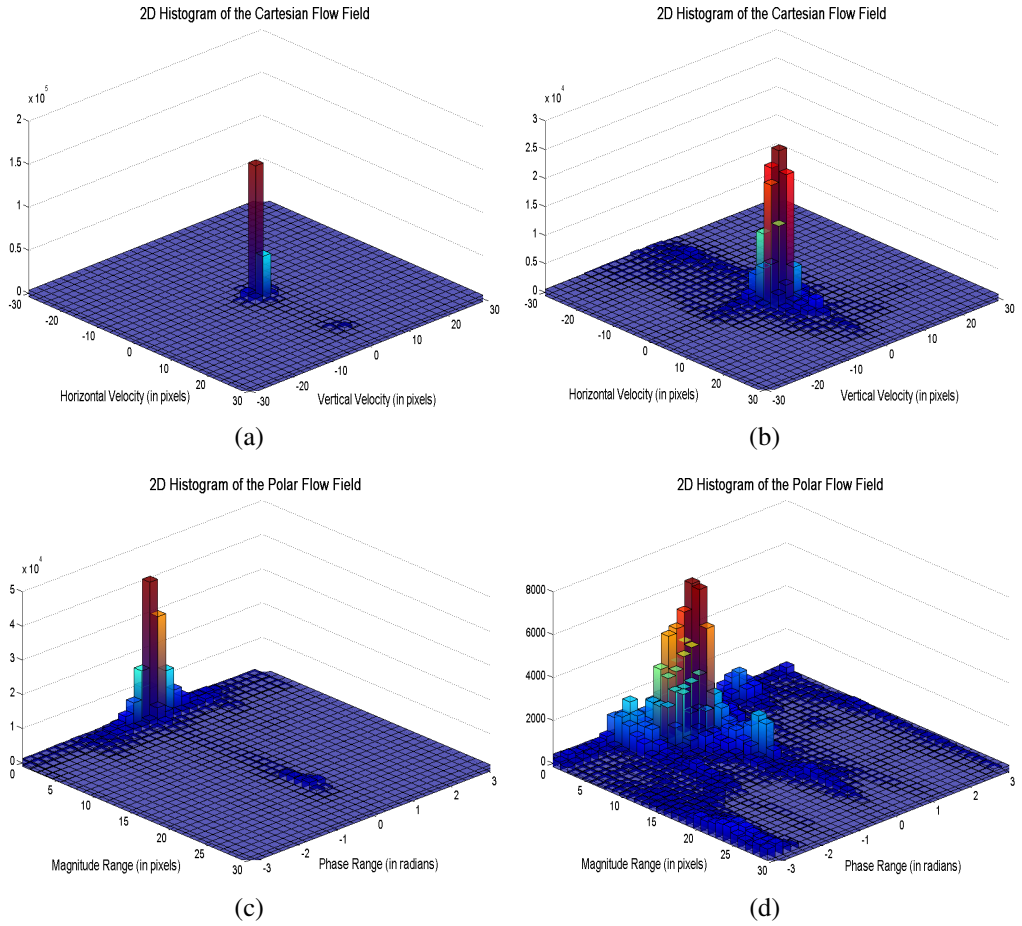


Figure 6.3: Two-dimensional histograms. 6.3(a) and 6.3(c) represent the distribution of the flow field 5.16(f) in the Cartesian and Polar coordinates, respectively. 6.3(b) and 6.3(d) represent the distribution of the flow field 5.17(k) in the Cartesian and Polar coordinates, respectively.

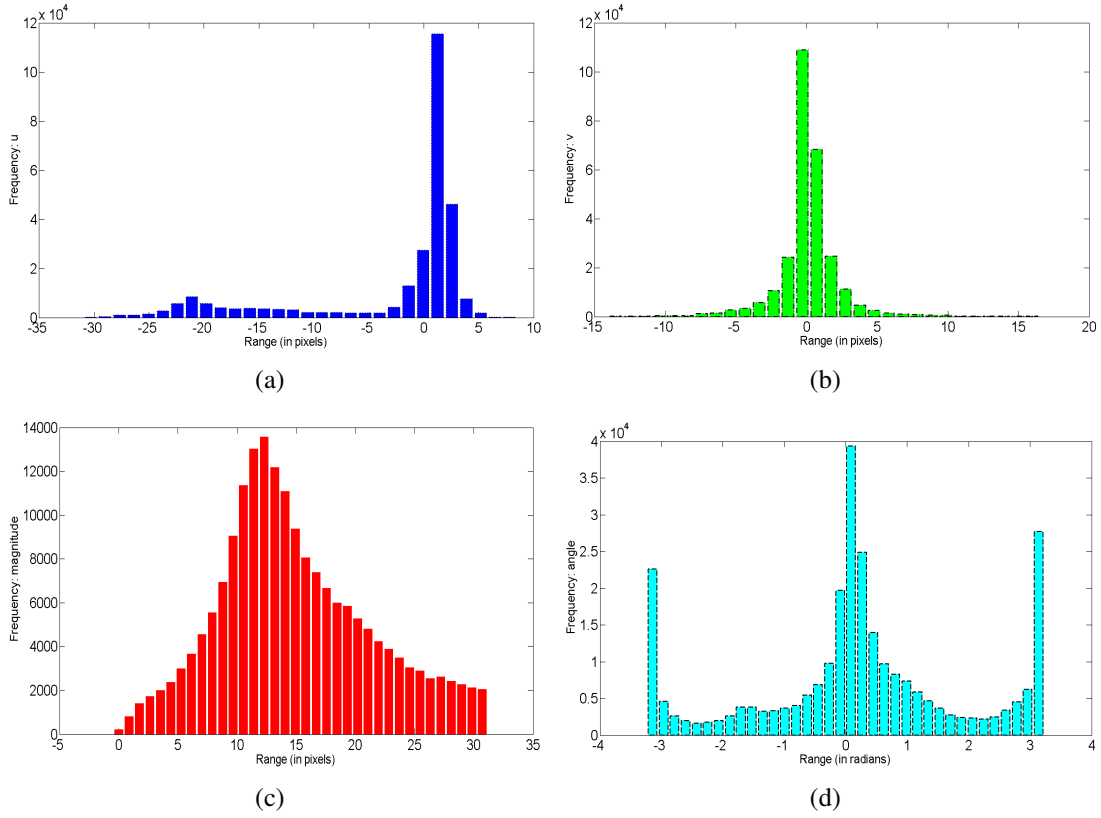


Figure 6.4: One-dimensional histograms. The flow field 5.16(f) depicts the motion of the robot and one external object moving in the other direction. 6.4(a) and 6.4(b) represent the distribution of the horizontal and vertical velocity. 6.4(c) and 6.4(d) represent the distribution of the magnitude and angle (in radians) of the flow vectors.

the histogram of the magnitude for the same flow field. As it can be confirmed, the histogram is unimodal; however, the histogram of the angle is very revealing. It shows 3 peaks at $\{-\pi, 0, \pi\}$; however, the peaks $\{-\pi\}$ and $\{\pi\}$ are of the same cluster because their normalized difference is zero. After adding the peaks with small differences, the histogram of the angle gives two clusters with high confidence. The clusters represent the components of the flow field that are associated with the egomotion of the robot and the movement of external objects.

6.2.2 Correlated histogram-based analysis

In this research, the histogram analysis focuses on the multimodal behavior of the dataset and the interpretation of 1D histograms (magnitude and phase) makes it possible to evaluate the density of movement more efficiently, see Figs. 6.4(c) and 6.4(d). However, the number of clusters cannot be directly obtained with a strong confidence and from the sum of the number of peaks since the peaks of magnitude and phase can be correlated. This means that, some peaks of the magnitude-histogram may be correlated with peaks of the phase-histogram, resulting in a different number of clusters. Therefore, this research measures the correlation between the set of peaks that is retrieved from the analysis of 1D histograms. A consensus decision making process measures this correlation, which means that the magnitude-field is evaluated by taking into consideration the bin range of each phase-peak. After the consensus process is completed, the correlation is detected if there is a significant number of phase-elements belonging to a phase-peak that also represents a magnitude-peak. This is similar to analyzing a two-dimensional histogram of magnitude and phase, but with less data dispersion. The total number of votes of phase-peaks that originate the maximum correlation for a magnitude-peak is retrieved at the end of the consensus process. The representativeness is computed afterwards by defining a ratio between the number of votes of the phase-peak and the total number of votes of each magnitude-peak. If this ratio is higher than a given threshold (0.6 is usually a good value), it means that there is correlation between a phase-peak and a magnitude-peak.

A set of hypotheses representing the number of clusters (histogram-based hypothesis) is estimated by considering not only the number of non-correlated peaks of both features but also the multi-correlations. In addition, the probability of each hypothesis is computed by combining the normalized confidence of the most important clusters. Experimental results show that the number of clusters can usually be estimated in this histogram-based approach by combining clusters until more than 68.2 % of the data (assuming a normal distribution) is represented in the hypothesis; however, the performance is strongly affected by noisy dense flow fields. Therefore, the resulting information of

the histogram-based approach is: several hypotheses for the number of clusters and the respective probability, $K_{(m,\psi)}$ and $P_{(m,\psi)}$.

6.2.3 Decay ratio of the normalized entropy criterion

In addition to the histogram analysis, various criteria have been proposed to measure a model's suitability by balancing the model complexity and the model fitness to data, for instance, the AIC (Akaike Information Criterion) [176], the BIC (Bayesian Information Criterion) [177], the HQIC (Hannan-Quinn Criterion) [178] and the NEC (Normalized Entropy Criterion) [179]. These techniques show that increasing the number of free parameters in the model improves the fit and, therefore, they penalize the model complexity which discourages the overfitting.

The K-means algorithm is executed for $K \in \{1,...,K_{max}\}$ and using the flow field in the Euclidean Space. After that, the log-likelihood of the dataset is computed, which makes it possible to formulate the normalized entropy criterion [179]. The normalized entropy criterion (NEC) was proposed by *Celeux and Soromenho (1996)* [179] and measures the ability of a Gaussian Mixture Model (GMM) to provide well-separated clusters. The entropy term provides the overlapped components and, therefore, $E(K) \approxeq 0$ if the GMM is well-separated; otherwise, the entropy value is large. Considering the entropy (E) and the classification log-likelihood (CML) :

$$E(K) = \sum_{j=1}^{K}\sum_{i=1}^{M} t_{ij} \ln t_{ij} \geq 0; \tag{6.1}$$

$$CML(K) = \sum_{j=1}^{K}\sum_{i=1}^{M} t_{ij} \ln \pi_j N(\mathbf{x}_i|\hat{\theta}_j); \tag{6.2}$$

$$t_{ij} = \frac{\pi_j N(\mathbf{x}_i|\hat{\theta}_j)}{\sum_{l=1}^{K} \pi_l N(\mathbf{x}_i|\hat{\theta}_l)}, \tag{6.3}$$

where $N(\mathbf{x}_i|\hat{\theta}_j)$ is the normal distribution of the feature vector $\mathbf{x}_i \in \mathbb{R}^n$ and considering the model $\hat{\theta}_j$. The t_{ij} denotes the conditional probability that $\mathbf{x}_i$ arises from the jth mixture component.

Considering $L_K = CML(K) + E(K)$, where $L_K = \ln L(X|\hat{\theta})$ is the maximized log-likelihood of the sample $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$. The normalized entropy criterion is given by:

$$NEC(K) = \frac{E(K)}{L_K - L_1}. \tag{6.4}$$

Equation 6.4 shows that the $NEC(1)$ is not defined. The research in [179] proposes an alternative way of estimating $NEC(1)$ by constraining the mixing normalized weights to avoid degenerate solutions concerning the choice between the K, which minimizes the NEC criterion (for $2 \leq K \leq K_{max}$[2] or $K = 1$).

In the literature, the model selection based on the NEC is the model that originates the lowest NEC value. The original concept of NEC is selecting the model from a set of candidates by maximizing the subsequent probability and using the maximum likelihood approach [179]. However, this research does not use the NEC criteria directly to assess the model since the returning model is overestimated in our situation (see the results section). Instead, the higher and the lower NEC value is retrieved from the set of models under evaluation, NEC_{max} and NEC_{min}. The total variation of NEC is computed, $\Delta NEC = NEC_{max} - NEC_{min}$ and the decay ratio of the NEC (DR_{NEC}) is calculated using Eq. 6.5.

$$DR_{NEC}(K) = 1 - \frac{NEC(K)}{\Delta NEC}. \tag{6.5}$$

The probability of each hypothesis is obtained by normalizing the DR_{NEC} according to Eq. 6.6 and considering the model with a lower decay ratio (DR^{min}_{NEC}):

$$P_{dr}(K) = \frac{DR_{NEC}(K) + |DR^{min}_{NEC}|}{\sum_{j=1}^{K_{max}} |DR_{NEC}(j)| + |DR^{min}_{NEC}|}. \tag{6.6}$$

Therefore, the hypothesis with higher confidence is the model with the highest decay ratio of the normalized entropy criterion. This avoids a hypothesis that overestimates the value of K.

The final estimation of parameter K is calculated by combining the hypotheses obtained from the feature (magnitude and angle) analysis and cost function. This combination is achieved using the following naïve Bayes's formulation, Eq. 6.7.

$$P(h|r_{(m,\phi)}, r_{dr}) = \frac{P(h)P(r_{(m,\phi)}, r_{dr}|h)}{P(r_{(m,\phi)}, r_{dr})}; \tag{6.7}$$

$$P(h = h_k|r_{(m,\phi)}, r_{dr}) = \frac{P(h = h_k)P(r_{(m,\phi)}, r_{dr}|h = h_k)}{\sum_{l=1}^{h_{max}} P(h = h_l)P(r_{(m,\phi)}, r_{dr}|h = h_l)}, \tag{6.8}$$

where h_{max} is the maximum number of hypotheses, $r_{(m,\phi)}$ and r_{dr} are the observed feature analysis and the decay ratio, respectively. The denominator does not depend on the hypothesis since it is the normalization factor that keeps the probability in the range $(0, 1)$. Equation 6.8 shows the posterior probability $\propto$ prior probability $\times$ likelihood. Therefore,

[2]Without additional information, the K_{sup} should vary up to the integer larger than $M^{0.3}$ [180].

maximizing the posterior probability $P(h = h_k | r_{(m,\phi)}, r_{dr})$ is equivalent to maximizing the likelihood $P(r_{(m,\phi)}, r_{dr} | h = h_k)$.

This research assumes that all hypotheses share the same prior probability and assumes a conditional independence between $r_{(m,\phi)}$ and r_{dr}, the likelihood of Eq. 6.8 can be rewritten as:

$$P(r_{(m,\phi)}, r_{dr} | h = h_k) = \prod_{i=(m,\phi)}^{dr} P(r_i | h = h_k). \tag{6.9}$$

Thus, the formulation that makes it possible to infer about the hypotheses (values of K) by considering the combination of probabilities is given in Eq. 6.10:

$$h^{new} \leftarrow arg\ max_{h_k}\ P(h = h_k) \prod_{i=(m,\phi)}^{dr} P(r_i | h = h_k). \tag{6.10}$$

The value of K is the hypothesis that maximizes the likelihood, Eq. 6.9. This formulation has an additional advantage of computing the relative confidence ratio of the log-likelihood of Eq. 6.8, which provides a measurement of the distance between different hypotheses. Therefore, the method depicted in Fig. 6.2 receives the optical flow field and analyzes the characteristics of the flow using the features that mostly represent different motion models and combines the hypotheses with conventional model selection techniques that measure fitness and model complexity. As it can be confirmed further ahead in this chapter, this method produces an estimation of K that is substantially more reliable than BIC, AIC, HQIC and NEC criteria. Moreover, the importance of the BFHE is related to the fact that the *EEyeRobot* operates autonomously in the environment and, therefore, the number of clusters enhances the segmentation of the different types of motion which allows a better understanding of external motion in realistic environments.

