ROBUST ANALYSIS AND PROTECTION OF DYNAMIC SCENES FOR PRIVACY-AWARE VIDEO SURVEILLANCE

Hatem Abd Ellatif FatahAllah Ibrahim Mahmoud Rashwan

Dipòsit Legal: T 1102-2014

Robust Analysis and Protection of Dynamic Scenes for Privacy-Aware Video Surveillance

PhD Dissertation

Author:
Hatem Abd Ellatif FatahAllah Ibrahim Mahmoud Rashwan
(Hatem A. Rashwan)

Advisors:
Dr. Domènec Puig Valls
Dr. Antoni Martínez-Ballesté

Departament d'Enginyeria Informàtica i Matemàtiques

Universitat Rovira i Virgili

2014

UNIVERSITAT ROVIRA I VIRGILI

DEPARTAMENT D'ENGINYERIA INFORMÀTICA I MATEMÀTIQUES

I STATE that the present study, entitled Robust Analysis and Protection of Dynamic Scenes for Privacy-Aware Video Surveillance, presented by Hatem Abd Ellatif FatahAllah Ibrahim Mahmoud Rashwan for the award of the degree of Doctor, has been carried out under my supervision at the Department of Chemical Engineering of this university, and that it fulfils all the requirements to be eligible for the international doctorate award.

Tarragona, 25July 2013

Doctoral Thesis Supervisors

Dr. Domènec Puig Valls

Dr. Antoni Martínez-Ballesté

2

To my wife Reham and my son Ahmed

To my mother and my father

2

Abstract

Recent advances in pervasive video surveillance systems pave the way for a comprehensive surveillance of every aspect of our lives. Computerized and interconnected camera systems can be used to profile, track and monitor individuals for the sake of security. Notwithstanding, these systems clearly interfere with the fundamental right of the individuals to privacy. To alleviate this privacy problem and avert the so-called Big Brother effect, the usage of privacy enhancing technologies is mandatory.

Privacy-aware video surveillance systems are based on a Detection Submodule that detects the so-called regions of interest (i.e. areas to protect to achieve privacy) from the captured video and on a Protection Submodule that protects the detected areas (aiming at preventing identity disclosure). Only a trusted manager might be able to access the protected video and unprotect it, for instance in case of criminal investigations and, in general, under permission of a law enforcer (judge, police, etc.). Most literature on privacy in video surveillance systems concentrates on the goal of detecting faces and other regions of interest, and in proposing different methods to protect them. However, the trustworthiness of those systems and, by extension the privacy they provide, is neglected.

In this thesis, the topic of privacy-aware video surveillance is tackled from a holistic point of view. Firstly, an introductory chapter defines the properties of a trustworthy privacy-aware video surveillance system, and reviews the techniques that can be used in the Detection Submodule and in the Protection Submodule. The remaining of the thesis is divided into two parts. In the first one, some contributions aiming at improving the detection of regions of interest are developed. Specifically, it addresses our contributions to optical flow detection techniques: it has been found that, despite its usefulness, the widely known variational optical flow has several limitations and shortcomings for providing accurate flow fields for motion estimation problems in computer vision. In order to overcome these limitations, new development models are introduced as an alternative to classic concepts. Two models are proposed in this dissertation in order to improve the robustness of variational optical flow model through tensor voting to be more robust against noise and to preserve discontinuities. In addition, the data term of the optical flow model based on brightness constancy assumption is replaced by a rich descriptor in order to obtain an illumination-robust optical flow model.

In the second part, the protection of regions of interest is addressed. A method based on coefficient alteration in the compressed domain of the video is presented and tested in terms of robustness and efficiency. The processes related to the information security of the data involved in the protection and unprotection processes are also comprehensively taken into account.

The thesis includes tests and implementations for all the theoretical proposals, aiming at demonstrating their validity in a real video surveillance scenario.

Finally, a chapter with a summary of the advances presented and further work concludes the thesis.

Keywords: Video surveillance systems, object detection, privacy enhancement, motion estimation, optical flow, tensor voting, histogram of gradients, coefficients alteration, random alteration attack.

Resumen

Los recientes avances en los sistemas de vigilancia de vídeo generalizados allanan el camino para una vigilancia exhaustiva de todos los aspectos de nuestras vidas. Sistemas de cámaras computerizadas e interconectadas se pueden utilizar para perfilar, rastrear y monitorizar los individuos por el bien de la seguridad. No obstante, estos sistemas interfieren claramente con el derecho fundamental de los individuos a la privacidad. Para aliviar este problema de privacidad y evitar el denominado efecto de Gran Hermano, el uso de tecnologías potenciadoras de la privacidad es obligatoria.

Los sistemas de videovigilancia respetuosos de la intimidad se basan en un submódulo de detección, que detecta las llamadas regiones de interés (es decir, las zonas a proteger) a partir del vídeo capturado y en un submódulo de protección, que protege las áreas detectadas. Sólo un administrador de confianza podría ser capaz de acceder al vídeos protegido y desprotegerlo, por ejemplo en el caso de investigaciones penales y, en general, con el permiso de un agente de la ley (jueces, policía, etc.). La mayoría de la literatura sobre la privacidad en los sistemas de vigilancia de vídeo se centra en el objetivo de la detección de rostros y otras regiones de interés, y en proponer diferentes métodos para protegerlos. Sin embargo, la fiabilidad de los sistemas y, por extensión, de la privacidad que proporcionan se descuidan.

En esta tesis, el tema de la videovigilancia respetando la privacidad se aborda desde un punto de vista holístico. En primer lugar, un capítulo introductorio define las propiedades de un sistema de videovigilancia confiable respetuoso de la privacidad, y se revisan las técnicas que se pueden utilizar en el submódulo de detección y en el submódulo de protección. El resto de la tesis se divide en dos partes. En la primera de ellas, se desarrollan algunas de las contribuciones destinadas a mejorar la detección de las regiones de interés. Específicamente, se ocupa de nuestras contribuciones a las técnicas de detección de flujo óptico: se ha encontrado que, a pesar de su utilidad, el flujo óptico variacional, ámpliamente conocido, tiene varias limitaciones y deficiencias para proporcionar campos de flujo precisos para problemas de estimación de movimiento en la visión por computador. Con el fin de superar estas limitaciones, nuevos modelos de desarrollo se introducen como una alternativa a los conceptos clásicos. Se proponen dos modelos con el fin de mejorar la robustez del modelo de flujo óptico variacional a través del votoción tensorial para ser más robusto frente al ruido y preservar discontinuidades. Además, el término datos del modelo de flujo óptico basado en la suposición de la constancia de brillo se sustituye por un descriptor rico con el fin de obtener un modelo de flujo óptico - iluminación robusta.

En la segunda parte, se trata la protección de las regiones de interés. Un

método basado en la alteración de coeficientes en el dominio comprimido se ha implementado y testeado en términos de robustez y eficiencia. Los procesos relacionados con la seguridad de la información de los datos que intervienen en los procesos de protección y desprotección también se ha considerado.

La tesis incluye pruebas e implementaciones para todas las propuestas teóricas, con el objetivo de demostrar su validez en un escenario de videovigilancia real.

Por último, la tesis concluye con una recapitulación de los avances presentados y con la propuesta de trabajos futuros.

Acknowledgements

Firstly, I would like to thank my advisors for their supervision: Dr. Domènec Puig, in the Intelligent Robotics and Computer Vision Group (IRCV), and Dr. Antoni Martínez-Ballesté in CRISES research group at Rovira i Virgili University, Tarragona, Spain, not only for their great guidance and fruitful discussions but also for their confidence in giving me the opportunity to carry on this thesis.

Secondly, Dr. Miguel Ángel García at the Autonomous University of Madrid for his guidance and his discussion during three years of work in this thesis. In addition, I would like the express my gratitude to Prof. Joachim Weickert for giving me the opportunity to work at the Mathematical Image Analysis Group (MIA) during my stay at Saarland University, Saarbrucken, Germany. I would also like to express my special gratitude to all, former and current members IRCV group: Carme, Jaime, Juan, Julian, Julio, Ling, Marcela, Mohamed, Rodrigo, Said, Tomás and Xavi, as well as the members of the MIA group in Saarland University: David, Oliver and Yan. Also, I want to give many thanks for all my friends in URV. Furthermore, I want to thank my friend and colleague Mahmoud in Paderborn University Germany.

In addition, I should express the deepest and warmers gratitude to my wonderful lovely wife Reham, my son Ahmed, my brothers and my sisters. Also, special thanks to my mother for supporting and praying for me. I am very grateful for the invaluable support received from innumerable friends at URV and elsewhere, Hany, Gorg, Abdali, Yousif and Hamdi.

In addition, this thesis would not have reached completion without the financial support from various institutions. Primarily, from the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) through the scholarships FI/DGR-2010. These scholarships have been partially supported by the Commissioner for Universities and Research of the Department of Innovation, Universities and Companies of the Catalonia Government and by the European Social Fund. In addition, I have received a financial support from Rovira i Virgili University through project 2012R2B-01 VIPP.