6.3 Unsupervised segmentation

The goal of clustering techniques is to group a collection of instances into subsets of clusters: similar instances (more closely related) are clustered together and different instances belong to different groups. It performs an efficient representation of the instances that characterize the population. An important notion is the similarity or dissimilarity between the individual objects being clustered. Two main types of measurements are used to estimate this relation: distance measurements (Euclidean, Minkowski) and similarity measurements (Cosine, Pearson Correlation, Dice Coefficient, Extended Jaccard) [181].

Many clustering techniques have been proposed over the last two decades. Usually, the clustering methods are divided into two classes [182], hierarchical and partitioning methods. The hierarchical techniques create the clusters by merging the observations using pairwise similarity measurements and the partitioning methods require the number of clusters to iteratively relocate instances by moving them from one cluster to another. The segmentation of the flow field groups the pixels with the same motion properties because they probably belong to the same motion model. In this research, the performance of several techniques is analyzed. Two unsupervised clustering techniques are proposed in this section: a hybrid hierarchical method (HHOFS) and a hybrid density-based method (HDBOFS). The performance of both techniques is compared to the baseline parametric methods: K-means and EM.

6.3.1 Expectation-Maximization

The Gaussian mixture model (GMM) is a parametric probability function [183] which assumes that the data belong to a probability distribution formed by a convex combination of a linear superposition of Gaussian distributions, see Eq. 6.12. Each n-dimensional Gaussian density function (Eq. 6.11) is called component and it is characterized by the mean μ_i and the variance Σ_i.

$$N(\mathbf{x}|\mu_i,\Sigma_i) = \frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right). \tag{6.11}$$

The input dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ with $\mathbf{x}_i \in \mathbb{R}^n$ is assumed to be sampled from a set of K source of distributions, meaning that the goal is to model the input data using a mixture of Gaussians. Thus, the probability distribution of the GMM can be written as:

$$p(\mathbf{x}) = \sum_{j=1}^{K} \pi_j N(\mathbf{x}|\mu_j,\Sigma_j), \tag{6.12}$$

where $\mathbf{x}_i$ is the n-dimensional feature vector and the parameters $\pi_i > 0$ are called mixing normalized weights that meet the constraint $\sum_{j=1}^{K} \pi_j = 1$. As it can be confirmed, the Gaussian mixture model is completely parameterized by the mixture weights, mean vectors and covariance matrices from all component densities. Hereafter, they are referred to as parameter model of the Eq. 6.12 and represented by $\hat{\theta} = \{\pi_j, \mu_j, \Sigma_j\}$ with $j = 1, ..., K$.

The likelihood of the dataset, assuming independence between the features in the GMM, can be written as:

$$L(X|\hat{\theta}) = \sum_{i=1}^{M} p(\mathbf{x}_i). \tag{6.13}$$

There are several techniques available for estimating $\hat{\theta}$; however, the most well-established method is the maximum likelihood estimation [183]. The EM is an efficient method [184] to maximize Eq. 6.12, by estimating maximum likelihood solutions (a set of statistical parameters) for the given dataset using an iterative scheme based on two consecutive procedures (E-step and M-step) until convergence. A brief review of this technique is presented below; however, more detail can be found in [170].

From Eqs. 6.12 and 6.13, the log-likelihood function is given by:

$$\ln L(X|\hat{\theta}) = \sum_{i=1}^{M} \ln \left[\sum_{j=1}^{K} \pi_j N(\mathbf{x}_i|\mu_j, \Sigma_j) \right]. \tag{6.14}$$

where $L_K = \ln L(X|\hat{\theta})$ is the maximum log-likelihood of the samples X and considering the model $\hat{\theta}$.

The EM algorithm starts with an initial model $\hat{\theta}$ and then, fits the data into the K Gaussian distributions by expecting the classes of all the features. Then, it maximizes the likelihood relative to the Gaussian centers.

From the Bayes' theorem, the responsibilities (or posterior probability) can be defined by r_{ij}, see Eq. 6.15.

$$r_{ij} = \frac{\pi_j N(\mathbf{x}_i|\mu_j, \Sigma_j)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}_i|\mu_j, \Sigma_j)}. \tag{6.15}$$

Setting the derivatives of Eq. 6.14 individually with regard to μ_j, Σ_j and π_j equal to zero, it is possible to determine the model parameters $\hat{\theta}$ for the GMM, Eqs. 6.16, 6.17 and 6.18.

$$\mu_j = \frac{1}{M_j} \sum_{i=1}^{M} r_{ij} \mathbf{x}_i; \tag{6.16}$$

$$\Sigma_j = \frac{1}{M_j} r_{ij} \left(\mathbf{x}_i - \mu_j\right) \left(\mathbf{x}_i - \mu_j\right)^T; \tag{6.17}$$

$$\pi_j = \frac{M_j}{M}, \tag{6.18}$$

where $M_j = \sum_{i=1}^{M} r_{ij}$ represents the effective number of points assigned to class j [170]. The formulas above guarantee a monotonic increase in the likelihood [170] during the iterative estimation of the model parameters. Thus, the EM algorithm consists of the expectation step (E-step) that resorts to the current estimate of $\hat{\theta}$ in order to compute the posterior probabilities and the maximization step (M-step) that re-estimates the model's parameters. This algorithm can be seen in the following list:

1. Initialize the model's parameters, $\hat{\theta}$, and evaluate the initial value of the log likelihood;

2. E-step: compute the responsibilities using the current parameters and Eq. 6.15;

3. M-step: re-estimate the model's parameters considering the current responsibilities and using Eqs. 6.16, 6.17 and 6.18;

4. Re-compute the log likelihood for the new parameters and repeat steps 2 and 3 until convergence.

The major advantage of the EM method is that it provides extremely useful results for the real dataset; however, it requires the number of cluster to be specified a priori and the process is highly complex in nature since it is an iterative scheme that computes the posterior probabilities and the log-likelihood.

6.3.2 K-means

The K-means is simpler than the EM; however, it is also a powerful technique to cluster the input dataset in a multidimensional Euclidean space [170]. It is an iterative refinement technique that assigns the feature points to clusters by minimizing the square distance between each data point and its closest vector μ_j with $j = 1, ..., K$, which is a n-dimensional vector that describes the center of the jth cluster. To assign each feature vector $\mathbf{x}_i$, a set of binary variables must be introduced, $s_{ij} \in \{0, 1\}$. These variables represent the cluster (one of the K clusters) that the feature vector is assigned to. In this context, if $s_{ij} = 1$ the $\mathbf{x}_i$ belongs to the j cluster.

$$J = \sum_{i=1}^{M} \sum_{j=1}^{K} s_{ij} \|\mathbf{x}_i - \mu_j\|^2. \tag{6.19}$$

Equation 6.19 defines an objective function based on the sum of squares distances of each feature to its assigned cluster center. This objective function is linear relatively to s_{ij} and quadratic for the μ_j, which means that its optimization can be achieved with a close form solution.

Starting with an initial center vector for each cluster, the K-means iteratively minimizes Eq. 6.19 in the direction of s_{ij}, by firstly assigning each feature to its closest cluster (using the Euclidean distance and considering the current estimate of each μ_j). Then, the center vectors of the clusters are recomputed, for instance, the mean of all points assigned to each cluster. This computation is performed using Eq. 6.20 with the assignment result

obtained in the previous step. This process is repeated until any change in the assignments or a maximum number of iterations is reached.

$$\mu_j = \frac{\sum_{i=1}^{M} s_{ij}\mathbf{x}_i}{\sum_{i=1}^{M} s_{ij}}. \tag{6.20}$$

The K-means guarantees convergence during the iterative optimization because in each step the value of the objective function is reduced; however, it can converge to a local or global minimum [170].

The EM algorithm requires a higher computational effort in each iteration and more iterations are necessary for convergence comparatively to the K-means. Thus, the K-means is initially used to find a suitable initialization for a GMM and then, the result is adapted for the EM. The mixing coefficients can be initialized using a fraction of number of data points assigned to each cluster obtained by the K-means algorithm and the covariance matrices can be set to the sample covariances of the clusters [170].

The major advantage of the K-means algorithm is its robustness and low computational effort since it is relatively efficient and provides good results when data sets are distinct, for instance, the clusters are well separated. However, it requires a priori specification of the number of cluster centers, the Euclidean distance may not be a good space for representing the dataset, cannot handle noisy or nonlinear data and the initial cluster center (randomly selected) can lead to poor results.

6.3.3 Hybrid Hierarchical Optical Flow Segmentation

Hierarchical techniques create the clusters by merging observations using pairwise similarity measurements. Usually, there are two approaches to operate hierarchically-based algorithms: top-down or bottom-up. In the agglomerative clustering (bottom-up), the clusters are successively merged until the desired structure is obtained; however, the divisive clustering (top-down) successively divides the parent cluster into sub-clusters. The output is a hierarchical representation and the highest level has only one cluster. The hierarchical partitioning is commonly presented using the dendrogram.

The segmentation of flow fields groups the pixels with similar motion properties because they probably belong to the same motion model. Two operational steps, *refining* and *collecting*, form the hybrid hierarchical optical flow segmentation (HHOFS). It is called a hybrid method because its first phase combines divisive and agglomerative clustering schemes. The *refining* iteratively decomposes the flow field into a set of distinctive clusters that represent image regions with different motion models. It successively splits and

merges the clusters by measuring the fitness of the estimated affine model to all observations that constitute the cluster. Parameters of the affine model are initially computed considering only a set of randomly selected instances (or observations). The resulting clusters from the first stage give information about the spatial segmentation of the flow field and are used to accelerate the convergence of the clustering process in the second stage. The *collecting* phase successively merges the set of clusters that was obtained in the *refining* phase, using a hierarchical scheme with the Mahalanobis distance. Features such as the angle and magnitude of the dominant flow vector for each cluster (average-link clustering) are considered at this phase.

Please notice, the flow vector of each observation is defined along this chapter as $\mathbf{w} = (u, v)$ in Cartesian coordinates and $\mathbf{w}^p = (m, \psi)$ in Polar coordinates. Therefore, a single observation is $\mathbf{x} = (\mathbf{w}, \mathbf{w}^p, x, y)$, where (x, y) is the coordinate position.

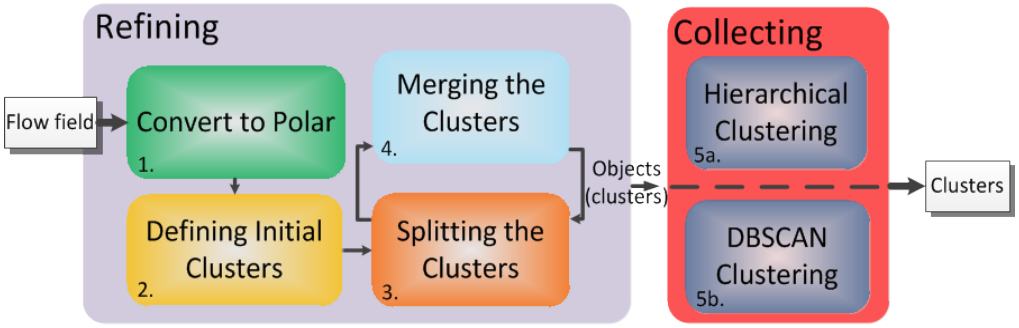
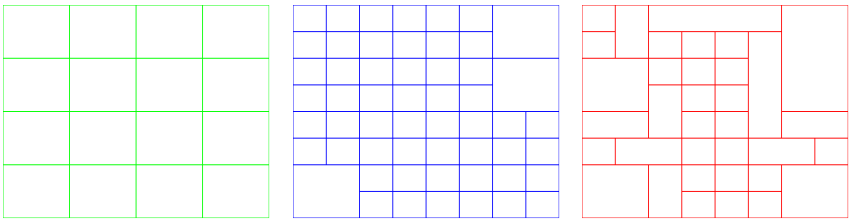


Figure 6.5: Architecture of the HHOFS and the HDBOFS methods. The overall structure and relations between different stages: *refining* and *collecting*. The difference between both methods relies on the *collecting* phase, for instance, the HHOFS and HDBOFS merge the clusters using a hierarchical and a density-based scheme, respectively.



(a) (b) (c)

Figure 6.6: 6.6(a) represents the initial and deterministic partitioning of the flow field 5.15(f) into clusters. A splitting procedure based on affine motion fitness divides some clusters into smaller and distinct subclusters, 6.6(b). 6.6(c) is the result obtained by merging the subclusters. It represents the decomposition of the flow field since the resulting clusters will be used (as objects) to initialize the *collecting* phase.

6.3.3.1 Refining phase

Figure 6.5 shows the structure of the HHOFS technique. The *refining* phase receives the flow field computed using *HybridTree* optical flow method [4] and returns a set of clusters that exhibit different motion characteristics. Each cluster represents a set of observations that are spatially related and share a similar motion model. The final set of clusters is considered objects for the *collecting* phase, which focuses on grouping these objects using a similarity measurement. The process starts by converting the horizontal and vertical velocities of the optical flow field, $\mathbf{w}$, to a Polar coordinate system $\mathbf{w}^p$. The *second stage* is responsible for an initial partitioning of the flow field into an initial set of clusters. This means that the flow field is uniformly divided into $W \times H$ non-overlapping regions, where W and H is the number of horizontal and vertical regions, respectively. The number of regions must be defined according to the smallest object that will appear.

The motion of an individual cluster can be represented using a six-parameter affine model[3], Eq. 6.21:

$$\hat{\mathbf{w}}(\hat{\mathbf{a}}, x, y) = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}, \tag{6.21}$$

where $\hat{\mathbf{a}} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ are the affine coefficients or parameters of the model. A solution for these equations can be obtained using at least 3 observations ($\mathbf{w}$ values) and the least squares method (LSQ) . Forty two observations belonging to the cluster are randomly selected in order to compute the affine parameters that describe their motion model[4]. The LSQ is not robust against noise or outliers; however, the robust estimation of these parameters using random sample consensus (RANSAC) or the iteratively re-weighted least squares (with the Charbonnier M-estimator) is more computationally demanding and, therefore, it can compromise the real-time demands of the robotic application that is presented in this thesis.

In reality, the robust estimation of the affine model is not an issue since each motion model is adjusted to the observations that constitute the corresponding cluster during the *third stage*. In this way, a fitting criterion is computed using the normalized residual error, see Eq. 6.22.

$$err_{sp}(\hat{\mathbf{a}}_j, j) = \frac{\sum_{i=1}^{M_j} \|\mathbf{w}_i - \hat{\mathbf{w}}_{ij}\|}{M_j}, \tag{6.22}$$

[3] Assumes that the depth variance in individual region is small enough compared with the distance from the scene to the camera.

[4] For a 95% of confidence level with 15 of confidence interval and considering 64 initial clusters.

where M_j is the number of observations of the jth cluster, $\hat{\mathbf{a}}_j$ are the affine parameters, $\mathbf{w}_i$ is the flow vector of the ith observation and $\hat{\mathbf{w}}_{ij} = (\hat{u}, \hat{v})^T$ is the estimated flow vector for the ith observation and considers the $\hat{\mathbf{a}}_j$. When the normalized error of the jth cluster is higher than a predefined threshold then, the cluster is characterized by more than a single motion model. Therefore, the cluster is split into four smaller clusters and the affine parameters of each subcluster are estimated once again. The splitting process is repeated for all the clusters and the next step is to merge motion models in the affine parameter space. This is the *fourth* and last stage of the *refining* task, see Fig. 6.5. The rectangular clusters c_s and c_r are merged if they are neighbors and share the same motion. To analyze the similarity of the motion properties for both clusters: Eq. 6.22 makes possible the fitting of $\hat{\mathbf{a}}_s$ into observations of cluster c_r and, furthermore, the model $\hat{\mathbf{a}}_r$ is fitted to observations of the c_s. This process is called by cross-validation and it originates two normalized errors that are combined as follows:

$$err_{mg}(s,r) = err_{sp}(\hat{\mathbf{a}}_s, r) + err_{sp}(\hat{\mathbf{a}}_r, s), \tag{6.23}$$

where $err_{mg}(s,r)$ is the merging error and $err_{sp}(\hat{\mathbf{a}}_s, r)$ is the normalized error considering the parameters of $\hat{\mathbf{a}}_s$ in the data of cluster c_r. Two clusters have similar motion properties when the merging error is lower than a threshold. After merging them, the affine model is re-estimated for the combined cluster to obtain more accurate model parameters.

Steps 3 and 4 are executed until the clusters converge or for a maximum number of iterations. Thus, the *refining* stage decomposes the dense optical flow field into a set of clusters. Each cluster defines a region of the flow field that shares the same affine motion model.

6.3.3.2 Collecting phase

The *refining* stage is a hybrid clustering scheme and the *collecting* phase is an agglomerative hierarchical-based scheme, where clusters obtained in the *refining* stage are considered as starting objects. The hierarchical clustering is conducted when the *refining* stage terminates without convergence and it computes the distance between two clusters using a similarity measurement in order to obtain a similarity matrix (distance between clusters).

This phase assumes that the observation is multivariate and normally distributed, and the feature vector has two dimensions $\mathbf{x} \in \Re^2$ since it is formed by the flow vector in Polar coordinates $\mathbf{w}^p$. Hence, the difference between two clusters, c_s and c_r, can be measured by a Mahalanobis squared distance of samples. The similarity between clusters

is considered in terms of a normalized difference between both mean vectors [171], $\overline{\mathbf{w}}_j^p$, and the positive-definitive covariance $\hat{\Sigma}$.

$$\hat{\Sigma} = \frac{\left[(M_s-1)\hat{\Sigma}_s + (M_r-1)\hat{\Sigma}_r\right]}{(M_s+M_r-2)}, \tag{6.24}$$

where M_j is the number of observations and $\hat{\Sigma}_j$ is the sample covariance matrix of the jth cluster.

The normalized difference between $\overline{\mathbf{w}}_s^p$ and $\overline{\mathbf{w}}_r^p$ is defined by $\overline{\mathbf{w}}_{s,r}$, and computed using Eq. 6.25.

$$\overline{\mathbf{w}}_{s,r} = \left(\frac{|\overline{m}_s - \overline{m}_r|}{m_{max}}, \frac{|g_{norm}(\overline{\psi}_s - \overline{\psi}_r)|}{\pi} \right), \tag{6.25}$$

where m_{max} is the maximum magnitude of the flow vectors, $\overline{m}$ is the mean of the magnitude and $\overline{\psi}$ is the mean of the angle. This equation normalizes and maintains positive the difference of the flow vectors that characterize each cluster. An important note is related to the angle, in radians. The angle subtraction is not straightforward because it must be followed by a normalization otherwise, the distance may be misleading. Equation 6.26 is used to normalize the result of the difference (or the sum) of angles, $\tilde{\psi} \in [-\pi;\pi]$.

$$\tilde{\psi} = g_{norm}(\psi) = atan2(sin(\psi), cos(\psi)). \tag{6.26}$$

Yielding the sample means $\overline{\mathbf{w}}_s^p$ and $\overline{\mathbf{w}}_r^p$ of clusters:

$$\Lambda_{cl}^2 = \overline{\mathbf{w}}_{s,r}^p \hat{\Sigma}^{-1} \overline{\mathbf{w}}_{s,r}^{p\,T}, \tag{6.27}$$

where Λ_{cl} is a metric that evaluates the distance between two clusters by considering the mean characteristics and confidence (represented by the covariance).

The clustering operation is an iterative process that merges two similarity clusters (lower distance) and the behavior can easily be represented using a dendrogram. The iterative process can be stopped when the similarity measurement is high, since the remaining clusters have higher distances and they probably should be disjoint clusters. Therefore, the process can stop according to a pre-specified number of clusters and/or a threshold value for the similarity.

The time complexity of original hierarchical algorithms is at least $O(M^2)$, where M is the total number of objects. In addition, the algorithms can suffer from sensitivity to noise and outliers according to the type of distance metric (or similarity measurement) that is chosen. However, the *refining* phase of the HHOFS creates a set of coherent clusters which reduces the number of objects that are used in the hierarchical scheme. This process

prevents the use of the initial objects at flow level, which reduces the computational costs of the hierarchical clustering. Therefore, advantages of the HHOFS segmentation include: no priori information about the number of clusters is required and the clustering process is conducted in a reliable and efficient manner.

6.3.4 Hybrid Density-Based Optical Flow Segmentation

This section presents a density-based technique for clustering dense optical flow fields. The structure of this technique is similar to the HHOFS and it is called Hybrid Density-Based Optical Flow Segmentation (HDBOFS). The major difference compared to the HHOFS is related to the *collecting* phase, which in Fig. 6.5 corresponds to step 5. This last stage is accomplished using the DBSCAN (density-based spatial clustering of applications with noise) clustering instead of a hierarchical clustering methodology. The DBSCAN method [185] is a density-based clustering algorithm since it finds clusters based on the density of data points inside a region. Usually, their advantage compared to hierarchical and partitioning methods is the computational complexity, which can be reduced to $O(MlogM)$. In addition, this method can discover clusters of arbitrary shapes [186].

The most important concept in the DBSCAN is its notion of density-reachability and density-connection [187]. These notions are defined by two parameters: the neighborhood's distance (ε) and the minimum number of points required to form a cluster ($minPts$). Consider a random point, p_s, this point will be directly density-reachable from a point p_r if the distance between both points is less than ε and if p_r is surrounded by at least $minPts$ points. Thus, the point p_s is called density-reachable which is an asymmetric property. The density-connected notion will now be introduced [187]: if there is a point p_c such that the two points p_r and p_s are density-reachable from p_c then p_r and p_s are density-connected. This notion is symmetric and makes it possible to define a cluster (a set of objects that are mutually density-connected) because if p_r belongs to some cluster and p_s is density-reachable from p_r, then p_s belongs to the same cluster. The process begins with a random point and its neighborhood is obtained using the density-reachable notion. If the size of its neighborhood is at least $minPts$, a cluster is started. Otherwise, the point is marked as noise. For the cases where no points are density-reachable from some point belonging to a cluster, then this point is a border object. When a point is found to be a dense part of a cluster, its neighborhood also belongs to that cluster. The process is repeated until all the points are visited. Table 6.1 depicts the behavior of the DBSCAN that is controlled by two parameters ε and $minPts$.

Briefly, the key idea of the DBSCAN is that for each data object, the neighborhood of a given radius (ε) must contain at least a minimum number ($minPts$) of objects. The major

Table 6.1: DBSCAN behavior for different configuration of the parameters.

ε	$minPts$	Result
Big	Big	A few, dense and large clusters
Big	Small	Large and less dense clusters
Small	Big	Small and dense clusters
Small	Small	Several smaller and less dense clusters

problem of density-based clustering algorithms is that they easily lead to memory problems when facing large datasets [186]. For this reason, the *refining* phase is performed initially. Thus, the proposed HDBOFS clustering method is based on the concept of space partitioning since it efficiently identifies the different densities in the dataset according to the motion properties. Next, the method performs a density-based clustering by taking into account a similarity measurement of the objects. This two phase method makes it possible to reduce the computational memory demands by providing a guideline based on coherent clusters (obtained by the *refining* stage) that are initially considered objects during the *collecting* phase.

The *refining* phase is similar to the HHOFS; however, the *collecting* stage is different, for instance, the clustering is not conducted by a hierarchical scheme but by using a density-based structure instead. The similarity measurement for the neighborhood's distance (ε) is defined based on a feature vector with two dimensions $\mathbf{x} \in \Re^2$ and formed by the flow vector in Polar coordinates $\mathbf{w}^p$. Hence, Eq. 6.27 is used to measure the distance between objects. The initial objects are obtained by the *refining* phase.

6.4 Wise Optical Flow Segmentation

Although the HHOFS be computationally efficient, the technique originates results with a blocky aspect which is not desirable for all surveillance applications. Therefore, this research presents the Wise Optical Flow Segmentation (WOFS) that is formed by two operational steps, *evaluating* and *resetting*. The *evaluating* phase decomposes the flow field into a set of distinctive clusters that represent image regions with different motion models. The affine model of each cluster is initially computed considering a set of randomly selected instances (or observations). A split-merge procedure is conducted by measuring the fitness of all the observations that constitute the cluster to the estimated affine model. Afterwards, a hierarchical scheme merges the set of clusters using the Mahalanobis distance of features such as the angle and magnitude of the dominant flow vector for each cluster (average-link clustering).

The *resetting* phase resorts to information regarding the spatial location of the different clusters and motion, during the segmentation of the colored representations of dense flow fields using a marker-controlled watershed technique. This watershed method has the advantage of flooding the topographic surface from the previously defined set of markers.

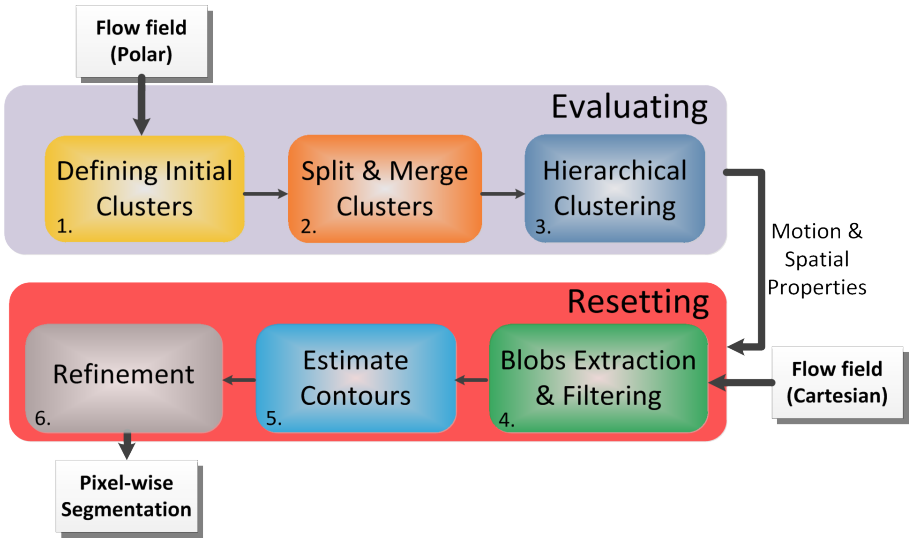


Figure 6.7: Architecture of the WOFS method.

6.4.1 Evaluation phase

Figure 6.7 shows the structure of the WOFS technique. The flow fields are computed using a dense optical flow technique [4] and the *evaluating* phase returns a set of clusters that exhibit different motion characteristics. Each cluster represents a set of observations that share a similar motion model. This clustering operation is obtained with the HHOFS method, presented in section 6.3.3. The method merges two similar clusters and the process stops when the similarity measurement is high, since the remaining clusters have higher distances which mean that they probably should be separated. In addition, the process can stop according to a pre-specified number of clusters. The *evaluation* of dense optical flow fields is a procedure that is computationally efficient and achieves a real-time performance (it takes 32ms to segment dense flow fields of 640×480).

6.4.2 Resetting phase

The *resetting* phase applies a marker-controlled watershed segmentation to the colored representation of dense flow fields. The watershed is a gradient-based technique that tries to prevent the over-segmentation effect, which means that the flooding process of the topographic surface is guided by a previously defined set of markers.

Stages 4 and 5 of the WOFS are responsible for defining the set of markers that best describe each cluster in terms of robustness. The main objective is to enhance the contour of the moving objects that can be characterized through the clusters that have been obtained so far. Information such as the spatial location and the motion model of the different clusters, originated in the last phase of the *evaluation* stage, are considered during the fourth stage of the WOFS method. This stage interprets the current block-wise clustering (see Fig. 6.1) and decomposes the clustering result into motion layers, where each level represents pixels that share the same motion model. Motion models were characterized by *evaluating* the flow field affected by a noise component (assumed to be normally distributed with a zero mean). Therefore, each motion layer is obtained by searching for all the flow vectors with a velocity (vertical and horizontal) belonging to a range given by the standard deviation and motion mean. These parameters are extracted from the motion profile of the clusters. Flow vectors with a velocity that matches the motion profile of a cluster are spatially filtered by taking into consideration the spatial location of the corresponding cluster and the position of these vectors in the flow field.

Afterwards, only a spatially-connected group of flow vectors (blob) is considered for each motion layer. In the fifth stage, the small blobs are removed from the motion layer since they will mislead the segmentation because these small blobs usually represent noisy flow vectors. Morphological operators are applied to the resulting blobs of the layered-structure in order to increase the spatial connectivity inside the blob and to reduce the size of their contours. The external contours are extracted and used to define a set of zones with unknown motion profiles that will be analyzed in the final stage of the WOFS.

Finally, all regions of the flow field that were marked as unknown zones are combined together in a single layer. Regions with unknown motion profiles usually represent the edges between distinct moving objects; however, they can also represent parts of moving objects that have a slightly different motion profile. For instance, arms and legs of a moving person have different motion models. These unknown zones capture inconsistent movements inside the cluster, which makes it possible to apply the watershed technique for analyzing the colored representation (in the Hue-Saturation-Value palette) of dense flow fields and, thus, fill the information gap. Therefore, this stage enhances the contours of moving objects and floods the topographic surfaces that represent noisy flow vectors.

6.5 Results

A comprehensive set of experiments were conducted as part of this work. They aim to analyze and understand the behavior of the designed techniques for motion analysis in

a robotic and surveillance context: segmentation of dense optical flow fields based on clustering approaches. The large majority of the experiments were conducted using the *EEyeRobot* in a real surveillance scenario (the corridor of our laboratories). Therefore, they depict real testing conditions which means that the visual system of the mobile robot is subjected to different light conditions, reflections, diffractions, shadows and ghost effects (due to glass walls). No filtering technique was previously applied to the sequences in order to maintain the reliability and the repeatability of the experiments. Otherwise, the results will be influenced by the type of the filter.

The first experiments focus on testing the accuracy of the model selection that was presented in section 6.2. The estimation of the value K (number of clusters) based on this method is compared to the true value, which is manually labeled. These trials resort to real dense flow fields, for instance, several realistic sequences that capture different types of motion are used by an optical flow technique [4] to generate the dense flow field. Then, the estimation of the number of clusters based on this method is compared to the value obtained from the BIC, AIC, HQIC and NEC criteria. The true value for the number of moving objects was manually labeled from the dense flow fields.