Last but not least, I want to thank all those anonymous reviewers who have made worthy suggestions on my conference and journal publications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Only two men live in this life: a science-speaking scientist and a conscious learner." - Prophet Muhammad

In the last years enormous advances of Information and Communication Technologies (ICT) have paved the way for the consolidation of a growing Information Society. Millions of users continuously upload tons of information (pictures, videos, opinions, etc.), using a variety of devices. That information is stored in multiple interconnected servers that are remotely accessible from almost everywhere. In addition, computer scientists have developed techniques for information gathering and analysis that allow the generation of huge amounts of knowledge.

In the last decade, we have witnessed an unprecedented increase of citizen data acquisition: search engines, medical systems, social networks, etc. collect vast amounts of data. In addition, video cameras can be found almost everywhere: from city-scale surveillance systems controlled by local authorities, to simple and cheap private systems in restaurants and shops. As a result, people are being monitored and recorded while doing some of their everyday activities: having lunch at the restaurant, leaving a parking, entering a companys building, using a bus, shopping in the supermarket, etc.

Video Surveillance Systems (VSS) have significantly evolved from simple CCTV monitored by authorized people to complex and interconnected pervasive video cameras, whose recorded materials are streamed, processed and mined so as to extract information and knowledge. Pervasive VSS inherently endanger the privacy of people due to the fact that their identities and activities could be easily retrieved from pictures and videos. Computerized and interconnected camera systems can be used to profile, track and monitor individuals for the sake of security.

Despite all the clear advantages of ICT, pervasive computing and the massive connection of ubiquitous computing devices (computers, smartphones, RFID readers , video cameras, etc.) may transform Information Society into the so-called

Dataveillance Society (Clarke, 2001) thus violating the fundamental right to privacy as stated in the Universal Declaration of Human Rights[1], "No one shall be subjected to arbitrary interference with his privacy". In fact, people profiling favors the so-called "Big Brother" effect. Regarding video surveillance, people might sacrifice part of their privacy for the sake of security[2]. However, most people dislike being monitored during their daily activities.

The rest of the chapter is organized as follows. The main aspects of the trustworthy in privacy-awareness VSS are introduced in Section 1.1. In addition, the research directions and objectives of this thesis are introduced in Section 1.2 and Section 1.3, respectively. Finally, the descriptions of the next chapters are shown in Section 1.4.

1.1 Privacy-awareness in video surveillance systems

Certainly, computerized video surveillance has become another main source for data collection. Moreover, the connection of these surveillance systems to the Internet allows the rapid spread of recorded videos, either because of administrators misbehaviors or because of attacks. According to legislations, pictures and videos where individuals can be recognized are considered personal data and, hence, the improper managing of the videos clearly jeopardises the privacy of the citizens.

In order to guarantee the protection of users rights, governments have been enacting legislations aiming at regulating video surveillance. The linchpin of these legislations is that people must trust the operators of VSS. In a nutshell, legislations require that the name of the operators must be clearly identified under the presence of a VSS. They also state that the videos must be destroyed after a one-month period and cannot be released to third parties, except in case of investigations.

Trust can be defined as "the degree to which a trustor has a justifiable belief that the trustee will provide the expected function or service"[3]. However, assessing the trust of current VSS, due to the pervasive nature of current VSS and the variety of operators, is not straightforward.

In principle, citizens *should* feel comfortable with trusting law enforcers that

[1]The universal declaration of human rights. [Online] available: http://www.un.org/en/documents/udhr/.

[2]Directive 95/46/EC of the European Parliament and of the council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. [Online] avaiable: http://europa.eu/legislation_summaries/information_society/data_protection/l14012_en.htm.

[3]The trusted computing group. [Online] available: http://www.trustedcomputinggroup.org/

operate city-scale CCTV systems. In contrast, they may be concerned of the trust with private security SMEs (specifically in their personnel) and individuals such as shop owners, waiters, etc. that have a full access to cameras and recordings. In that sense, people can only conceal their right to privacy by trusting that the operator behaves according to the law.

Hence, it is essential to provide a framework willing at defining and materializing the concept of *trustworthy privacy* for video surveillance systems.

Prior to defining the concept, a model that will be used throughout this thesis. Figure 1.1 presents an example of a possible video surveillance scenario. We can observe a corridor in which two cameras record digital video and perform some pre-processing (*e.g.* decrease frame rate, lossy compression of video). This video is handled by a *Video Processing Module* that consists of two sub-modules:

- The *Detection Submodule*, that utilizes some computer vision procedures to detect the Regions of Interest (ROIs) in the *original video*. This submodule outputs a list of ROIs found in each frame.
- The *Protection Submodule*, that obfuscates the detected ROIs in order to preserve the privacy of the identified people. Hence, this submodule outputs the *protected video*.

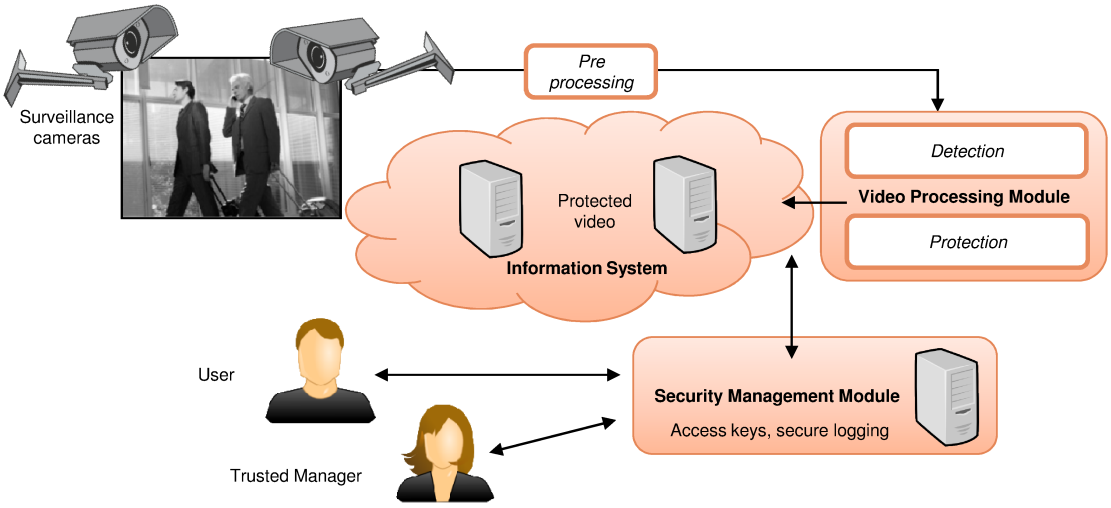


Figure 1.1: Example of a typical video surveillance scenario.

Firstly, an input video from the sensor (camera) needs to be preprocessed before sending to a video processing module stage. The composite video signal from the CCTV camera is digitized with a video capture board into a time series of raw RGB images. Each RGB color image is then converted into an alternative color

representation, such as YUV representation, which is typically enabling transmission errors or compression artifacts to be more efficiently masked by the human perception than using a direct RGB representation. Also, RGB image can be converted into HSI or HSV representation, which is more robust against the illumination change, lights change, shading or shadow in the input image sequences than RGB color mode.

Sequently, the video stream is analyzed to find ROIs, which are for example faces or car plates. ROIs are tracked in time into records, corresponding to a single object (*e.g.* person, car). These records are analyzed to determine the identity of the object (*i.e.* face recognition). Moreover, behaviors can be analyzed (*e.g.* action, gait recognition), to generate alerts upon certain conditions. In the literature, there are many algorithms used for ROIs detection that reflects object identification, such as face detection algorithms (Rowley et al., 1998, Viola and Jones, 2001, Zhang and Zhang, 2010). However, it must be stated that if a VSS considers the faces as ROIs, some identity disclosure could be done by merely analyzing clothes or via gait recognition processes. Therefore, motion detection algorithms based on background subtraction techniques, such as in (Stauffer and Grimson, 1999, Elgammal et al., 2000, Kim et al., 2004), or based on optical flow estimation approaches, such as (Lucas and Kanade, 1981, Horn and Schunck, 1981, Farneback, 2000, Bruhn et al., 2006), are other options for a robust ROIs detection. In addition, many robust techniques have been used to track objects in a scene, such as Yue et al. (2009) that presented a wide survey for the state-of-the-art of object tracking methods.

In turn, the protected video, provided by the Protection sub-module, is sent to an *Information System*: a set of computers capable of storing and controlling the access to the data. We assume that the system stores a compressed video (*e.g.* MPEG-2, H.264, etc.) instead of a set of raw uncompressed frames. Finally, a *Security Management Module* controls that only authorized users (*e.g.* a trusted manager) access the unprotected data (*i.e.* without obfuscated/obscured faces). In this scenario, the individuals privacy is protected by means of blurring their faces or bodies and controlling the access to the unprotected video. However, a number of privacy concerns might still remain: (i) Do people know that they are being recorded? (ii) Is the access to the data correctly managed? (iii) Does the performance of the video processing techniques ensure privacy?