The second experiments present and analyze in detail the performance of each clustering technique, namely, the HHOFS and the HDBOFS. These techniques are compared to two alternative clustering methods, for instance, EM and K-means. In this way, it is possible to evaluate the pros and cons of block-wise techniques compared to pixel-wise techniques. Factors such as the computational effort and the quality of the visual segmentation are considered.

The third and last experiment aimed to provide a reliable characterization of the performance and computational cost involved during the WOFS. This technique is also compared to two alternative clustering methods, for instance, EM and K-means. Both baseline methods provide a pixel-wise segmentation and factors such as the computational effort and the quality of the visual clustering are considered. The assessment was performed using an objective (quantitative) and subjective (qualitative) evaluation. An objective metric, namely, the F-score [188] was used to provide quantitative quality evaluations of the clustering results since it is often employed [189, 190]. F-score is a weighted average of precision and recall that reaches the best value at 1, see Eq. 6.28.

$$Fscore = 2\frac{precision.recall}{precision + recall}. \tag{6.28}$$

All the results in this section were obtained with an I3-M350 2.2GHz computer and without parallel programing or GPU. The methods were implemented in C++ using the

commonly used OpenCV library[5]. The EM and the K-means are used in this research as baselines and implemented as standard functions. The real sequences were obtained using the *EEyeRobot* , for instance, the images have a resolution of 640×480 and were captured from a "The Imaging Source DFK 21AU04" camera with a 4mm focal lens.

6.5.1 Number of moving objects

The proposed model selection method does not need the complete dataset (flow field) to estimate the number of clusters. The computation of the value of K is demanding for a full dataset since the BFHE uses the K-means to obtain the likelihood. It returns the probability of each hypothesis which means that, it has a mechanism that makes it possible to evaluate the confidence of the estimation of K. A larger dataset can be used only when there are several hypotheses with similar probabilities and, in this way, a sampling procedure reduces the time spent during the selection of the model that fits better into the data. The results of the BFHE method reported in this research use a sampling process of 5% of the original dataset (more than 15000 flow vectors) due to real-time constraints.

Table 6.2: Comparison of the performance achieved by the BFHE with the BIC, AIC, HQIC and NEC criteria: the average accuracy and the average computational time. 30 dense flow fields were considered during this experiment.

	Accuracy	(in percentage)		
BIC	AIC	HQIC	NEC	BFHE
43.33%	23.33%	30.00%	40.00%	86.66%
	Time	(in seconds)		
BIC	AIC	HQIC	NEC	BFHE
0.0479	0.0480	0.0485	0.0142	0.0174

Table 6.2 compares the accuracy of the BFHE with the BIC, AIC, HQIC and NEC criteria. The computation of the log-likelihood for the BIC, AIC and HQIC criteria was obtained from the EM and the K-means was used for the NEC criterion. The table 6.2 shows that the BIC is the most accurate baseline criteria in most of the times. However, the BFHE outperforms all criteria since it achieves a global accuracy close to 90% in real sequences. In fact, the other criteria present severe difficulties when estimating the correct number of clusters (K_{gt}) since their performance is poor. The low accuracy of the baseline methods is caused by an overestimation of the K since these methods are significantly affected by different motion models that a single person has during its own motion. Motion analysis becomes even more difficult when parts of the body are depicted

[5]Version 2.4.3 of the OpenCV

in the sequence, for instance, if only the legs of a person are visible then each leg can easily be interpreted as a moving object. The proposed model selection algorithm is robust enough to recognize this type of situation. However, the presence of "noise" in the dataset reduces the accuracy of all the methods. This noise is caused by visual artifacts, such as the aperture problem, changes of illumination, shadows, reflections and sensor noise. These factors lead to a misleading estimation of the optical flow. The presence of noise is evident in some of the flow fields, for instance, Fig. 5.17(l) and the results shown that most of the faulty trials for the BFHE were obtained for dense flow fields corrupted with a similar noise. Furthermore, the BFHE achieves a global accuracy in real sequences close to 90%. This value should not be interpreted as the final accuracy of the method (for that, more trials must be conducted) but it gives a good picture of the method's capacity for estimating the number of clusters based on dense flow fields.

When examining the trials that originate wrong estimations by the BFHE, it was possible to confirm that the confidence ratio in 50 % of the faulty trials is lower than 1.1. This means that the likelihood of the best hypothesis that was returned by the method is only 1.1 times higher than the hypothesis that yields the K_{gt}. Most of the faulty trials were obtained in real cases with $K_{gt} = 1$. The egomotion of the robot creates one cluster; however, the optical flow technique and the environment's condition cause visual artifacts. The optical flow technique is limited to some issues that were discussed in chapter 5.

In addition, the BFHE method was implemented using the K-means in order to reduce the computational effort required to estimate K. Thus, the likelihood is computed after the K-means and the results that were obtained are similar to the EM. Therefore, the BFHE is capable of estimating the number of clusters, regardless of the type of flow field and it takes 18 milliseconds (on average) to compute, which is a proof of its low computational complexity. The results should not be interpreted as the final accuracy of the method but it provides a good picture of the method's capacity for estimating the number of clusters based on dense flow fields.

6.5.2 Separating the estimated motions

Several experiments were conducted in order to evaluate the behavior of the HHOFS and the HDBOFS. Factors such as the quality of the visual segmentation and the computational performance are considered. Both techniques are compared to the well-known EM and K-means. The EM and K-means are parametric techniques and, hence, they need information about the number of the clusters. For this reason, the information related to the K_{gt} is available during the experiments: the EM, K-means and HHOFS can be parameterized with the value of K being automatically configured by the BFHE. Instead, the

HDBOFS is more complex to setup and it was manually configured in this section.

6.5.2.1 Visual segmentation

The results start by presenting the most difficult case which is depicted in Fig. 5.15(i) and it represents two people moving in the opposite direction. The *EEyeRobot* is moving in the same direction as the person on the left when capturing this scene. The flow field is shown in Fig. 5.15(l). This situation demonstrates the difficulty of segmenting different types of motion because the egomotion of the robot is in the same direction as one person which may mislead some clustering processes. This is why the features space and the similarity metric are so important because if they do not reflect the difference between distinct motions correctly, then the segmentation will produce poor results. In Fig. 5.15(l) it is possible to visualize an interaction region between both people, which leads to an area of confusion since the people are spatially close. The optical flow technique [4] also measures the apparent velocity of shadows and, therefore, it creates an interaction between the two motions in the flow field. This area increases the difficulty of extracting three clusters (egomotion, person on the left and right).

Using the EM and the K-means for clustering the flow field in Polar space resulted in Figs. 6.8(a) and 6.8(b), respectively. The segmentation conducted by the EM produces 3 clusters. However, the clustering process does not originate a suitable segmentation because the clusters of people appear larger and they have spatially isolated regions that are meaningless (hereafter, called clustering noise). The result of the K-means is better than the EM because the clusters depict more faithfully the person's movement. In addition, the clustering noise is lower than the EM.

Figures 6.8(d) and 6.8(e) show the motion segmentation based on the HHOFS and HDBOFS. Both methods resort to a flow field in Polar representation and they do not deal with a single flow vector, as in previous techniques. Therefore, the results have a blocky aspect due to the *refining* phase, see Fig. 6.8(c). The segmentation conducted by the HHOFS produces the best result because there is no clustering noise: only a small region in the person on the right is misclassified; however, this is not a real misclassification because the velocity of this small region is similar to the person on the left, see the flow field. The *collecting* phase of the HHOFS is more robust to the presence of shadows since the resulting clusters only represent the people. In the last result, the parameter's configuration of the HDBOFS was only able to extract a pair of clusters: the two people are combined in a same cluster. This may be caused by the setup of the HDBOFS or by the density-based concept (*collecting* phase) because the region of confusion is joining both clusters and, thus, it misled the entire clustering process.

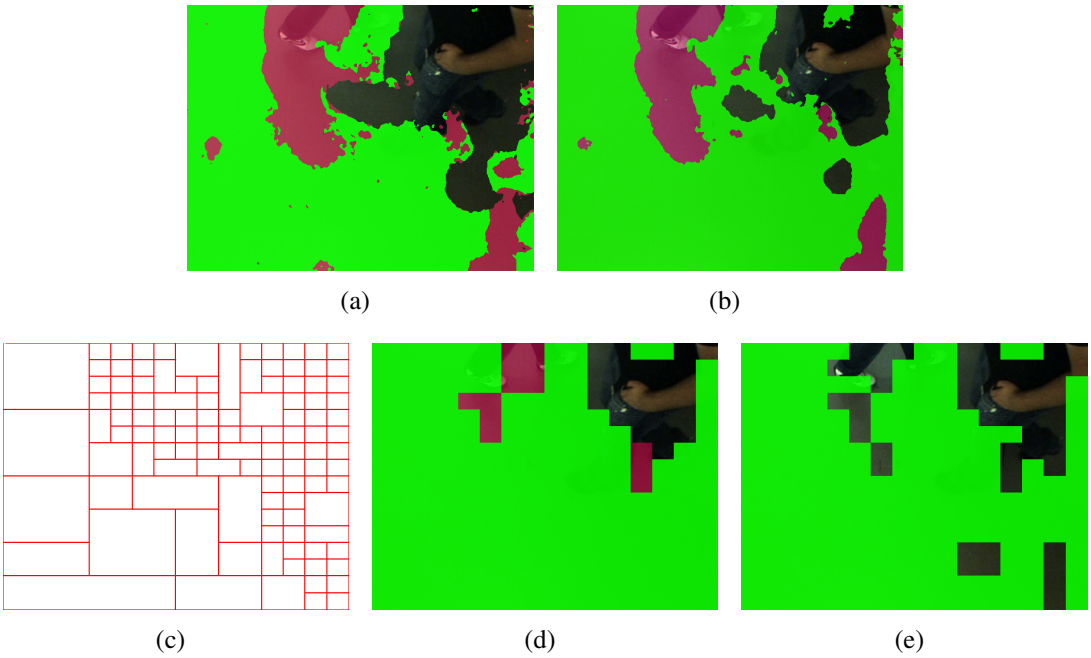


Figure 6.8: Motion segmentation for the case 5.15(i) and using the flow field 5.15(l). Comparison between the EM, K-means, HHOFS and HDBOFS methods, Figs. 6.8(a), 6.8(b), 6.8(d) and 6.8(e), respectively. 6.8(c) depicts the *refining* phase of the HHOFS and the HDBOFS.

Segmentation results are presented for the flow fields in Figs. 5.15(e), 5.15(f) and 5.15(j). These cases reflect situations where the *EEyeRobot* captures moving people with different motion models. They aims were to analyze and compare the robustness of the clustering procedures. Figures 6.9(a), 6.9(b) and 6.9(c) depict the segmentation conducted by the EM and Figs. 6.9(d), 6.9(e) and 6.9(f) presents the results of the K-means. These results are baselines for the proposed HHOFS (Figs. 6.9(g), 6.9(h) and 6.9(i)) and the HDBOFS, Figs. 6.9(j), 6.9(k) and 6.9(l).

The visual illustration of the motion segmentation based on dense flow fields shows that the EM produces clusters affected by noise. The K-means has a visual segmentation that it is close to the EM, although with a lower amount of noise. Both methods were able to characterize the two motion models present in each trial; however, the clustering noise is higher than the results of HHOFS and HDBOFS. This is caused by the architecture presented in Fig. 6.5, for instance, the *refining* phase makes it possible to gather the flow vectors that are related in space and share a similar motion model. This increases the robustness to noise and reduces the computational effort during the agglomeration of motion models (*collecting* phase). Figures 6.9(g), 6.9(h) and 6.9(i) show that the HHOFS produces the best visual segmentation since the clustering noise is small and the resulting clusters reliably represent the different motions. In addition, the HHOFS is easy and intu-

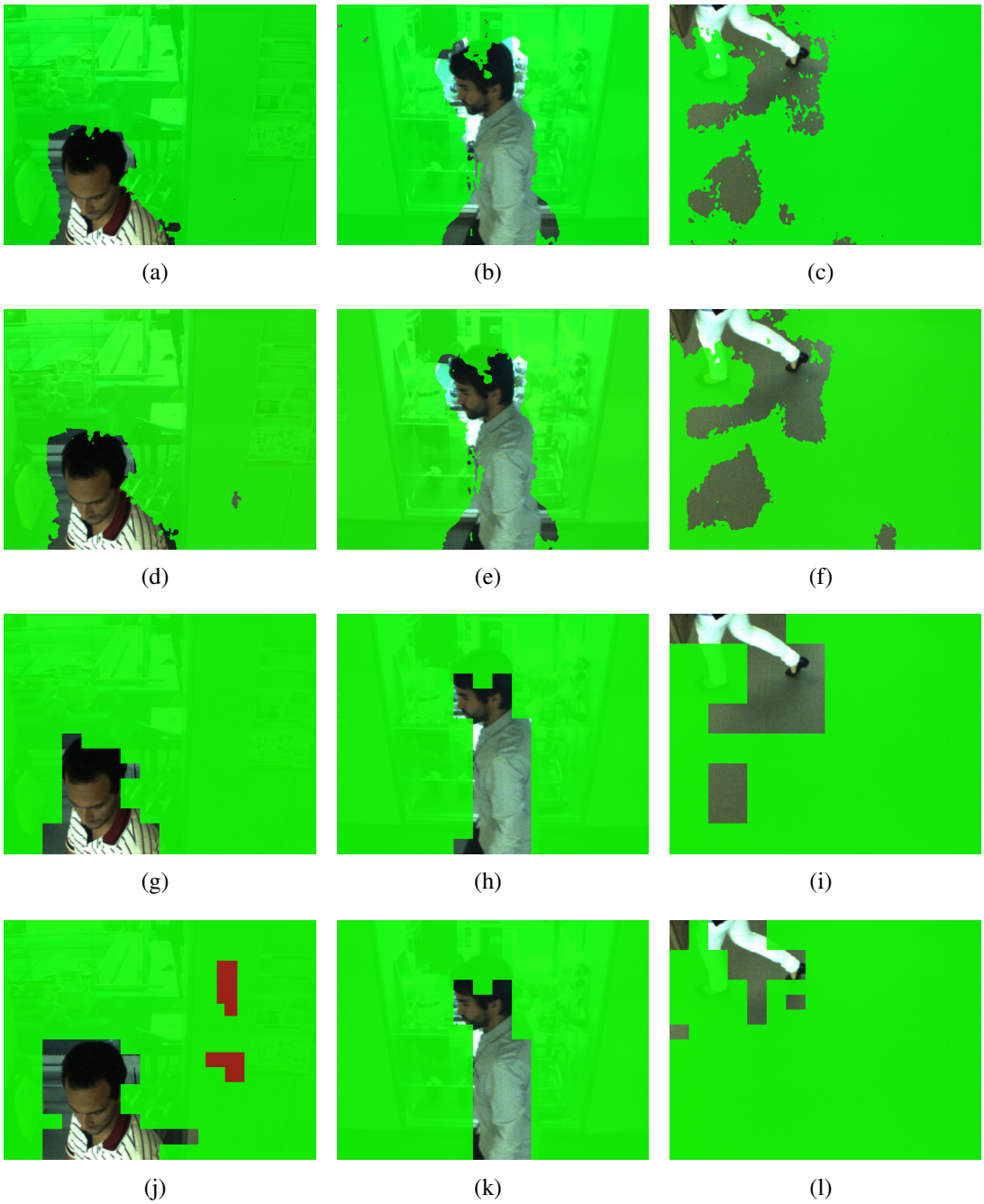


Figure 6.9: Motion segmentation for the cases 5.15(b), 5.15(c) and 5.15(g): the flow fields 5.15(e), 5.15(f) and 5.15(j) were obtained from the multi-channel formulation of the *HybridTree* technique. Comparison between the EM (first row), K-means (second row), HHOFS (third row) and HDBOFS (fourth row) methods.

itive to configure because it requires only the maximum similarity level between clusters and/or the number of clusters. This makes possible to setup the technique in running time and according to the BFHE model selection method. On the other hand, the HDBOFS is more difficult to configure since its parameters are less intuitive to setup in this context. This problem is well-known in the literature [191]. Without a proper configuration, clustering based on the HDBOFS could result in a poor visual segmentation (more frequent when more than 2 motion models are depicted in the flow field). Figures 6.9(j), 6.9(k) and 6.9(l) show the results for the HDBOFS. As it is possible to confirm, the technique returns a segmentation that is similar to the HHOFS; however; the clustering noise is higher because the *collecting* phase is executed based on the DBSCAN and not hierarchically. This strongly influences the clustering process since it changes the order in which the clusters are grouped (the density-reachability and the density-connected concepts). Figure 6.9(j) shows the influence of this issue since the visual segmentation shows in fact 3 clusters: the smaller cluster (dark purple) on the upper right side has an optical flow (green region in Fig. 5.15(e)) that is quite different from the rest of the clusters. Thus, the DBSCAN isolates this cluster since it does not consider the number of clusters like the HHOFS, but considers the ε and the $minPts$ instead.

From the results shown, it is possible to see that the visual segmentation of the HHOFS is suitable for robotic and surveillance applications because the parameters of the technique can be adjusted in running time. Furthermore, it is reliable and more robust to the influence of shadows that cause regions of confusion. In addition, density-based clustering approaches are not recommended for segmenting dense flow field, especially, for robotic and surveillance applications that are operating in a dynamic environment.

6.5.2.2 Computational performance

The HHOFS and the HDBOFS can have different computational requirements according to the parametrization of the *refining* phase, for instance, the initial division of the flow field space into non-overlapping regions. In this context, table 6.3 presents the expected performance of the four methods during the clustering of all real cases presented in section 5.4.1. The table shows that the K-means is computationally efficient since the processing time is a fraction of the time spent by the EM (with 3 iterations). As confirmed, the visual segmentation of the K-means is very close if not better than the EM and, thereby, the K-means reveals better characteristics for motion segmentation of dense flow fields in robotic applications.

As expected, the HDBOFS is computationally more efficient than the HHOFS. Several experiments were conducted considering different resolutions for the *refining* phase.

Table 6.3: Comparison of the computational performance between the EM, K-means, HHOFS and HDBOFS. The performances of the proposed methods were evaluated by considering different initial resolutions in the *refining* phase: 4×4^a and 8×8^b. The time is given in seconds.

Flow Field	EM	K-means	HHOFS^a	HDBOFS^a	HHOFS^b	HDBOFS^b
Fig. 5.15(d)	1.2417	0.0203	0.0201	0.0253	0.5011	0.3632
Fig. 5.15(e)	3.0477	0.1216	0.0462	0.0270	0.5880	0.3872
Fig. 5.15(f)	2.9271	0.1200	0.0289	0.0291	0.5794	0.3719
Fig. 5.15(j)	3.0347	0.1493	0.0398	0.0329	0.5809	0.3759
Fig. 5.15(k)	2.7604	0.1214	0.0309	0.0279	0.5871	0.3795
Fig. 5.15(l)	4.3106	0.1951	0.0321	0.0284	0.5477	0.3670

Table 6.3 shows that increasing the resolution leads to a higher processing time because, commonly, there are more initial regions for the *collecting* phase, which increases the clustering time. A direct comparison between the HHOFS and the HDBOFS can be found in the table. Going from a 4×4 to an 8×8 resolution increases the processing time of the HHOFS and the HDBOFS on average by 18.57 and 13.3 times, respectively. In theory, the clustering time of Fig. 5.15(d) should remain smaller in both resolutions because the *refining* phase makes it possible to detect the presence of only one motion model; however, this does not happen because the flow field that was computed is not perfect due to some noise and visual artifacts.

The HDBOFS is more computationally efficient but it has parameters that are more difficult to setup. Moreover, the visual segmentation produced by this method is usually worse when compared to the HHOFS. Therefore, the HHOFS makes it possible to segment motion from dense flow fields and in real-time. For instance, the technique took 33 milliseconds on average to analyze the flow field and to extract the set of clusters. In this way, the HHOFS seems to be a more balanced technique when considering both computational efficiency and the quality of the clustering process.

The results have shown that pixel-wise techniques are better to discriminate the shape of moving objects but less robust to the aperture problem and less computational efficient. On the other hand, experiments prove that the HHOFS and the HDBOFS achieve a real-time performance with lower computational resources (without parallel programming). Their results share a blocky aspect which is acceptable for the *EEyeRobot* however, a pixel-wise technique starting from the results of HHOFS and HDBOFS can enhance the visual segmentation of the flow field.

6.5.3 Motion segmentation of dense flow fields

An extensive set of experiments was conducted in this section and to evaluate the clustering performance of the WOFS method for all the real cases presented in section 5.4.2. Factors such as the quality of the visual segmentation and the computational performance are considered during this evaluation. The proposed technique is tested with well-known and frequently used clustering techniques, namely, Expectation-Maximization and K-means.

The section starts by presenting the clustering results that were obtained by the methods: Figs. 6.10(a) to 6.11(l) show the results of WOFS, EM and K-means, where each row depicts the multiple results for the same flow field. The visual illustration of the motion segmentation based on dense flow fields shows that the EM produces clusters affected by relevant noise, for instance, Figs. 6.10(b), 6.10(e) 6.10(h) and 6.11(b). The K-means produces a clustering result that it is better than the EM since the results depicted in Figs. 6.10(c), 6.10(i), 6.11(c), 6.11(f) and 6.11(i) have a substantially lower amount of noise. Both methods were able to characterize the two motion models present in each trial with more or less difficulty; however, they present clustering noise (small blobs) that affects the quality of the clustering which is not suitable for surveillance and robotic applications. Figures 6.10(a), 6.10(d), 6.10(g), 6.10(j), 6.11(a), 6.11(d) and 6.11(g) show that the WOFS produces the best visual segmentation since the clusters reliably represent the different motions, the edges and the contours. The moving objects are clearly extracted, and the clustering noise is nonexistent. The clear advantage of the WOFS technique over the EM and the K-means is related to its architecture (Fig. 6.7) because the *evaluating* phase makes it possible to gather a set of flow vectors that are related in space and share a similar motion model. This increases the robustness against noise and reduces the computational effort that is required to cluster the set of motion models. In addition, the WOFS is intuitive to configure because it only uses a few parameters that can be configured in running time and according to the BFHE model selection method.

The most difficult case for motion analysis is depicted in Fig. 5.17(h) since it represents two people moving in the opposite direction when the observer is moving in the same direction as the person on the left. The flow field is shown in Fig. 5.17(k) and it demonstrates the difficulty of segmenting different types of motion because the egomotion of the robot is in the same direction as one person which may mislead the clustering processes. In this case, the feature space and the similarity metric are relevant because if they do not reflect the difference between distinct motions correctly then, the segmentation will produce poor results. Figure 5.17(k) shows an area of confusion because it creates an interaction between the two motions in the flow field which increases the difficulty of extracting and segmenting three clusters. Figures 6.11(k) and 6.11(l) show the

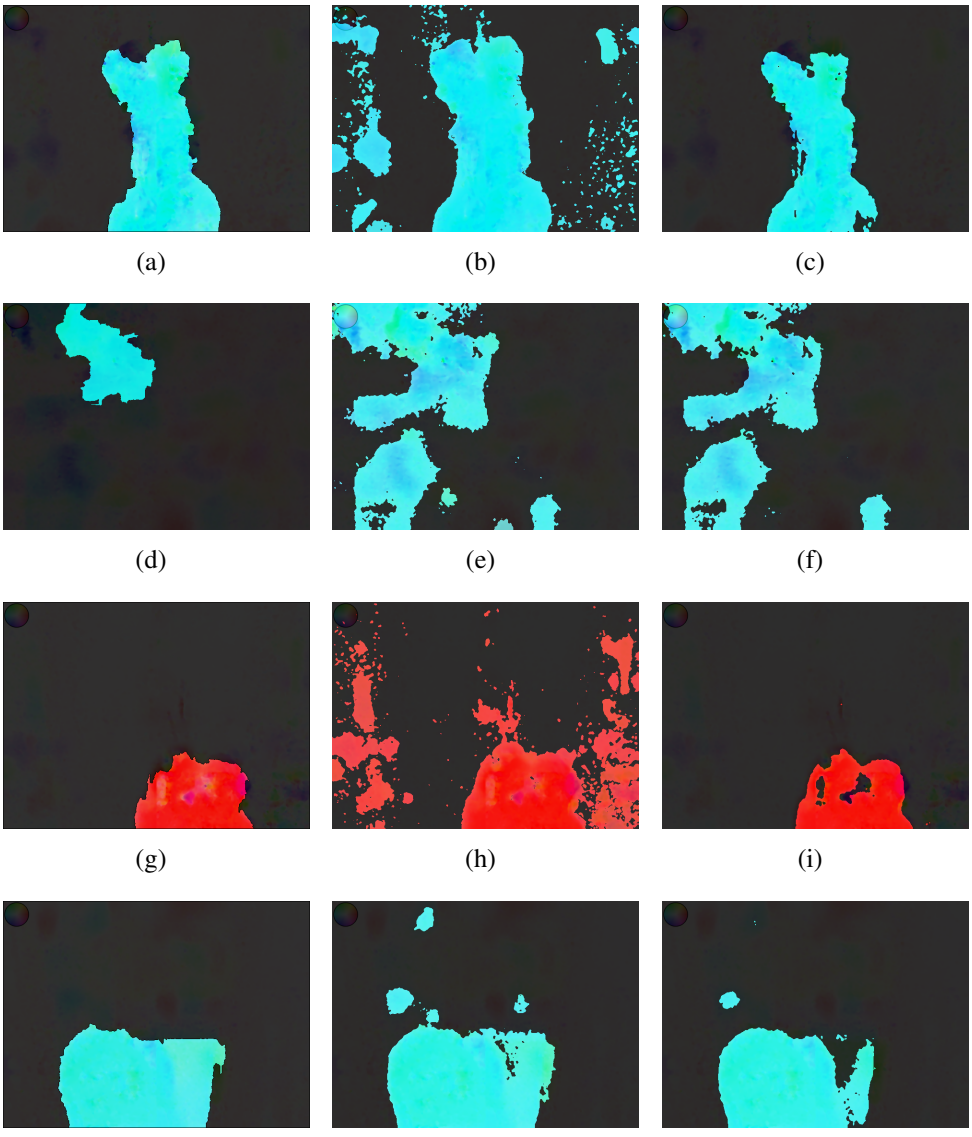


Figure 6.10: Motion clustering for the cases 5.16(d), 5.16(e), 5.16(f) and 5.17(d). Comparison between the WOFS (first column), EM (second column) and K-means (third column).

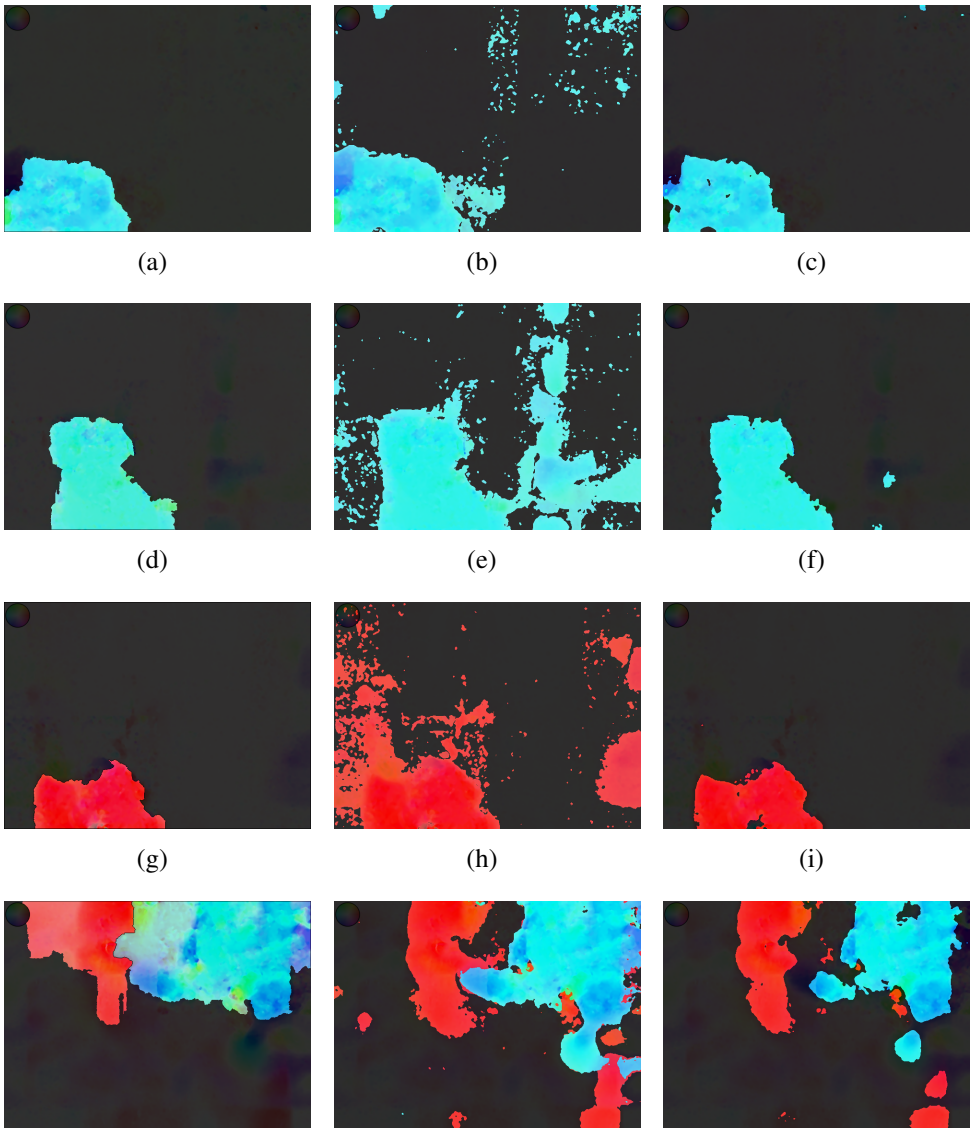


Figure 6.11: Motion clustering for the cases 5.17(e), 5.17(f), 5.17(j) and 5.17(k). Comparison between the WOFS (first column), EM (second column) and K-means (third column).

clustering of the flow field 5.17(k) conducted by the EM and the K-means, respectively. As it can be seen, the EM produces 3 clusters (dark, blue and red); however, the process originates clusters that are larger than the motion of the people and there are spatially isolated blobs that are meaningless. The result of the K-means is better than the EM because the clusters depict the person's movements more reliably and the clustering noise is lower. Clustering flow field based on the WOFS 5.17(k) is depicted in Fig. 6.11(j). This technique produces the best visual result because there is no clustering noise and the size of clusters that represent the two moving people are appropriate (the shapes of the people are retrieved with accuracy). The WOFS is more robust to the presence of the outliers (noisy flow vectors) since the result is spatially-connected and the contour of each cluster is constrained to the region of the flow field with motion properties that match the motion profile of the moving person. However in this case, the area of confusion causes an over-segmentation of clusters which effects the quantitative evaluation of the proposed technique, table 6.4.