Related literature is mainly devoted to computer vision algorithms whose goal is to detect and obscure ROIs of an image (*cf.* Senior (2009) for a comprehensive overview of privacy-aware video surveillance proposals). Several articles describe how to detect people, how to blur their faces (usually by scrambling or encrypting them) and how authorized users should gain access the original data (usually by means of secret keys). Notwithstanding, as stated in Winkler and Rinner (2010),

researchers must go one step beyond when tackling the problem of privacy in VSS, and provide manufacturers and operators of VSS with tools to build trustworthy systems.

In our model, for attack protection, the goal of an attacker is to retrieve the original or unprotected video from the system, with the purpose of disclosing the identity of the individuals. Therefore, we assume that the Information System and the Security Management Module fulfill with standard information security compliances (*e.g.* authentication, confidentiality, etc.) with regards to the information stored and the actors involved in the VSS. Fortunately, these aspects are solved with well-known cryptographic techniques and protocols. Last but not least, the trusted manager may need the permission of a law enforcer to effectively unprotect the video in case of investigations. Hence, the problem of a trusted manager arbitrarily unprotecting videos is avoided.

1.1.1 The Three Aspects of Trustworthy ICT

Thousands of users surf the Internet and enter a wide variety of sites that offer free services (*e.g.* e-mail, music, videos, newspapers etc.). What makes these services interesting to most people is their price – they are free and widely accessible through the Internet. However, the linchpin of their success is the *trust* of users in the service[4]. Thus, a VSS is trustworthy, if only stores the protected version of the video and does not require human supervision.

We distinguish three fundamental pillars sustaining trust in ICT (Martínez-Ballesté et al., 2012): (i) trust in technology, (ii) law enforcement and (iii) user collaboration:

- **Trust in technology.** Users know that technology, if properly used, is reliable. But behind the facade of reliability, technology can hide inconsistent performance and behavior – can users be sure that technology behave as it should? (*e.g.* users may install software that, in addition to its desired functionality, sends passwords to remote servers.) In this regard, non-expert users trust technology but they are not fully aware of its real behavior.

- **Law enforcement.** Technology evolves fast but legislation adapts slowly (Elizondo et al., 2012). The definition of laws covering ICT services and their actual enforcement is somehow arduous. A lack of legislation can potentially slow down the social acceptance of ICT and ICT-based services. To mitigate this effect, widespread technologies must appear along with the laws that guarantee the protection of users rights.

[4]Notice that there are many sites offering the same free services, and clients/users decide on using one or another based on factors such as trust

- **User collaboration.** Since the dawn of the collaborative web (Web 2.0) made of users contributions, the concept of collaborative trust has gained importance. For instance, recommendation systems use collaborative filtering (Balabanović and Shoham, 1997), and other services such as location-based services (Solanas and Martínez-Ballesté, 2008) and user profiling (Viejo and Castellà-Roca, 2010) rely on distributed peers trust. Thus, it appears to be natural to seriously consider user collaboration in the design of ICT to ease the management of trust.

In this thesis, we focus on the first aspect of trustworthy privacy in video surveillance (*i.e.* trust in technology). Certainly, privacy-aware systems may be based on tamper proof devices (*e.g.* smart cards or Trusted Platform Module chips[5] as proposed in Winkler and Rinner (2010), where the authors utilize a trustworthy camera to deploy their proposal. However, trusting a video surveillance device is not straightforward (*e.g.* someone may hack the camera firmware so as to send ROIs to a remote server prior to their encryption). Consequently, users might trust that technology behaves correctly (*e.g.*, detecting ROIs and protecting them as expected, granting access to authorized people only, etc).

In order to be trustworthy the technology used in the submodules must fulfill the next properties:

- *Real time performance.* The procedures used in the Detection and Protection submodules must work in real time. Otherwise since, some portions of the original video should be temporarily stored, a security leak could compromise the privacy of the individuals.

- *High accuracy.* The techniques used in the Detection Submodule must detect correctly all ROIs. If the technique fails to detect them, the system will not protect the identity of some individuals. Moreover, the process may need to be supervised by humans and, as a result, this could lead to a lack of privacy.

- *Utility.* The techniques used in the Protection Submodule must protect the ROIs in a reversible manner. Hence, disclosing the identity of the individuals in the video (for instance, under petition of law enforcers) should be done by applying some technique over the protected video. If so, there is no need to store a copy of the original video to be shown upon requests by trusted managers.

Last but not least, trustworthy systems must securely manage the information. Currently, properties such as *confidentiality* (*e.g.* encryption of the video bitstream during transmission), *authenticity* and *integrity* (*e.g.* fast digital signature of the

[5]The trusted computing group. [Online] available: http://www.trustedcomputinggroup.org/

protected video) can be fulfilled by means of state-of-the-art cryptographic techniques and well-known protocols. Moreover, the access to the data stored in the information system must be controlled by a Security Management Module and, in addition, every access to the system must be registered by secure logging techniques. Finally, users identity (specially the identity of the Trusted Manager) could be guaranteed by means of smart cards and biometric techniques.

1.2 Research directions

Without the shadow of a doubt, there is a vast literature on privacy in video surveillance. Based on the revision of the literature, this thesis addresses two different stages to enable privacy in VSS: one with regarding to ROIs detection and another with respect to ROIs protection.

Regarding ROIs detection, using face detection algorithms instead of motion detection algorithms (*i.e.* the ROI is any moving object in the scene) poses some constraints on the efficiency and accuracy of the system. Moreover, using simple motion detection algorithms, such as background subtraction models, may fail in scenes with moving backgrounds or dynamic textures such as rain or leaves. In order to cope with this problem, the use of the motion detection based on optical flow techniques is highly recommended. However, a wide variety of optical flow approaches have been proposed during the last years achieving outstanding levels of accuracy under ideal conditions[6]. In addition, most of these techniques are based on two main assumptions: brightness and gradient constancy. Both constancy assumptions respectively depend on the brightness (brightness constancy assumption) and the derivative of the brightness (gradient constancy assumption) of the pixels contained in a given pair of images. However, the brightness of a point on an object can dramatically change if the object moves to another part of the scene with different illumination or after global or local illumination changes. Furthermore, the two assumptions are sensitive to noise with its various types. Therefore, this dissertation proposes a development for optical flow estimation methods based on tensor voting in order to solve the effect of noise on estimating the motion vectors. In turn, the extracted features based on histogram of gradients (HOG) are used instead of the brightness of images to generate an illumination-robust optical flow model to cope with illumination changes.

With respect to ROIs protection, the approaches must be invertible methods in order to retrieve the original data under law enforcement authorities without needing to store the original video. In addition, the computational complexity of the protection method must be low to avoid system overloading. Hence, the

[6]Middlebury datasets, http://vision.middlebury.edu/flow/data/

protection methods used during the pixel-domain can not properly retrieve the complete clear data in a case of need, due to some data and details will be lost during the decoding process. As result, the protection approaches applied during the compression-domain are more practical than the pixel-domain based methods, because the clear original data can be properly got back. Therefore, we have observed that VSS that protect ROIs in the pixel-domain do not offer high trust and these systems must keep an unprotected version of the video that could be exposed in case of attacks yielding security leaks. In turn, regarding the protection during the compression-domain, there is no proposal that entirely takes into account all the aspects needed to asses the trust on the surveillance system, as pointed out in Chapter 2. In addition, we will concern in this thesis about the definition of protection streams, which are required for both protecting and unprotecting a video sequence.

1.3 Objectives

Bearing in mind the previous discussion about the privacy-aware surveillance systems, both ROIs detection and ROIs protection in computer vision and cryptography fields and according to our definition of VSS, the main contributions of this thesis can be summarized in three main topics:

1. Assess the state of the art on optical flow methods with regarding to the factors and conditions that influence on the estimation of accurate flow fields. In addition, one of the main contributions of this thesis is the proposal of an optical flow model used in ROIs detection that is more robust against noise than the state-of-the-art approaches. As well as, it can cope with different environmental factors such as illumination and shadow changes.

2. Propose a practical ROIs protection model during the compression-domain based on content protection, not only by using convenient cryptographic techniques, but also law enforcement and user cooperation in order to get feedback with regard to the whole VSS.

3. Build a comprehensive trustworthy VSS that can be used for practical and different purposes.

1.4 Chapter descriptions

Chapter 2 presents a brief survey for the techniques that could be used in a privacy-aware VSS. These techniques are discussed with regards to the fulfilment of the

properties that a trustworthy system must achieve (accuracy, real-time performance and utility of the protected video).

The reminder of this thesis is distributed in two parts. On the one hand, Part 1 is dedicated to the description of the proposed robust optical flow estimation. On the other hand, Part 2 defines a new trustworthy VSS.

Part 1 is organized as follows:

Chapter 3 proposes a robust optical flow model that combines a local and a global optical flow method based on an adaptation of the approaches described in Bruhn et al. (2005) and more recently in Zimmer et al. (2009), by replacing the isotropic Gaussian filtering based on the structure tensor by a discontinuity-preserving filtering stage based on tensor voting introduced in Medioni et al. (2000) in order to get more robust and accurate flow fields.