Table 6.4: F-score - Performance comparison between the EM, K-means and WOFS. Parameters such the precision ("Prec.") and the recall ("Rec.") are presented. ^{a} represent the clustering result for the two foreground clusters of Fig. 6.11(j).

Fig.	EM			K-means			WOFS		
	Pre.	Rec.	F-score	Pre.	Rec.	F-score	Prec.	Rec.	F-score
5.16(d)	0.586	0.967	0.730	1.000	0.853	0.921	1.000	0.931	**0.964**
5.16(e)	0.371	0.984	0.539	0.560	0.923	0.697	0.942	0.560	**0.703**
5.16(f)	0.514	0.998	0.679	0.988	0.860	0.920	1.000	0.903	**0.949**
5.17(d)	0.785	0.941	0.856	0.971	0.869	0.917	0.998	0.973	**0.986**
5.17(e)	0.365	0.972	0.531	0.999	0.853	0.920	1.000	0.896	**0.945**
5.17(f)	0.357	0.976	0.523	0.983	0.943	0.962	1.000	0.974	**0.987**
5.17(j)	0.264	0.990	0.417	0.998	0.900	0.946	0.999	0.928	**0.963**
5.17(k)^{a}	0.417	0.966	0.583	0.621	0.989	**0.763**	0.618	0.913	0.738
5.17(k)^{a}	0.833	0.913	0.871	0.977	0.790	**0.874**	0.675	0.907	0.774

Several experiments were conducted in order to provide a direct comparison between the WOFS with the EM and the K-means. Manually annotated images[6] that represent the ground truth for the segmentation of the flow fields were considered during the quantitative evaluation, see table 6.4. The poor visual performance of the EM is confirmed by the objective evaluation since its results have originated the lowest F-score in every flow fields. Statistically, the WOFS achieved the best performance since the technique produced clusters with the best F-score in 87.5% of the flow fields. Although, the K-means

[6]These images are available in `http://paginas.fe.up.pt/~dee10015/_wofs.htm`.

achieved interesting results since its performance was usually close to the WOFS, and better for the flow field 6.11(j) (due to aspects that already were discussed). In most of scenarios, the WOFS reaches high precision and recall which demonstrate the ability to segment dense flow fields.

Finally, table 6.5 presents the expected computational performance of the WOFS, EM and K-means. The table shows that the K-means is computationally efficient since the processing time is a fraction of the time spent by the EM (with 3 iterations). As previously confirmed, the visual segmentation of the K-means is better than the EM and, thereby, the K-means reveals better characteristics for motion clustering of dense flow fields in robotic applications. However, the WOFS is computationally more efficient than the other methods since the processing time of the EM and the K-means is on average 95.03 and 1.51 times higher, respectively.

Table 6.5: Computational performance comparison between the EM, K-means and WOFS. The time is given in seconds.

Flow Field	EM	K-means	WOFS
Fig. 5.16(d)	7.2821	0.0904	**0.0627**
Fig. 5.16(e)	3.2812	0.1010	**0.0657**
Fig. 5.16(f)	7.0561	0.0908	**0.0614**
Fig. 5.17(d)	3.0945	0.0904	**0.0663**
Fig. 5.17(e)	4.5414	0.0903	**0.0623**
Fig. 5.17(f)	10.077	0.0926	**0.0622**
Fig. 5.17(j)	7.0774	0.0902	**0.0617**
Fig. 5.17(k)	8.3766	0.1623	**0.0921**

The WOFS makes it possible to segment motion from dense flow fields in a short period of time and using generic computer systems, for instance, it took 66 milliseconds on average to perform a clustering that is better than the other techniques. Therefore, the WOFS is a balanced technique when considering both computational efficiency and visual segmentation quality.

6.6 Final considerations

Motion analysis techniques based on moving observations are still in a preliminary stage when compared to static observations because the motion of the observer creates new paradigms that make the analysis even more complex and challenging. This chapter studied the real-time motion segmentation based on dense optical flow fields for mobile robotic applications. It proposes two block-wise segmentation methods, called the

Hybrid Hierarchical Optical Flow Segmentation (HHOFS) and the Hybrid Density-Based Optical Flow Segmentation (HDBOFS), that are able to extract different types of motion. Both methods are composed by two stages: the *refining* phase decomposes the flow field into distinctive clusters that represent image regions with different motion models and the *collecting* phase merges the set of clusters using the Mahalanobis distance and the hierarchical or the DBSCAN scheme. In addition, a model selection method is presented. The method is called Bayesian Fusion of Histogram and Entropy (BFHE) and combines the histogram analysis of the flow field in Polar coordinates with the highest decay ratio of the normalized entropy criterion (NEC). This scheme makes it possible to infer about the number of clusters in the flow field; however, this information is not directly required by the HHOFS, HDBOFS and WOFS, although, it enhances the quality of the segmentation.

This research also presents an innovative technique for segmenting moving objects at flow level. The technique is called Wise Optical Flow Segmentation (WOFS) and was designed for mobile robotic applications with real-time demands and computational constraints. Unlike traditional methods, the proposed technique uses high level information of the flow field to guide the segmentation process. The WOFS interprets the flow field and identifies areas with similar motion profiles. This stage is called *evaluation* and returns a provisional clustering result that has a blocky appearance. In this way, information regarding distinct motion profiles is considered as high level information during the second stage, which is called *resetting*. This information makes it possible to guide a marker-controlled watershed technique that resorts to the colored representation of the dense flow field to enhance the contours and the edges of moving objects.

Experiments conducted have proven the ability and flexibility of the HHOFS to segment different motions that may be present in a realistic surveillance scenario. In addition, the HHOFS and the HDBOFS proved to be computationally efficient with regard to other techniques reported in the literature and to conventional clustering techniques, for instance, Expectation-Maximization and K-means. In some cases, the K-means presented an interesting visual segmentation with low computational requirements. However, the HHOFS and the HDBOFS can deal with noisy flow fields since they are less affected by the quality of the optical flow estimation. The parameters of the HHOFS are very intuitive and simpler to adjust in running time. The HHOFS can lead to reliable results using only the number of clusters that is provided by the BFHE and neglecting the maximum similarity distance between clusters. On the other hand, the HDBOFS is more computationally efficient, especially for higher resolutions in the *refining* phase; however, their parameters are less intuitive in our context, which make the method more difficult to setup in running time.

The proposed block-wise techniques meet the computational demands of common

robotic systems since they segment dense flow fields (with a resolution of 640 $\times$ 480) in less than 35 milliseconds, without specialized hardware or parallel programming. Moreover, the results of the WOFS show that incorporating high level information of motion characteristic in the clustering procedure is advantageous. The WOFS achieves a better perceptual quality and a higher computational efficiency when compared to other approaches, namely, the Expectation-Maximization and the K-means. The experimental evaluations reveal that the proposed technique achieved the best F1 score. Therefore, the proposed motion analysis technique meets the computational requirements of common surveillance systems, as it can segment the flow field in less than 66 milliseconds (dense flow fields with a resolution of 640 $\times$ 480), without specialized hardware or parallel programming.

Finally, the BFHE outperforms other model selection methods, even when subjected to different moving objects in several conditions. Factors such as the aperture problem, changes of illumination, shadows, reflections and the sensor noise cause a misleading estimation of the optical flow. These visual artifacts are present in realistic scenarios and affect the performance of all the motion selection methods that were tested in this research. However, the BFHE is more robust to these issues, which makes the method more suitable for real robotic applications.

In short, two reliable and efficient methods were developed as part of this study (HHOFS and WOFS). They make possible to analyze motion in dense flow fields and limited computational applications, such as robotic and surveillance systems. Therefore, the moving robot is capable of analyzing external motions, providing important information for the detection and tracking of danger situations, namely, an intrusion, an unrecognized object, or even, for access control and person identification.

Chapter 7

Conclusion

7.1 The final assessment

Motion perception and analysis can be performed using two distinct ways which are directly related to the movement of the observer that is capturing the scene: *stationary observation* or *moving observation*. The large majority of works about perception resort to fixed cameras. However, this research goes one step further by discussing motion detection and analysis for a new generation of robotic moving systems.

In this context, the thesis has proposed a set of novelties for the areas of robotics and computer vision. In more detail, an innovative mobile robotic system called *EEyeRobot* was designed for active surveillance. This mobile robot moves along a rail and is equipped with a monocular camera. The major advantage of the *EEyeRobot* relatively to conventional systems is its ability to perform surveillance procedures without crowding the environment with cameras. The application that controls the robot is a distributed software and the perception architecture has two operating modes that are triggered according to the vehicle's motion [19]: static perception and dynamic perception. Scientifically, this research was focused in motion analysis based on moving observations. It proposed a dynamic perception scheme that is formed by: a spatiotemporal filter [20] to enhance the visual appearance of image sequences, a method to compute dense flow fields [4] in a short period of time and several techniques [22] to extract distinct motion models from the flow fields.

The novel filtering technique is named robust bilateral and temporal (RBLT) [20]. The RBLT assumes a temporal correlation for the pixel brightness over time which is valid if temporal increments between consecutive images are small. Furthermore, robust error norms were incorporated in the formulation of the filter which decreased the influence

of outliers during the estimation of the pixel value. The RBLT was compared to state-of-the-art methods and the performance of the filter was evaluated in realistic scenarios. The experiments conducted with Gaussian and Salt-and-Pepper noise proved that it is possible to reduce the noise component even in images having low SNR, that is, extremely degraded image sequences. Results showed that this filter meets the visual requirements of a surveillance system based on a mobile robot.

The technique that estimates the optical flow of colored images sequences is called *HybridTree* [4]. The technique interprets the images, identifies areas with distinct motion characteristics and assigns the optical flow technique that best suits for each image region. The *HybridTree* resorts to a methodology that blends in a symbiotic and hierarchical scheme the advantages of local and global optical flow formulations. The experiments have demonstrated that the proposed method extracts visual motion information in a short period of time and is more suitable for applications that do not have specialized computer devices: it took less than 150 milliseconds[1] to provide an acceptable estimation of the flow field, which demonstrates that incorporating high level information on the image sequence during the optical flow estimation is advantageous for many robotic applications.

Extensive results have demonstrated that dense optical flow fields provide relevant information about motion; however, their interpretation is a difficult problem. For this reason, the thesis studied the real-time motion analysis based on dense optical flow fields. A model selection method that infers about the number of motion models in the flow field was presented. The method is called Bayesian Fusion of Histogram and Entropy (BFHE) and combines the histogram analysis of the flow field in Polar coordinates with the highest decay ratio of the normalized entropy criterion (NEC). In addition, two motion segmentation techniques were presented: the Hybrid Hierarchical Optical Flow Segmentation (HHOFS) and the Hybrid Density-Based Optical Flow Segmentation (HDBOFS). The methods analyze different types of motion by decomposing the flow field into a set of distinctive clusters that represent image regions with different motion models and then, they merge the set of clusters using the Mahalanobis distance and the hierarchical or the DBSCAN scheme, respectively. The experiments proved that the BFHE and the HHOFS methods extract reliable visual motion information in a short period of time and they are more suitable for applications without specialized computer devices (the HHOFS took on average 34 milliseconds to separate motions). During these experiments, the BFHE was capable of estimating the number of clusters, regardless of the quality of flow field. The proposed model selection algorithm was robust enough to recognize motion in noisy

[1]The results in this section were obtained for images or flow fields with a resolution of 640×480 and a I3-M350 2.2GHz.

dense flow fields because the histogram analysis identifies the group of pixels with similar motion properties.

Although more computational efficient than state-of-the-art algorithms, the HHOFS and HDBOFS originate results with a blocky aspect [22]. Thus, an extension to the HHOFS is proposed in this thesis, that aimed to provide a pixel-wise segmentation of the moving objects. The technique is called Wise Optical Flow Segmentation (WOFS) and it was designed for mobile robotic applications with real-time demands and computational constraints. Unlike more traditional methods, the proposed technique uses high level information of the flow field to guide the process of motion segmentation. The WOFS interprets the flow field and identifies areas with similar motion profiles. This information makes it possible to guide a marker-controlled watershed technique that resorts to the colored representation of the dense flow field and enhances the contours and the edges of moving objects. Results show that this technique achieved a F-score on average of 0.906 and it took less than 66 milliseconds to segment flow fields without specialized hardware or parallel programming.

Concluding, this thesis provided a system for motion analysis that is capable of perceiving and understanding external motions. The system is able to achieve a processing frequency around of 5.5 and 6.6 frames per second when the HY and the HHOFS are executed in sequential and in parallel (causing a temporal delay of 1 frame), respectively. This research has studied motion analysis based on dense optical flow fields and for practical use in a computationally-constrained robotic application. Dense optical flow fields provide good information of the apparent motion for mobile robotic applications; however, extracting regions with similar motion characteristics is a complex and challenging procedure that requires sophisticated techniques. Therefore, the implications of this research lead to innovations in several important areas, such as computer vision, surveillance and mobile robotics: increasing the robot's ability, intelligence and autonomy.

7.2 Future works

Motion perception and analysis can significantly expand several areas of application: track human behaviors, correct the camera jitter (stabilization), aligning images into mosaics, thee-dimensional shape reconstruction and special effects. The proposed robotic application is currently installed in the Faculty of Engineering of the University of Porto and conducts active surveillance. Nonetheless much work can be developed in order to provide a reliable motion perception system for generic robotic applications that resort to MOb. A better interpretation of the scene makes it possible to automate the surveillance

processes that nowadays are carried out through remote monitoring. Yet, it is important to run tests and validate the performance of the proposed techniques in other scenarios. This can help scaling new setups for different applications in matters of both performance and cost.

The architecture for motion perception that is proposed in this thesis is supposed to lay the basis for new methods for tracking and recognizing human behaviors. The surveillance at homes or business is a growing market for service mobile robots; and the supply-demand law is currently requiring for robotic solutions with the ability to change from a conservative behavior (with a low profile but still aware of what is happen in the security zone) to a more showoff profile. From here, an exciting approach to explore will be the development of different robotic behaviors with the ability to evolve according to factors of internal and external nature, for instance, the mission's goals and the dynamic elements that populate the environment.

One of the most relevant improvements to this research will be the enhancement of hardware and software for the mobile robot in the following terms. Embedding the dynamic perception module in the onboard computer unit will lead to a more efficient operation of the *EEyeRobot* since the controllability of the autonomous behavior is currently affected by communication delays caused by the wireless network. Future directions of this research may include the evaluation of the *HybridTree* optical flow technique using a different optical flow formulation (especially, non-quadratic penalizers), the development of an adaptive process for adjusting some parameters in running time and the enhancement of the spatial decomposition by incorporating other features, for instance, flow fields of previous time instants or a rough estimation of the optical flow. On the performance level, it is important to study the actual impact of varying different formulations during the optical flow computation because is currently the bottleneck of the perception system. Overcoming the problem of computational requirements, optical flow techniques will be widely used in robotics. They have a remarkable reliability for perceiving motion without many assumptions that restrict the measurement of apparent motion. An interesting approach for motion analysis will be the extraction of different models using information about the angular displacement of the camera and depth. This should be straightforward but is very helpful for solving the parallax problem and will be of extremely importance for generalizing the algorithms that are presented in this thesis. Finally, the RBLT filter can be improved using the decomposition of motion that is obtained from the WOFS method. The goal is to stabilize different regions of the video and to reduce the ghost effect.