Chapter 4 presents a robust algorithm for estimating accurate flow fields. The proposed algorithm consists of replacing the discontinuity-preserving filtering stage based on tensor voting, previously proposed in Chapter 3, by a similar stage exclusively based on stick tensor voting in order to reduce the computational cost. An additional weighted non-local term based on stick tensor saliency is introduced, similarly to the one proposed in Sun et al. (2010a) in order increase the robustness of the resulting flow fields.

Chapter 5 proposes the replacement of the classical brightness constancy assumption by a local texture descriptor that is highly invariant to illumination changes. In particular, the Histogram of Oriented Gradients (HOG) Dalal and Triggs (2005) is proposed as a texture descriptor in order to extract texture features from two consecutive images.

In turn, Part 2 is organized as follows:

Chapter 6 presents a platform for trustworthy storage of privacy-aware surveillance videos. In this platform, the data needed to protect and unprotect the video are created and stored in a secure manner, using well-known cryptographic functions. The protection is done using a variation of the coefficient alteration method proposed by Dufaux and Ebrahimi (2008). Using this method, the property of utility is achieved and, at the same time, the size of the compressed video is not increased due to the protection process.

Chapter 7 proposes an algorithm to unprotect ROIs (faces) in a protected frame by assuming a previous knowledge about the faces protected, by having access to a public database of facial images. These faces have been protected with the algorithm proposed in Martínez-Ballesté and Rashwan (2013). The algorithm generates a random set of attacked images based on generating set of random streams in order to break the protection of AC coefficients of the protected faces in the current scene with a fixed DC coefficient value. The Eigenfaces algorithm proposed in Turk and Pentland (1991) is used to measure a similarity score for

each face in the generated images in order to select the best facial images. In addition, we describe a method to correct DC coefficient values of the attacked regions to reconstruct a complete attacked face.

Finally, Chapter 8 summarizes the contributions of this work and proposes future research directions and applications of the new concepts introduced in this thesis.

Chapter 2

Background on Techniques for Privacy-Aware Video Surveillance

In order to have trustworthy systems, the surveillance systems should be deeply studied to know the factors that can cause a rise or fall for the system. Thus by considering the great importance of surveillance cameras, some related work, Winkler and Rinner (2010), used a trustworthy camera in the surveillance system, but it is not a sufficient condition to get a trusted surveillance system. As result, individual video surveillance cameras studies may not be reliable on their own. However, system administrators and users must also trust in system technologies that behave correctly (*e.g.* detecting moving objects and protecting them, allowing access to the original video under authorized legalization, etc.) and there must not be any interaction between the system operators and the system to be completely an automatic trustworthy surveillance system.

Measuring the trust of video surveillance systems (VSS) is complex. In evaluating the merits of video surveillance, it is important to look at the overall trend of multiple studies and place particular reliance on studies with rigorous methodology. Thus, this chapter presents a survey for one of the most important factors used to increase the trust in surveillance systems that is the trust in the surveillance system technology.

Indeed, there are a few constraints on technology taken into account when implementing a trustworthy VSS keeping privacy, such as the computer vision algorithms whose goal is to detect and track ROIs in the video scene and the confidential protection scheme for ROIs detected (data utility). In other words, a trusted automatic VSS requires a robust trusted video analysis technique to extract (ROIs) in the input video and trusted protection schemes to obscure them. Because, any fail in ROIs detection and protection leads to a lack of protecting the video stream, or hacking during observing, saving and sending over system

networks.

In this chapter, we aim at dealing with the trustworthy in the surveillance systems from the standpoint of the technology used in different levels. That is firstly done by evaluating whether ROIs detection approaches work properly or not (*i.e.* algorithms perform well and in real-time). In addition, we assess protection schemes that can be used for privacy-sensitive data (*i.e.* the complexity of algorithms computations and the plain data can be easily retrieved or not) in order to choose the best algorithms of ROIs detection and protection for implementing a trusted video surveillance framework, which satisfies the observers of the system and the observed by the system.

The rest of the chapter is organized as follows. Section 2.1 presents a brief review of the common methods used for ROIs detection, including two face detection algorithms in Section 2.1.1, and seven motion detection algorithms (three background subtraction and four optical flow estimation models) in 2.1.2. In addition, in Section 2.1, an evaluation of the all tested algorithms is presented. Furthermore, serval ROIs protection methods are classified into two groups: pixels domain and compression domain in Section 2.2. Finally, the existing proposals in the previous sections is discussed in Section 2.3.

2.1 Techniques for detection

The ROI is a particular region in a scene in which we are interested. Therefore, it is essential to extract that region from the scene which has significant information. In order to extract significant region there need to determine its cognitive boundary. The selection of this cognitive boundary by human itself is difficult. This is because humans have different psychology of interest and decision making criteria. Then how will we define such boundary autonomously? What things are to be included and what things are to be excluded from this boundary. For video surveillance, ROIs should be detected that reflects object identification such as face, gait, skin color and other regions, which people usually aim to conceal them to be free in their daily activities.

In the literature, there are several methods proposed to automatically detect ROIs in an image or a video. Thus, the state-of-the-art techniques used in VSS are analyzed. If the ROI detection process does not work reliably, there is a risk for privacy leaks. Even if, the module fails in a single frame, the privacy is broken for the entire sequence. The confidence and trust in a surveillance system depend on the performance and accuracy of the ROI detection technique used.

We consider two trends of application in video surveillance: first, we describe face detection methods assuming that ROIs are faces, which are considered the main key for human; second, we describe other ROI detection techniques that can

be applied to more general scenarios, in which ROIs might be any moving object in the scene.

In order to be trustworthy, the *Detection* sub-module in Figure 1.1 must fulfill the aforementioned properties (*Real Time Performance* and *High accuracy*)

2.1.1 Face detection

Recently, face detection and recognition have attracted much attention of the computer vision researchers. Many research demonstrations and applications have been implemented for these efforts. A first step of any face processing system is detecting the locations in images where faces are present. However, face detection from a single image is a challenging task because of variability in scale, location, orientation (up-right, rotated), and pose (frontal, profile). Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. For an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face. The challenges associated with face detection can be attributed to many factors, such as pose, presence, facial expression, occlusion, image orientation and imaging conditions.

Face is the key of human identification, therefore here are many techniques enabling privacy concerning with face obscuration to protect individuals in the scenes. Its immediate application is automated people recognition and, although identification can be performed based on other factors (such as clothing or gait), the protection of a face is sufficient and widely accepted for a privacy protection as proposed in Dufaux (2006).

Most of the face detection algorithms consider a face detection as a *feature pattern-classification problem*. The content of a given patch of an image is transformed into special features, after which a classifier trained on example faces decides either that particular region is a face or not. That classifier is used to distinguish the small patches of an image, for all locations and scales, as either faces or non-faces (Yang, 2009). It is very complex to build a robust face classifier. Therefore, learning-based approaches, such as AdaBoost (Viola and Jones, 2001), neural-network-based methods (Rowley et al., 1998) or support vector machines (Shavers et al., 2006, Osuna et al., 1997), have been proposed to find a good classifier. A review of the face detection techniques is presented in Zhang and Zhang (2010).

The main challenges of a face detection are related to the illumination and complexity of the scene, the rotation and even the occlusion of the faces and other environmental tricks and traps. Most of the face detection methods use pixels values as features for the classification problem. However, they are very sensitive to illumination conditions and noise. In turn, numerous methods have been proposed

to detect faces in an image using the image features. Among the face detection methods, the ones based on learning algorithms have attracted much attention recently and have demonstrated excellent results. In this thesis, we consider the two most common techniques for a face detection used in VSS:

- **Haar-like Features (HF):** A framework for robust and extremely rapid object detection was presented in Viola and Jones (2001). The goal of this framework is the detection of faces. The authors introduce the use of Haar-like Features to construct a strong classifier by cascading a small number of distinctive features using Adaboost as shown in figure 2.1. The features employed by the detection framework universally involve the sums of image pixels within rectangular areas (integral image). As such, they bear some similarities to the Haar basis functions, which have been used previously in the field of image-based object detection. Its result is more robust and computationally efficient. Although, Haar-like features provide a good accuracy and performance in extracting textures and features, the cascading architecture and integral image representation make them computationally efficient.

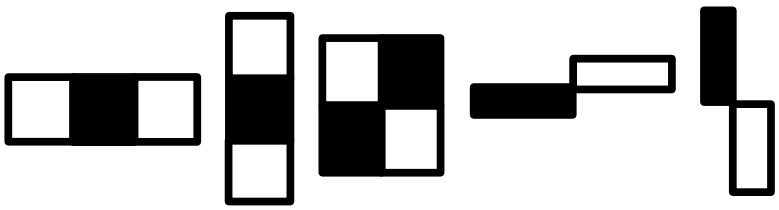


Figure 2.1: Feature types used by Viola and Jones.

- **Local Binary Pattern (LBP):** New rotation invariant and computationally lighter feature sets was proposed in Hadid and Pietik (2004). The basic Local Binary Pattern features have performed very well in various applications, including texture classification and segmentation, image retrieval and surface assessment. The original LBP operator labels the pixels of an image by thresholding the 3×3 neighborhood of each pixel with the center pixel value and considering the result as a binary number. The 256-bin histogram of the labels computed over an image can be used as a texture descriptor, see Figure 2.2. Each bin of histogram (LBP code) can be regarded as different types of edges, corners, flat areas, etc. The LBP operator has been extended to consider different neighbors sizes of 4 or 16. Each face image can be considered as a composition of micro-patterns which could be effectively detected by the LBP operator. Although the LBP feature is simple and can

distinguish faster between faces and non-faces, it suffers from environmental changes. Also, it is difficult to determine the threshold used to differentiate between faces and non-faces.

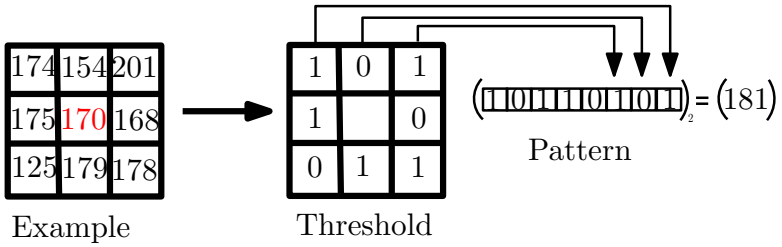


Figure 2.2: LBP pattern calculation.

Evaluation of face detection techniques

To assess the accuracy of face detection, we have implemented the HF-based and LBP-based methods using the OpenCV 2.3 library. They have been compared using CMU-VASC databases for face evaluation [1]. Both algorithms have been tested upon 115 images containing 434 faces as test images, and 48 images containing 85 faces rotated in different angles. Qualitative results are shown in Figure 2.3. In addition, table 2.1 shows the face detection rate and the cost in frames per second (the larger the better). It is apparent that the face detection rate with HF-based is higher than with LBP features. Regarding performance, we have tested the methods with 320×240 pixel images on a 3.2 GHz Intel Pentium DualCore computer. As shown in Table 2.1, face detection based on LBP works faster than based on HF. However, the accuracy of HF-based is better than using LBP-based. As a conclusion, one should use a face detection based on HF, when the accuracy is the most important issue and the process can be executed on a fast hardware.

Methods	Face dataset	Detection rate	fps
Haar-features	Normal	91%	15.3
	Rotated	68%	
Local binary pattern	Normal	83%	29.5
	Rotated	48%	

Table 2.1: Comparison of the two analyzed methods for face detection.

Figure 2.4 shows a good qualitative comparison for the two faces detectors using (320x240) real images captured by a real camera (Logitech QuickCam Orbit/Sphere AF) on a standard PC (3.2 GHZ Intel(R) Pentium (R) DualCore). The

[1] CMU/VASC Database: http://vasc.ri.cmu.edu/idb/html/face/index.html.

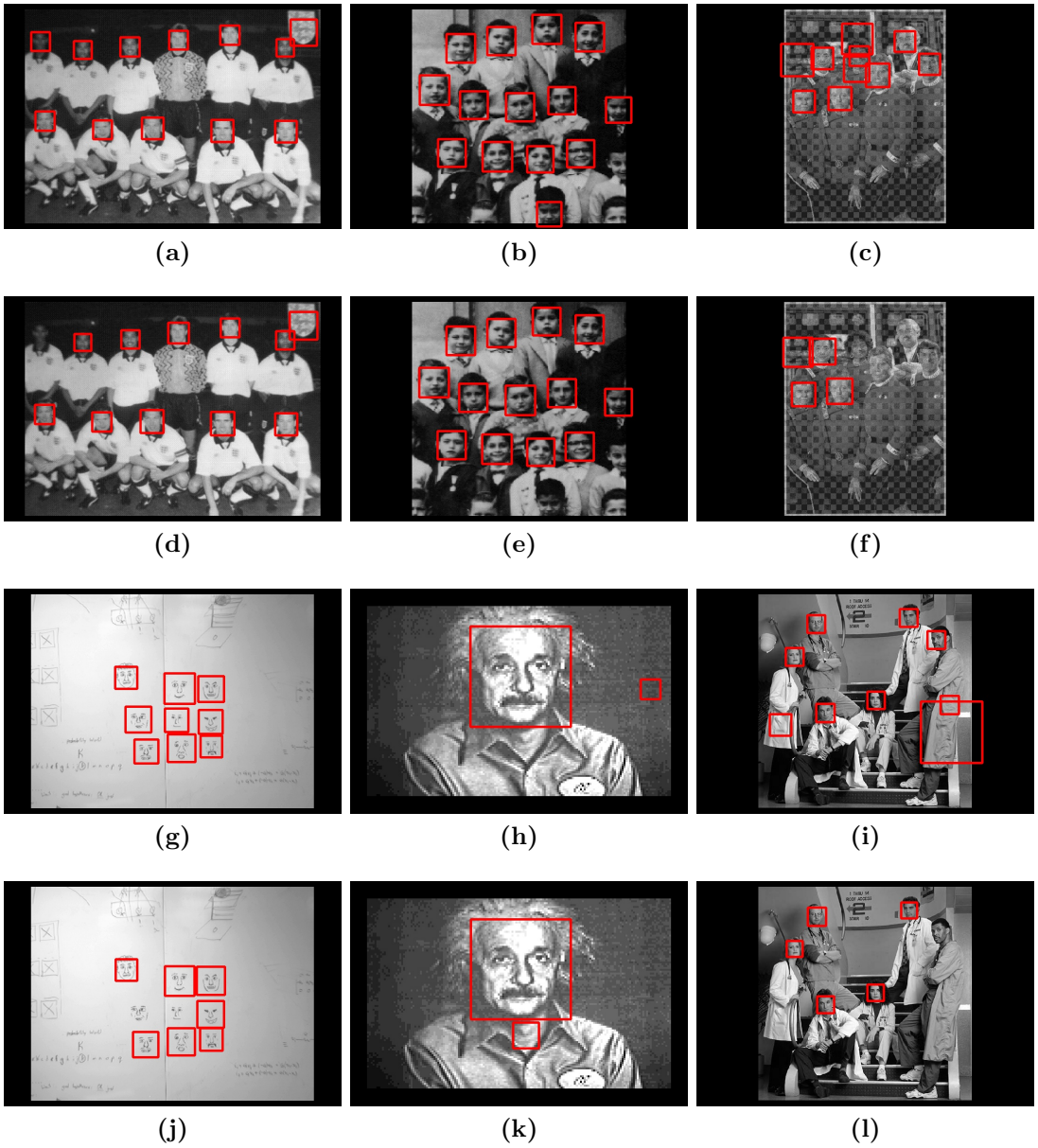


Figure 2.3: Results with CMU/VASC faces database. (1st and 3th rows) results with Viola and Jones (2001). (2nd and 4th rows) results with Hadid and Pietik (2004).

throughput, accuracy and trust score with the two mentioned techniques have been shown in table 3.2. As shown, face detection based LBP detector is faster than face detection based on HF features, as shown in 2.2. However, the accuracy of HF-based algorithm is better than LBP-based algorithm. Thus face detection

based on HF is more trusted technique than one based on LBP. However, it must be stated that if a VSS considers the faces as ROIs, some identity disclosure could be done by merely analyzing clothes or via gait recognition processes.

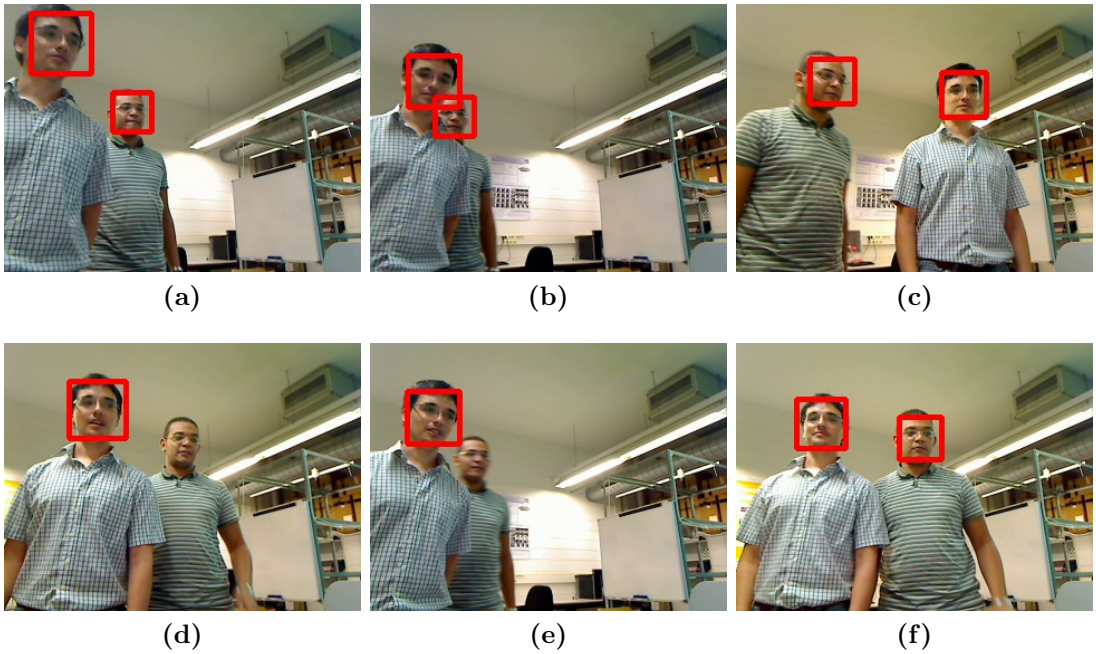


Figure 2.4: Results with 320x240 real-time images with Logitech QuickCam Orbit/Sphere AF. (a-c) Detected faces based on HF features. (e-f) Detected faces based on LBP features.