Bibliography

[1] P. Hill, C. Canagarajah, and D. Bull. Texture gradient based watershed segmentation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 3381–3384, 2002.

[2] Oisin Mac, Ahmad Humayun, Marc Pollefeys, and Gabriel Brostow. Learning a confidence measure for optical flow. *IEEE transactions on pattern analysis and machine intelligence*, 1(99):1–14, 2012.

[3] Simon Baker, Daniel Scharstein, J. Lewis, Stefan Roth, Michael Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[4] Andry Maykol Pinto, A. Paulo Moreira, Miguel Velhote Correia, and Paulo Gomes Costa. A flow-based motion perception technique for an autonomous robot system. *Journal of Intelligent and Robotic Systems*, (in press) doi: 10.1007s10846-013-9999-z:1–18, 2013.

[5] G. Varghese and Zhou Wang. Video denoising based on a spatiotemporal gaussian scale mixture model. *IEEE Trans. Cir. and Sys. for Video Technol.*, 20(7):1032–1040, 2010.

[6] Alberto Rosales-Silva, Francisco Gallegos-Funes, and Volodymyr Ponomaryov. Fuzzy directional (fd) filter for impulsive noise reduction in colour video sequences. *Journal of Visual Communication and Image Representation*, 23(1):143–149, 2012.

[7] Ethan Eade and Tom Drummond. Edge landmarks in monocular SLAM. *Image and Vision Computing*, 27(5):588–596, 2009.

[8] Saurav Kumar. Binocular Stereo Vision Based Obstacle Avoidance Algorithm for Autonomous Mobile Robots. In *IEEE International Advance Computing Conference*, pages 254–259, 2009.

[9] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[10] Miguel Fernando Paiva Velhote Correia. *Técnicas computacionais na percepcão visual do movimento*. PhD in electrical and computer engineering, Faculty of Engineering of the University of Porto, Porto, Portugal, 2001.

[11] Carol Martínez, Thomas Richardson, Peter Thomas, Jonathan Luke du Bois, and Pascual Campoy. A vision-based strategy for autonomous aerial refueling tasks. *Robotics and Autonomous Systems*, 61(8):876 – 895, 2013.

[12] Aryo Ibrahim, Pang Ching, Gerald Seet, Michael Lau, and Witold Czajewski. Moving Objects Detection and Tracking Framework for UAV-based Surveillance. In *Pacific-Rim Symposium on Image and Video Technology*, pages 456–461, 2010.

[13] S. Campbell, W. Naeem, and G.W. Irwin. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2):267 – 283, 2012.

[14] Donato Di Paola, Annalisa Milella, Grazia Cicirelli, and Arcangelo Distante. An autonomous mobile robotic system for surveillance of indoor environments. *International Journal of Advanced Robotic Systems*, 7:19–26, 2009.

[15] Antonio Fernández-Caballero, José Carlos Castillo, Javier Martínez-Cantos, and Rafael Martínez-Tomás. Optical flow or image subtraction in human detection from infrared camera on mobile robot. *Robotics and Autonomous Systems*, 58(12):1273–1281, 2010.

[16] Jaime dos Santos Cardoso. *Metadata assisted image segmentation*. PhD in electrical and computer engineering, Faculty of Engineering of the University of Porto, Porto, Portugal, 2006.

[17] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD in electrical and computer science, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009.

[18] Ninad Thakoor, Jean Gao, and Huamei Chen. Automatic object detection in video sequences with camera in motion. In *Advanced Concepts for Intelligent Vision Systems*, pages 7–14, 2004.

[19] Andry Maykol. Pinto, António P. Moreira, and Paulo G. Costa. An architecture for visual motion perception of a surveillance-based autonomous robot. In *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 205–211, 2014.

[20] Andry Maykol Pinto, Paulo G. Costa, Miguel V. Correia, and A. Paulo Moreira. Enhancing dynamic videos for surveillance and robotic applications: The robust bilateral and temporal filter. *Signal Processing: Image Communication*, 29(1):80–95, 2014.

[21] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[22] Andry Maykol Pinto, Miguel V. Correia, A. Paulo Moreira, and Paulo G. Costa. Unsupervised flow-based motion analysis for an autonomous moving system. *Image and Vision Computing*, 22(6-7):391–404, 2014.

[23] Andry Maykol Pinto, A. Paulo Moreira, Paulo G. Costa, and Miguel V. Correia. Revisiting lucas-kanade and horn-schunck. *Journal of Computer Engineering and Informatics (JCEI)*, 1(2):23–29, 2013.

[24] Andry Maykol Pinto, António P. Moreira, and Paulo G. Costa. Streaming image sequences for vision-based mobile robots. In *Portuguese Conference on Automatic Control (CONTROLO)*, pages 1–6, 2014.

[25] Junqiu Wang and R Cipolla. Image-based localization and pose recovery using scale invariant features. In *Robotics and Biomimetics, 2004.*, pages 8–12, 2004.

[26] Andry Maykol Pinto, Luís Rocha, and António Moreira. Object recognition using laser range finder and machine learning techniques. *Robotics and Computer-Integrated Manufacturing*, 29(1):12–22, 2013.

[27] R. Castle and D. Murray. Keyframe-based recognition and localization during video-rate parallel tracking and mapping. *Image and Vision Computing*, 29(8):524–532, 2011.

[28] Yong-ju Lee and Jae-bok Song. Visual SLAM in indoor environments using autonomous detection and registration of objects. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 671–676, 2008.

[29] Cabido, Montemayor, Pantrigo, Martínez-Zarzuela, and Payne. High-performance template tracking. *Journal of Visual Communication and Image Representation*, 23(2):271–286, 2012.

[30] Abhijit Kundu, K. Madhava Krishna, and C. Jawahar. Realtime multibody visual slam with a smoothly moving monocular camera. In *IEEE International Conference on Computer Vision (ICCV))*, pages 2080–2087, 2011.

[31] H. Liu, Z. Huo, and G. Yang. Omnidirectional Vision for Mobile Robot Human Body Detection and Localization. In *IEEE International Conference on Systems Man and Cybernetics (SMC)*, pages 2186–2191, 2010.

[32] R.O. Castle, G. Klein, and D.W. Murray. Combining monoSLAM with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28(11):1548–1556, 2010.

[33] Donggyu Sim and Yongmin Kim. Detection and compression of moving objects based on new panoramic image modeling. *Image and Vision Computing*, 27(10):1527 – 1539, 2009.

[34] P. Metkar Shilpa and N. Talbar Sanjay. Dynamic Motion Detection technique for fast and efficient video coding. In *IEEE Region Conference on TENCON*, pages 1–5, 2008.

[35] Gary Overett and David Austin. Stereo vision motion detection from a moving platform. In *Australasian Conference on Robotics and Automation*, pages 1–10, 2004.

[36] In Kim, Hong Choi, Kwang Yi, Jin Choi, and Seong Kong. Intelligent visual surveillance: A survey. *International Journal of Control, Automation and Systems*, 8(5):926–939, 2010.

[37] P. Spagnolo, T. Orazio, M Leo, and A. Distante. Moving object segmentation by background subtraction and temporal analysis. *Image and Vision Computing*, 24(5):411–423, 2006.

[38] Michael Harville, Gaile Gordon, and John Woodfill. Foreground segmentation using adaptive mixture models in color and depth. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 3–11, 2001.

[39] Shu-Te Su and Yung-Yaw Chen. Moving Object Segmentation Using Improved Running Gaussian Average Background Model. In *Computing: Techniques and Applications (DICTA)*, pages 24–31, 2008.

[40] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[41] J. Cheng, J. Yang, Y. Zhou, and Y. Cui. Flexible background mixture models for foreground segmentation. *Image and Vision Computing*, 24(5):473–482, 2006.

[42] Ying Ren, C. S. Chua, and Y. K. Ho. Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24(1-3):183–196, 2003.

[43] Zheng Li, Pohuang Jiang, Jian Yang Hong Ma, and D. M. Tang. A model for dynamic object segmentation with kernel density estimation based on gradient features. *Image and Vision Computing*, 27(6):817–823, 2009.

[44] Yazhou Liu, Hongxun Yao, Wen Gao, Xilin Chen, and Debin Zhao. Nonparametric background generation. *Journal of Visual Communication and Image Representation*, 18(3):253–263, 2007.

[45] X.L. Lv and G. L. Zhao. A new method for selecting gradient weight in incremental eigen-background modeling. In *Information and Automation*, pages 801–805, 2009.

[46] X. J. Cao, B. C. Pan, S. L. Zheng, and C. Y. Zhang. Motion object detection method based on piecemeal principal component analysis of dynamic background updating. In *International Conference Machine Learning and Cybernetics*, volume 5, pages 2932–2937, 2008.

[47] Yang Wang, Tele Tan, Kia-Fock Loe, and Jian-Kang Wu. A probabilistic approach for foreground and shadow segmentation in monocular image sequences. *Pattern Recognition*, 38(11):1937–1946, 2005.

[48] Xingzhi Luo, S. M. Bhandarkar, Wei Hua, and Haisong Gu. Nonparametric Background Modeling Using the CONDENSATION Algorithm. In *IEEE International Conference on Video and Signal Based Surveillance (AVSS)*, pages 3–9, 2006.

[49] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 256–252, 1999.

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

[51] Peng Suo and Yanjiang Wang. An improved adaptive background modeling algorithm based on Gaussian mixture model. In *IEEE International Conference on Signal Processing (ICSP)*, pages 1436–1439, 2008.

[52] T. Bouwmans, F. El. Baf, and B. Vachon. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Pattents on Computer Science*, 3(1):219–237, 2008.

[53] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O' Reilly Media Inc., first edition, 2008.

[54] Li Cheng and Minglun Gong. Realtime background subtraction from dynamic scenes. In *IEEE International Conference on Computer Vision*, pages 2066–2073, 2009.

[55] Yaser Sheikh and Mubarak Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[56] Wei Shuigen and Nanchang Hangkong. Motion Detection Based on Temporal Difference Method and Optical Flow field. In *International Symposium on Electronic Commerce and Security (ISECS)*, volume 2, pages 85–88, 2009.

[57] S. Murali and R. Girisha. Segmentation of Motion Objects from Surveillance Video Sequences Using Temporal Differencing Combined with Multiple Correlation. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 472–477, 2009.

[58] Marco Tagliasacchi. A genetic algorithm for optical flow estimation. *Image and Vision Computing*, 25(2):141–147, 2007.

[59] Shui-gen Wei, Lei Yang, Zhen Chen, and Zhen-feng Liu. Motion Detection Based on Optical Flow and Self-adaptive Threshold Segmentation. *Procedia Engineering*, 15:3471–3476, 2011.

[60] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[61] Encyclopedia Britannica. Biography of the Johann H. Lambert. Access date, 2012.

[62] P. O'Donovan. Optical flow: Techniques and applications. Master's thesis, University of Saskatchewan, Saskatchewan, Canada, 2005.

[63] J. L. Barron and N. A. Thacker. Tutorial: Computing 2d and 3d optical flow. Technical Report Tina Memo 2004-012, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, Stopford Building, Oxford Road, Manchester, M13 9PT, January 2005.

[64] Michael Julian Black. *Robust Incremental Optical Flow*. PhD in computer science, Yale University, Department of Computer Science, Yale, 1992.

[65] David J. Fleet and Yair Weiss. *Mathematical models for Computer Vision: The Handbook.* N. Paragios, Y. Chen, and O. Faugeras (eds.), Springer, first edition, 2005.

[66] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In Springer-Verlag, editor, *European Conference on Computer Vision*, pages 237–252, 1992.

[67] Bradley Atcheson, Wolfgang Heidrich, and Ivo Ihrke. An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in Fluids*, 46:467–476, 2009.

[68] Berthold Horn and Brian Schunck. Determining optical fow. *Artificial Intelligence*, 17(1):185–203, 1981.

[69] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *In International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[70] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433–466, 1995.

[71] J. Weber and J. Malik. Robust computation of optical flow in a multiscale differential framework. *International Journal of Computer Vision*, 14(1):67–81, 1995.

[72] M. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, Department of Computer Science, New Haven, CT, 1992. PhD thesis in computer science.

[73] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, 2010.

[74] M. Black and P. Anandan. Robust dynamic motion estimation over time. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–302, 1991.

[75] M. Black and D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[76] A. Bab-Hadiashar and D. Suter. Robust optical flow computation. *International Journal of Computer Vision*, 29(1):59–77, 1998.

[77] E. Ong and M. Spann. Robust optical flow computation based on least-median-of-squares regression. *International Journal of Computer Vision*, 31(1):51–82, 1999.

[78] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, volume 4, pages 25–36, 2004.

[79] A. Bruhn and J. Weickert. Towards ultimate motion estimation: combining highest accuracy with real-time performance. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 749–755, 2005.

[80] Henning Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, Agustín Salgado, B. Rosenhahn, and H. Seidel. Complementary optic flow. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 207–220, 2009.

[81] Li Xu, Jiaya Jia, and Yasuyuki Matsushita. Motion detail preserving optical flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1744–1757, 2012.

[82] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. Optic flow in harmony. *International Journal of Computer Vision*, 93(3):368–388, 2011.

[83] P. Golland and M. Bruckstein. Motion from color. *Computer Vision and Image Understanding*, 68(3):346–3621, 1997.

[84] S. Denman, C. Fookes, and S. Sridharan. Improved simultaneous computation of motion detection and optical flow for object tracking. In *IEEE Digital Image Computing: Techniques and Applications (DICTA)*, pages 175–182, 2009.

[85] Wendi Li and Jianqin Han. Detection of the Mobile Object with Camouflage Color Under Dynamic Background Based on Optical Flow. *Procedia Engineering*, 15:2201–2205, 2011.

[86] Naoya Ohnishi and Atsushi Imiya. Dominant plane detection from optical flow for robot navigation. *Pattern Recognition Letters*, 27(9):1009–1021, 2006.

[87] José Martín, Aitzol Zuloaga, Carlos Cuadrado, Jesús Lázaro, and Unai Bidarte. Hardware implementation of optical flow constraint equation using fpgas. *Computer Vision and Image Understanding*, 98(3):462–490, 2005.

[88] H. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London*, 208(1173):385–397, 1980.

[89] Jing Li, Hong Ai, and Jianzhu Cui. Moving Vehicle Detection in Dynamic Background from Airborne Monocular Camera. *Energy Procedia*, 13:3955–3961, 2011.

[90] Won Jin Kim and In-So Kweon. Moving Object Detection and Tracking from Moving Camera. In *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 23–26, 2011.

[91] Fernando Caballero, Luis Merino, Joaquín Ferruz, and Aníbal Ollero. Unmanned aerial vehicle localization based on monocular vision and online mosaicking. *Journal of Intelligent and Robotic Systems*, 55(4-5):323–343, 2009.