Methods	fps	Accuracy	Trust score
Face detection with HF	15.3fps	good	High trust
Face detection with LBP	29.5fps	Average	Low trust

Table 2.2: Performance throughput with mentioned techniques Viola and Jones (2001) and Hadid and Pietik (2004) on a standard PC. (3.2 GHZ Intel(R) Pentium (R) DualCore). Execution time calculated for the 320x240 video stream captured using Logitech QuickCam Orbit/Sphere AF.

2.1.2 Motion detection

Motion detection has great importance in the analysis of dynamic scenes, with a variety of applications to motion segmentation and object tracking. Important

applications of motion detection include video surveillance (Wren et al., 1997, Collins et al., 2000), remote sensing (Bruzzone and Prieto, 2002, Huertas and Nevatia, 1998), medical diagnosis and treatment (Bosc et al., 2003, Rey et al., 1999), civil infrastructure (Nagy et al., 2001) and driver assistance systems (Fang et al., 2003). Despite the diversity of applications, motion detection researchers employ many common processing steps and core algorithms. The goal of this section is to present a brief survey of the common algorithms of the optical flow estimation.

The goal of motion detection is to identify the set of pixels that are significantly different between the last image of the sequence and the previous images; these pixels comprise the change mask. The motion mask may result from a combination of underlying factors, including appearance or disappearance of objects, motion of objects relative to the background, or shape changes of objects. In addition, stationary objects can undergo changes in brightness or color. the main challenge of detection is that the detection regions should not contain unimportant or nuisance forms of change, such as those induced by camera motion, sensor noise, illumination variation, non-uniform attenuation, or atmospheric absorption.

Many techniques have been proposed in order to estimate motion from a given sequence of images. Common techniques used for motion detection in a scene are *Background Subtraction* and *Optical Flow Estimation*.

Background subtraction

Background subtraction techniques depend on two main stages: constructing the background model and then detecting the foreground. In addition, according to Cristani et al. (2003), three aspects can describe the background model respectively: the initialized model, the represented model and the updated model. The correct initialization yields the best background model with small errors. Therefore, techniques that analyze video sequences with presence of moving objects in the whole sequence should consider different initialization schemes to avoid the acquisition of an incorrect background of the scene.

The main contribution of background subtraction (BS) algorithms is the detection of foreground objects as the difference between the current frame and a static background of the scene, assuming a fixed camera. Recently, numerous methods have been developed and the most used are the statistical ones (see Ren et al., 2003, Bouwmans, 2008, Baf et al., 2008, Herrero and Bescós, 2009). There are many challenges in developing a good background subtraction algorithm. First, it must be a robust algorithm against illumination changes. Second, it should avoid the detection of non-stationary background objects such as moving leaves, rain, etc., and shadows cast by moving objects. In the literature, there are three common algorithms in a background subtraction: background subtraction based

on Mixture of Gaussians, Kernel Density Estimation and Codebook Construction.

- **Mixture of Gaussians (MoG-BS):** This technique proposed in Stauffer and Grimson (1999) characterizes each pixel by its intensity in the RGB color space. Then, the probability of observing the pixel value in the multi-dimensional case is expressed by means of a Gaussian probability density function that can be expressed as:

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t}.\eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{2.1}$$

where k is the number of distributions, $\omega_{i,t}$ is a weight associated to the i Gaussian at time t with mean $\mu_{i,t}$, standard deviation $\Sigma_{i,t}$, and η is a Gaussian probability density function:

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t-\mu)\Sigma^{-1}(X_t-\mu)}. \tag{2.2}$$

The MoG technique is robust against illumination changes. Unfortunately, this model performs poorly when the background consists of dynamic textures such as trees waving in the wind and rippling water. Furthermore, it gives non-coherence foreground objects that have many gaps, which is could be solved through a morphological dilation operator.

- **Kernel Density Estimator (KDE-BS):** This technique proposed in Elgammal et al. (2000) estimates the probability density function of each pixel by using the last N frames. KDE proposes a Parzen-window estimate of every background pixel and, when dealing with color video frames, products of one dimensional kernels (typically Gaussian ones) as:

$$P(I_s, t) = \frac{1}{N} \sum_{i=t-N}^{t-1} \prod_{\{R,G,B\}} K(\frac{I^j_{s,t} - I^j_{s,i}}{\sigma_j}), \tag{2.3}$$

where, K is a kernel (typically a Gaussian one) and N is the number of previous frames used to estimate $P(.)$. And σ_j can be fixed or preestimated as proposed in Elgammal et al. (2000).

Foreground/Background pixel classification is decided if its likelihood of belonging to the pixel PDF is lower or higher than a predefined threshold. This approach is able to analyze sequences with multimodal backgrounds and it is more reliable on noisy images. However it still suffers from the problem of dynamic textures and outdoor conditions.

- **Codebook Construction (CC-BS):** This is an adaptive background subtraction technique that is able to model a background from a training sequence (Kim et al., 2004). the authors assume that the pixels are mostly distributed along the axis going toward the origin point over time. The CC model assumes that the background pixel intensities lie along the principal axis of the codeword with the low and high bound of pixel intensity since the change is only due to the brightness as shown in Figure 2.5. The proposed algorithm works well on moving backgrounds, with illumination changes, and compressed videos. Comparisons with other background modeling algorithms such as the MoG model and the KDE model show that CB is faster than the others and has good properties for several background modeling problems. However, it still suffers from the aforementioned outdoor environmental factors.

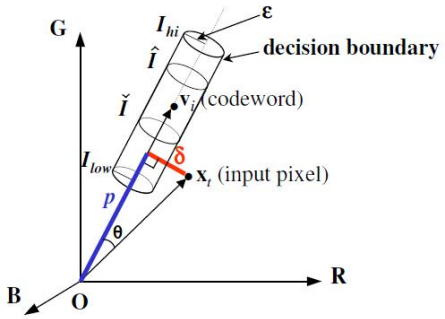


Figure 2.5: Codebook color model proposed in Kim et al. (2004).

Optical flow estimation

Optical flow methods aim at estimating the spatial displacement of every image pixel between two sequential images. In particular, optical flow is an approximation of the local image motion based on local derivatives given consecutive images (see Weickert et al., 2006). It is assumed that intensity variations in the images are only due to the motion of the objects present in the depicted scenes, not to illumination changes. The advantage of optical flow estimation used for ROIs detection is not only to determine the localization (position) of the observed objects in the scene, but also to detect the direction of objects motion, which is very important for tracking.

Among a large amount of families used for estimating flow fields, the *variational* approaches (or differential-based) yield the best performance to estimate the optical flow field and are the most widely used techniques (Baker et al., 2010). They allow the estimation of dense optical flow fields, in many cases even in regions

without distinctive features, where other techniques would generate no more than voids. Thus, the main assumption for the optical flow estimation is the brightness constraint or brightness constancy that assumes the brightness is constant between two consecutive frames that can be defined as:

$$I(x + dx, y + dy, t + dt) - I(x, y, t) = 0$$

$$I_x u + I_y v + I_t = 0 \tag{2.4}$$

The variational techniques are classified into two main categories: (i) local methods (that filter image gradients in a local neighborhood around a pixel and assume that the velocity field of small patch of pixels changes slowly) (see Lucas and Kanade, 1981), and (ii) global methods (that apply a global optimization procedure based on the regularization term for estimating flow field) (see Horn and Schunck, 1981). Local methods give robust flow fields against noise, but they fail to obtain a dense optical flow field. In contrast, global methods present dense optical flow fields but they are more sensitive to noise.

Next, we recall the most outstanding variational techniques in optical flow:

- **Lucas/Kanade (LK-OF):** This is the basic approach of local variational methods (Lucas and Kanade, 1981). It assumes that the optical flow in the local neighborhood of every pixel is uniform and can be estimated by applying least squares as:

  $$\nabla_2 I w = -I_t, \tag{2.5}$$

  where I is an image, $\nabla_2 I$ is the spatial gradients and I_t the temporal gradient.

  In practice, it is usually better to give more weight to the pixels that are closer to the central pixel. Thus, it is often used the weighted version of the least squares equation by applying a Gaussian filter around a small patch. They yield flow fields except in homogeneous image regions, in which gradients are null. Since they filter out the input gradients, these approaches have a good noise tolerance.

- **Horn/Shrunk (HS-OF):** This technique introduced in Horn and Schunck (1981) minimizes functional errors by forcing the smoothness of the resulting flow field over the whole image to solve the aperture problem. The energy function proposed in Horn and Schunck (1981) can de defined as:

  $$E(w) = \int_{\Omega} M(w) + \alpha V(w) dx dy, \tag{2.6}$$

  where Ω is video (spatial-temporal) plane, $M(w)$ is called data term and $V(w)$ is the smoothness (regularization) term.

Therefore, this method yields dense flow fields even in homogeneous image regions. However, it is more sensitive to noise since it does not apply any kind of local filtering to the input gradients.

- **Farnebäck (FB-OF):** This is a more recent method (Farneback, 2000) also known as tensor-based method. It uses polynomial expansion to approximate the neighbors of a pixel. The expansion could be seen as a quadratic equation with matrices and vectors as variables and coefficients. This method yields a dense optical flow that produces a displacement field from two consecutive frames by computing $3D$ orientation tensors from the image sequence. These tensors are combined under the constraints of a parametric motion model to produce velocity estimates. This approach is more robust than Lucas/Kanade and Horn/Shrunk methods.

- **Bruhn/Weickert (BW-OF):** A new combination between the local and global optical flow methods was recently proposed in Bruhn et al. (2006). This work introduced to a unifying multi-grid approach to variational optic flow computation in real-time and analyzed the smoothing effects in local and global differential methods. As a result, they have proposed the application of the 2D Gaussian filtering with structure tensors suggested in Lucas and Kanade (1981) to the global method originally proposed in Horn and Schunck (1981) in order to obtain a dense flow field less sensitive to image noise. They proposed a very accurate and fast algorithm that is a very robust against noise, and gives an accurate dense flow field based on multi-grid techniques to speed up the minimizing of the main optimal procedure with regularization:

$$E(u,v) = \int_{\Omega} \varphi(M(u,v,I)) + \alpha\psi(V(\nabla_2 u, \nabla_2 v))dxdy, \tag{2.7}$$

where $\varphi(.)$ and $\psi(.)$ are convex functions to avoid outliers.

Moreover, they used image pyramids that are used to detect large displacements. Unfortunately, Gaussian filters with structure tensors used are isotropic and do not preserve discontinuities and boundaries of the objects in the scene.

Evaluation and experimental results

We have evaluated the aforementioned techniques with respect to our trustworthy privacy aware VSS requirements. We show in Figure 2.6 a qualitative comparison of the presented methods. We have used the video sequences of the CAVIAR

database [2]. In addition, another qualitative comparison is shown in Figure 2.7 for the tested motion detection algorithms using (320x240) real images captured by a real camera (Logitech QuickCam Orbit/Sphere AF) on a standard PC (3.2 GHZ Intel(R) Pentium (R) DualCore).

Among optical flow estimation techniques, LK-OF and HS-OF are more sensitive to texture, noise and illumination and light changes than the others. Table 2.3 shows the throughput, accuracy and trust of the aforementioned algorithms. The accuracy of algorithms was gauged by using the unsupervised boundary-based evaluation proposed in Chabrier et al. (2006). Hence, the accuracy of techniques can be classified into *poor* for the interval $[0\%, 60\%]$, *average* for the interval $(60\%, 85\%]$ and *good* for the interval $(85\%, 100\%]$. The running time has been calculated on the aforementioned computer. Throughput is considered to be in real time if it allows processing more than 10 fps. In addition, the trust in the tested techniques can be classified into *no-trust* when the accuracy of the technique is poor, *low-trust* when the accuracy of the technique is average and it works in real-time, or when the accuracy of the technique is good and it does not work in real-time, and *high-trust* when the accuracy of the technique is good and it works real-time.

It can be observed that CC-BS is the fastest technique among the proposed schemes. In contrast, BW-OF is the slowest one, although it gives an accurate segmentation for moving regions. The CC-BS technique gives the best accurate segmentation for moving objects. Moreover, FB-OF gives acceptable results for the detection of moving regions with a reasonable throughput. MoG-BS and KDE-BS work in real-time, but they give only an average accuracy. Finally, LK-Of and HS-OF provide the poorest results and they are not trusted ROIs detection schemes. Algorithms CC-BS and FB-OF give a good accuracy and work in real-time, therefore they are highly recommended for implementing a TP-VSS.

Methods	Detection rate	Accuracy	fps	Trust level
MOG-BS	76.8%	Average	16	Low
KDE-BS	77.3%	Average	15.5	Low
CC-BS	93.1%	Good	400	**High**
LK-OF	33.7%	Poor	80	No trust
HS-OF	35.2%	Poor	12	No trust
FB-OF	87.4%	Good	14.5	**High**
BW-OF	93.5%	Good	1.8	Low

Table 2.3: Evaluation of the trust offered by ROI detection methods according to their accuracy and their performance in real time.

[2]CAVIAR: Context Aware Vision using Image-based Active Recognition http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

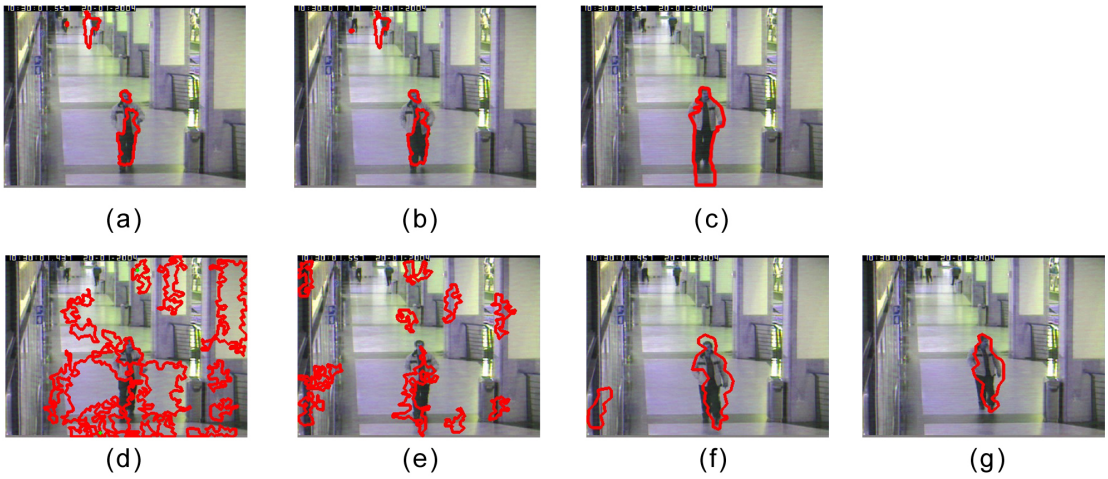


Figure 2.6: One of the experiments for testing motion detection using (a) Mixture of Gaussians, (b) Kernel Density Estimator, (c) Codebook Construction, (d) Lucas/Kanade, (e) Horn/Shrunk, (f) Farnebäck and (g) Bruhn/Weickert.

2.2 Techniques for protection

After automatically detecting ROIs, privacy is protected in those regions by the *Protection* sub-module (cf. Figure 1.1). It may perform distortion operations such as blurring to protect the data (which is stored in the Information System) against hacking and release over untrusted networks. In this section, we present a novel categorization of the techniques found in the literature, paying special attention to the *utility* of the data retrieved by the trusted manager. In general, the data gathered by a VSS may be used for monitoring purposes (*e.g.* identifying users, tracking people, etc.). We divide the proposals in two groups as shown in Figure 2.8, depending on the domain in which ROIs are protected: first, *pixels domain* techniques, which modify the ROI in every frame (raw images), before compression of the video; second, *compression domain* techniques, which modify the data in the container of the compressed video.