[92] S. Berrabah, G. De Cubber, V. Enescu, and H. Sahli. MRF-Based Foreground Detection in Image Sequences from a Moving Camera. In *IEEE International Conference on Image Processing*, pages 1125–1128, 2006.

[93] Rita Cucchiara, Andrea Prati, and Roberto Vezzani. Real-time motion segmentation from moving cameras. *Real-Time Imaging*, 10(3):127–143, 2004.

[94] Jean Gao, Ninad Thakoor, and Sungying Jung. A motion field reconstruction scheme for smooth boundary video object segmentation. In *International Conference on Image Processing*, volume 1, pages 381–384, 2004.

[95] Nathan Michael, Davide Scaramuzza, and Vijay Kumar. Special issue on micro-uav perception and control. *Autonomous Robots*, 33(1):1–3, 2012.

[96] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi. A vision-based guidance system for uav navigation and safe landing using natural landmarks. *Journal of Intelligent and Robotic Systems*, 57(1-4):233–257, 2010.

[97] Heiko Helble and Stephen Cameron. Oats: Oxford aerial tracking system. *Robotics and Autonomous Systems*, 55(9):661–666, 2007.

[98] Javier Traver and Alexandre Bernardino. A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems*, 58(4):378 – 398, 2010.

[99] Dar-Shyang Lee. Effective gaussian mixture learning for video background subtraction. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):827–32, 2005.

[100] Jay Hyuk Choi, Dongjin Lee, and Hyochoong Bang. Tracking an unknown moving target from UAV: Extracting and localizing an moving target with vision sensor based on optical flow. In *5th International Conference on Automation, Robotics and Applications (ICARA)*, pages 384–389, 2011.

[101] Jiman Kim, Guensu Ye, and Daijin Kim. Moving object detection under free-moving camera. In *IEEE International Conference on Image Processing (ICIP)*, pages 4669–4672, 2010.

[102] Abhijit Kundu, C. V. Jawahar, and K. Madhava Krishna. Realtime moving object detection from a freely moving monocular camera. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1635–1640, 2010.

[103] Quian Yu and Gerard Medioni. A GPU-based implementation of motion detection from a moving platform. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, 2008.

[104] Masaaki Shibata, Tomohiko Makino, and Masahide Ito. Target distance measurement based on camera moving direction estimated with optical flow. In *IEEE International Workshop on Advanced Motion Control*, pages 62–67, 2008.

[105] Ming-Yu Shih, Yao-Jen Chang, Bwo-Chau Fu, and Ching-Chun Huang. Motion-based Background Modeling for Moving Object Detection on Moving Platforms. In *International Conference on Computer Communications and Networks*, pages 1178–1182, 2007.

[106] Boyoon Jung and Gaurav S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *8th Conference on Intelligent Autonomous Systems*, pages 980–987, 2004.

[107] Juliana Wanderley and Mark H. Fisher. Spatial-feature parametric clustering applied to motion-based segmentation in camouflage. *Computer Vision and Image Understanding*, 85(2):144–157, 2002.

[108] Niloofar Gheissari, Alireza Bab-Hadiashar, and David Suter. Parametric model-based motion segmentation using surface selection criterion. *Computer Vision and Image Understanding*, 102(2):214–226, 2006.

[109] Aurelie Bugeau and Patrick Perez. Detection and segmentation of moving objects in complex scenes. *Computer Vision and Image Understanding*, 113(4):459–476, 2009.

[110] Dimitrios Alexiadis and George Sergiadis. Motion estimation, segmentation and separation, using hypercomplex phase correlation, clustering techniques and graph-based optimization. *Computer Vision and Image Understanding*, 113(2):212–234, 2009.

[111] Kai-Kuang Ma and Hai-Yun Wang. Unsupervised semantic video objects segmentation over optical-flow field. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, volume 3, pages 1216–1221, 2002.

[112] G. Georgiadis, A. Ayvaci, and S. Soatto. Actionable saliency detection: Independent motion detection without independent motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 646–653, 2012.

[113] Samuel Schulter, Christian Leistner, Peter Roth, and Horst Bischof. Unsupervised object discovery and segmentation in videos. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–12, 2013.

[114] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Lecture Notes in Computer Science Computer Vision (ECCV)*, volume 6315, pages 282–295, 2010.

[115] G. Eibl and N. Brandle. Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[116] Min Hu, S. Ali, and M. Shah. Learning motion patterns in crowded scenes using motion flow field. In *International Conference on Pattern Recognition*, pages 1–5, 2008.

[117] Andry Maykol G. Pinto, A. Paulo Moreira, and Paulo G. Costa. Indoor localization system based on artificial landmarks and monocular vision. *TELKOMNIKA*, 10(1):609 – 620, 2012.

[118] Andry M. Pinto, António P. Moreira, and Paulo G. Costa. A localization method based on map-matching and particle swarm optimization. *Journal of Intelligent and Robotic Systems*, pages 1–14, 2013.

[119] Laurent Condat. Color filter array design using random patterns with blue noise chromatic spectra. *Image and Vision Computing*, 28(8):1196–1202, 2010.

[120] Chi-Yi Tsai and Kai-Tai Song. A new edge-adaptive demosaicing algorithm for color filter arrays. *Image and Vision Computing*, 25(9):1495–1508, 2007.

[121] Jingjing Dai, C. Oscar, Feng Zou, and Chao Pang. Generalized multihypothesis motion compensated filter for grayscale and color video denoising. *Signal Processing*, 93(1):70–85, 2013.

[122] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, ICCV '98, pages 839–846, Washington, DC, USA, 1998. IEEE Computer Society.

[123] B. Weiss. Fast median and bilateral filtering. In *ACM Transactions on Graphics 25(3)*, Proceedings of the ACM SIGGRAPH, pages 519–526, New York, NY, USA, 2006.

[124] Ehsan Nadernejad, Jari Korhonen, Soren Forchhammer, and Nino Burini. Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering. *Signal Processing: Image Communication*, 28(3):222 – 240, 2013.

[125] Laurence Likforman-Sulem, Jerome Darbon, and Elisa Barney Smith. Enhancement of historical printed document images by combining total variation regularization and non-local means filtering. *Image and Vision Computing*, 29(5):351–363, 2011.

[126] V. Varghees, M. Manikandan, and R. Gini. Adaptive mri image denoising using total-variation and local noise estimation. In *International Conference on Advances in Engineering, Science and Management (ICAESM)*, pages 506–511, 2012.

[127] Ci Wang, Jun Zhou, and Shu Liu. Adaptive non-local means filter for image deblocking. *Signal Processing: Image Communication*, 28(5):522 – 530, 2013.

[128] Z. Mustafa and Y. Kadah. Multi resolution bilateral filter for mr image denoising. In *1st Middle East Conference on Biomedical Engineering (MECBME)*, pages 180–184, 2011.

[129] C. Anand and J. Sahambi. Mri denoising using bilateral filter in redundant wavelet domain. In *IEEE Region Conference TENCON 2008*, pages 1–6, 2008.

[130] Hong-Ying Yang, Xiang-Yang Wang, Tian-Xiang Qu, and Zhong-Kai Fu. Image denoising using bilateral filter and gaussian scale mixturesin shiftable complex directional pyramid domain. *Computers and Electrical Engineering*, 37(1):656–668, 2011.

[131] Zhang Ping and Chen Lihui. Document filters using morphological and geometrical features of characters. *Image and Vision Computing*, 19(12):847 – 855, 2001.

[132] Romain Lerallut, Étienne Decencière, and Fernand Meyer. Image filtering using morphological amoebas. *Image and Vision Computing*, 25(4):395 – 404, 2007.

[133] Michal Seeman, Pavel Zemčík, Roman Juránek, and Adam Herout. Fast bilateral filter for hdr imaging. *Journal of Visual Communication and Image Representation*, 23(1):12–17, 2012.

[134] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, 81(1):24–52, 2009.

[135] R. Szeliski C. Liu, W. Freeman and S. Kang. Noise estimation from a single image. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 901–908, 2006.

[136] K. Mak, P. Peng, and K. Yiu. Fabric defect detection using morphological filters. *Image and Vision Computing*, 27(10):1585 – 1592, 2009.

[137] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 1–50, New York, NY, USA, 2008.

[138] F. Porikli. Constant time o(1) bilateral filtering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[139] Qingxiong Yang, Kar-Han Tan, and N. Ahuja. Real-time o(1) bilateral filtering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 557–564, 2009.

[140] Jordi Salvador, Axel Kochale, and Siegfried Schweidler. Patch-based spatio-temporal super-resolution for video with non-rigid motion. *Signal Processing: Image Communication*, 28(5):483 – 493, 2013.

[141] Wen Li, Jun Zhang, and Qiong hai Dai. Video denoising using shape-adaptive sparse representation over similar spatio-temporal patches. *Signal Processing: Image Communication*, 26(4–5):250 – 265, 2011.

[142] Randa Atta, Rawya Rizk, and Mohammad Ghanbari. Motion-compensated {DCT} temporal filters for efficient spatio-temporal scalable video coding. *Signal Processing: Image Communication*, 24(9):702–717, 2009.

[143] A. Buades, B. Coll, and J. Morel. Denoising image sequences does not require motion estimation. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 70–74, 2005.

[144] Nawal Benmoussat, M. Faouzi Belbachir, and Beloufa Benamar. Motion estimation and compensation from noisy image sequences: A new filtering scheme. *Image and Vision Computing*, 25(5):686–694, 2007.

[145] Peter J. Huber. *Robust Statistics : An R and S Plus Companion to Applied Regression*. John Wiley and Sons, 1981.

[146] John Fox and Sanford Weisberg. *An R and S Plus Companion to Applied Regression*. Sage Publications, Inc, second edition, 2002.

[147] M. J. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, New Haven, CT, 1992. Research Report YALEU-DCS-RR-923.

[148] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[149] A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. In *International Conference on Pattern Recognition (ICPR)*, ICPR 2010, pages 2366–2369, Washington, DC, USA, 2010. IEEE Computer Society.

[150] V. Zlokolica, A. Pizurica, and W. Philips. Wavelet-domain video denoising based on reliability measures. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(8):993–1007, 2006.

[151] S. Rahman, M. Ahmad, and M. Swamy. Video denoising based on inter-frame statistical modeling of wavelet coefficients. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(2):187–198, 2007.

[152] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3-d transform-domain collaborative filtering. In *Proc. Eur. Signal Process.*, pages 145–149, 2007.

[153] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003.

[154] V. Ponomaryov, F. Gallegos-Funes, and A. Rosales-Silva. Real-time color imaging based on rm-filters for impulsive noise reduction. *Journal Imaging Science and Technology*, 49(3):205–219, 2005.

[155] K. Plataniotis, D. Androutsos, S. Vinayagamoorthy, and A. Venetsanopoulos. Color image processing using adaptive multichannel filters. *IEEE Transactions on Image Processing*, 6(7):933–949, 1997.

[156] Volodymyr Ponomaryov, Alberto Rosales-Silva, Francisco Funes, J. Gallegos, and Igor Loboda. Adaptive vector directional filters to process multichannel images. *IEICE Transactions*, 90-B(2):429–430, 2007.

[157] V. Zlokolica, W. Philips, and D. Van. A new non-linear filter for video processing. In *Third IEEE Benelux Signal Processing Symposium (SPS- 2002)*, pages 221–224, 2002.

[158] Shui-gen Wei, Lei Yang, Zhen Chen, and Zhen-Feng Liu. Motion detection based on optical flow and self-adaptive threshold segmentation. *Procedia Engineering*, 15:3471–3476, 2011.

[159] John Barron and Reinhard Klette. Quantitative color optical flow. In *International Conference on Pattern Recognition 4(1)*, pages 251–255, 2002.

[160] Carlo Ciliberto, Ugo Pattacini, Lorenzo Natale, Francesco Nori, and Giorgio Metta. Reexamining lucas-kanade method for real-time independent motion detection: Application to the icub humanoid robot. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 4154–4160, 2011.

[161] S. Kim, Hong Choi, Kwang Yi, Jin Choi, and Seong Kong. Intelligent visual surveillance - a survey. *International Journal of Control, Automation and Systems*, 8(5):926–939, 2010.

[162] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[163] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2(1):54–61, 2003.

[164] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[165] D. Martin, CC Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):530–549, 2004.

[166] J.M. Laferté, P. Pérez, and F. Heitz. Discrete markov image modeling and inference on the quadtree. *IEEE Transactions on Image Processing*, 9(3):390–404, 2000.

[167] Michael Lightstone and Sanjit K. Mitra. Quadtree optimization for image and video coding. *The Journal of VLSI Signal Processing*, 17(2):215–224, 1997.

[168] S. Oudin, P. Helle, J. Stegemann, C. Bartnik, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand. Block merging for quadtree-based video coding. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 15, pages 1–6, 2011.

[169] M. Schroders and R. Gulik. Quadtree relief mapping. In *Proceedings of the 21st ACM SIGGRAPH EUROGRAPHICS symposium on Graphics hardware*, pages 61–66, 2006.

[170] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 2006.

[171] Dimitrios Ververidis and Constantine Kotropoulos. Information loss of the mahalanobis distance in high dimensions: Application to feature selection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2275–2281, 2009.

[172] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O' Reilly Media Inc., Beijing [u.a.], first edition edition, 2008.

[173] David Fleet and Yair Weiss. *Mathematical models for Computer Vision: The Handbook*. N. Paragios, Y. Chen, and O. Faugeras (eds.). Springer-Verlag New York, Inc., Secaucus, NJ, USA, first edition edition, 2005.

[174] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[175] M. Guzel and R. Bicker. Optical flow based system design for mobile robots. In *IEEE Conference on Robotics Automation and Mechatronics (RAM)*, pages 545–550, 2010.

[176] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[177] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[178] E. Hannan and B. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):190–195, 1979.

[179] Gilles Celeux and Gilda Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification*, 13(2):195–212, 1996.

[180] H. Bozdogan. Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-fisher information matrix. In *Information and Classification: Studies in Classification, Data Analysis and Knowledge Organization*, pages 40–54, 1993.

[181] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2st edition, 2005.

[182] C. Fraley and A. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.

[183] Douglas Reynolds, Stan Z. Li, and Anil K. Jain. *Gaussian Mixture Models*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[184] Yu-Ren Lai, Kuo-Liang Chung, Guei-Yin Lin, and Chyou-Hwa Chen. Gaussian mixture modeling of histograms for contrast enhancement. *Expert Systems with Applications*, 39(8):6720–6728, 2012.

[185] Martin Ester, Hans peter Kriegel, Jörg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, pages 226–231. AAAI Press, 1996.

[186] Hua Jiang, Jing Li, Shenghe Yi, Xiangyang Wang, and Xin Hu. A new hybrid method based on partitioning-based {DBSCAN} and ant clustering. *Expert Systems with Applications*, 38(8):9373–9381, 2011.

[187] Damodar Reddy Edla and Prasanta K. Jana. A prototype-based modified dbscan for gene clustering. *Procedia Technology*, 6(0):485–492, 2012.

[188] Dan Melamed, Ryan Green, and Joseph P. Turian. Precision and recall of machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL*, pages 61–63, 2003.

[189] G. Kootstra and D. Kragic. Fast and bottom-up object detection, segmentation, and evaluation using gestalt principles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3423–3428, 2011.

[190] Pir Qureshi and Nasrullah Memon. Hybrid model of content extraction. *Journal of Computer and System Sciences*, 78(4):1248–1257, 2012.

[191] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, 1999.