2.2.1 Pixels Domain

There are several proposals in the literature dealing with ROI protection in the pixel domain. The Video Processing Module proceeds as follows:

1. Extract the frame from the compressed stream
2. Decompress the frame

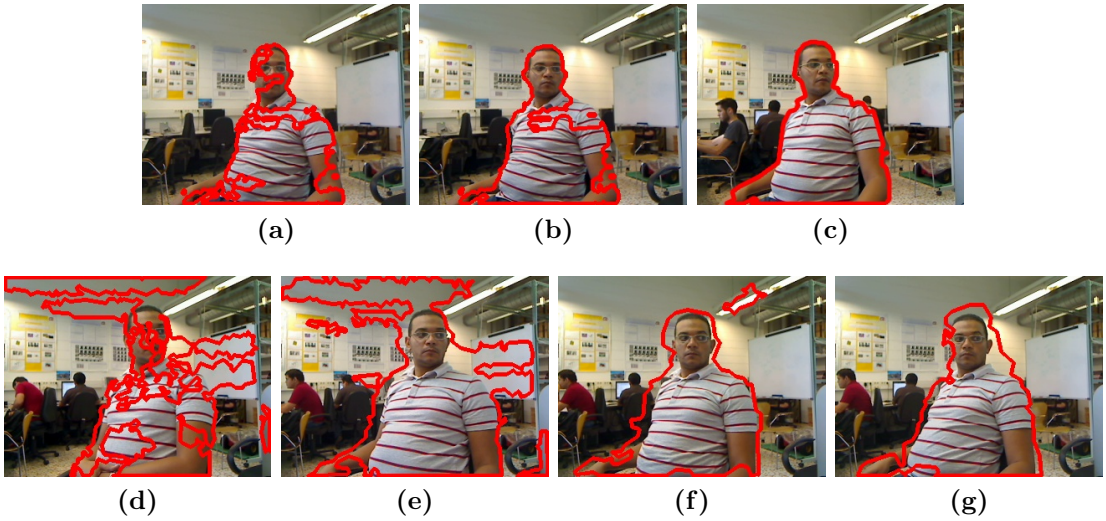


Figure 2.7: (an experiment for testing motion detection using (a) Mixture of Gaussians, (b) Kernel Density Estimator, (c) Codebook Construction, (d) Lucas/Kanade, (e) Horn/Shrunk, (f) Farnebäck and (g) Bruhn/Weickert.

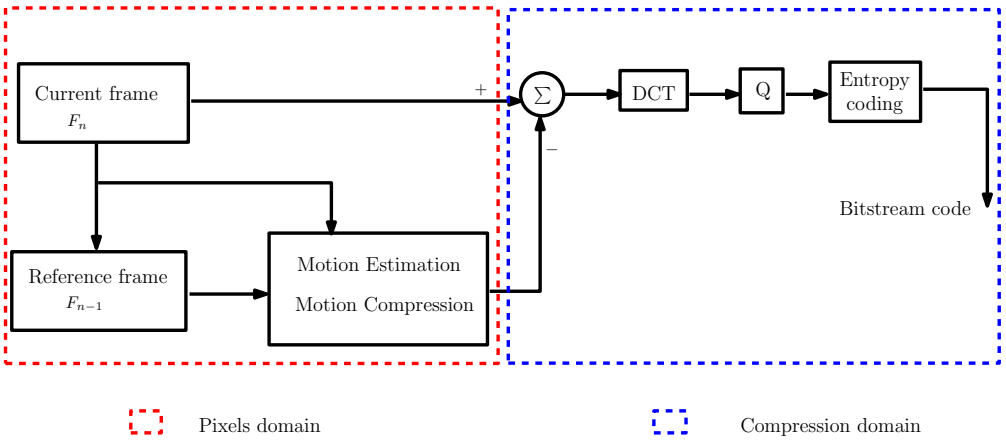


Figure 2.8: A pixels domain and compression domain system diagram.

3. Detect the ROIs in the frame

4. For each ROI in the frame, obscure it

5. Compress the frame

6. Insert the frame in the compressed stream

Steps 1 and 2 are only for stored video. If we use a camera, we may directly operate over the uncompressed raw images. Regarding ROI protection in pixel domain, we can classify the existing literature in three trends:

- **Simple pixel transformation.** It consists of replacing the value of a pixel. The most common approaches are blurring (applying a Gaussian filter to remove the details of the ROI) and pixelization (replacing a block of pixels by their average values) Berger (2000), Newton et al. (2005), Wickramasuriya et al. (2004), H. Wactlar and Ng (2002). The implementation of such techniques is very simple but their application results in a non-invertible protected video (*i.e.* it is a one-way operation).

- **Cryptography-based techniques.** Some other methods in the pixel domain make use of encryption (see Spinder et al., 2006, Carrillo et al., 2009, Upmanyu et al., 2009). For instance, in Carrillo et al. (2009) the pixels in a ROI are permuted pseudo-randomly. The generation of this permutation depends on a key that is the seed of a pseudo-random number generator (PRNG). This means that the same key allows the PRNG to output the same series of numbers and hence it can be used for both encryption (protecting the ROIs) and decryption (obtaining the unprotected ROIs). If ROIs are protected using this technique, the utility of the protection technique is low: the permutation of pixels results in a set of high-frequency image blocks; then, these blocks will pass through the compression procedure, which will discard high frequency components so as to decrease the video size; as a result, protected ROIs will suffer a heavy information loss after compression and it will be difficult to obtain the original image from a compressed and protected frame.

- **Abstraction-based techniques.** Those consist in replacing a ROI (*e.g.* a person) by a shape (*e.g.* a silhouette) in the pixel domain. An example of those techniques can be found in Tansuriyavong and Hanaki (2001), Senior et al. (2005), Cavallaro (2004). In addition, Cavallaro (2007) presents a surveillance system that use the camera to directly separates the gathered data into personal data and behavioral data as shown in Figure 2.9. In this system, users can only access the behavioral data, and only authorities (through law enforcement) have access to the personal data.

All these approaches are computationally feasible. However, in all cases, the information system must store a copy of the non-protected (*i.e.* original) video so as to provide a trusted manager with an unprotected version of the video. As stated in the case of ROIs detection, a security leak in the information system would allow the access to this non-protected copy and compromise the privacy of

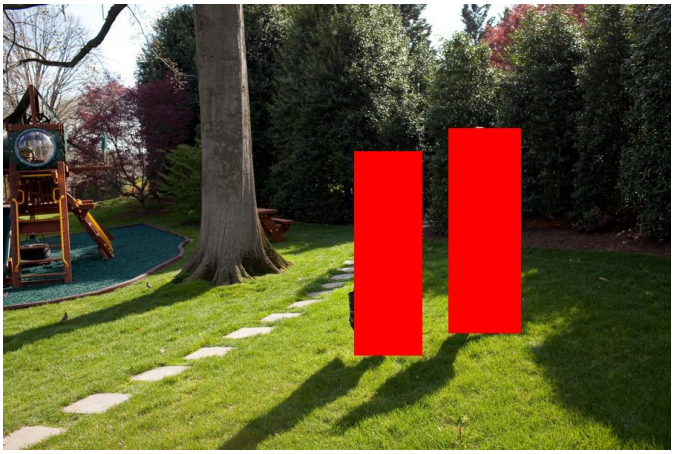


Figure 2.9: A frame protected by an abstraction technique.

the individuals. As a conclusion, the use of ROIs protection in the pixel domain is clearly discouraged.

Protection Techniques done through pixel domain are only worth in uncompressed video, since the protection process would become irreversible. However, if any compression is done on the protected video, we would need a copy of the unprotected video in order to retrieve the original video and this does not yield a trustworthy surveillance system.

2.2.2 Compression domain

The solutions based on the compression domain protect the gathered data during or after the image compression. For the sake of completeness, we briefly introduce the concept of compressed video. A compressed video is a set of compressed frames, grouped in GOPs (*Group of Frames*). Each GOP starts with an I-frame (*intra-coded*) and contains several P-frames (*predicted*) and B-frames (*bi-predictive*). I-frames are stored and compressed entirely: the frame is divided into blocks; a frequency transform (*e.g.* Discrete Cosine Transform) is applied to each block; a quantization is applied to each block (each frequency component is divided by a number, aiming at reducing the number of discrete symbols but resulting in a lossy compression and, also, a set of zero coefficients); finally, entropy encoding (for the non-zero coefficients) and run-length encoding (for the zero coefficients) are applied for a lossless compression of the block. The information needed to reconstruct the frame is stored in a specific and standardized data structure. In addition, P and B-frames are not stored entirely: in a nutshell, they just consist of the changing blocks between frames in the GOP; each P and B-frame is stored as a succession of *slices* (a collection of consecutive blocks), where blocks are not

stored as pixel values, but describing how they change through the frames in the GOP (using *motion compensation* techniques).

As an example, if ROIs are encrypted in the compression domain (*e.g.* some values of the compressed video stream data structure are encrypted), unauthorized users (*i.e.* without a proper decryption key) would obtain noise in the ROI pixel area of the decompressed frame. On the contrary, authorized users (*i.e.* with the corresponding decryption key) would be able to decrypt the structure and hence reconstruct the original ROI. These solutions are dependent on image compression techniques used (since each codec uses a specific data structure for the compressed video) and may require modification of the video encoders. Note that the complete encryption for the video streams does not serve the purpose of surveillance due to the fact that (authorized) viewers could not understand the context (*i.e.* the scene, background, etc.) without decrypting the video.

The literature can be classified in three trends: *cryptographic approaches*, *scrambling-based* and *information hiding*. These transformations are totally reversible. As a result, protecting ROIs in the compression domain does not require storing the original copy, which fulfills our trust goals. Still, note that ROIs may suffer from information loss if some kind of transcoding or recompression is done over the protected stream.

With regard to the input video stream, the protection process should be performed over the compressed video, instead of over a sequence of still frames. And, whenever possible, the protection process should be implemented along with the compression module of the video system. Last but not least, the cryptographic operations involved in the protection should rely on the use of smart cards.

Cryptographic approaches

Several proposals fall into this category. In Boult (2005) the authors use the DES cryptosystem to encrypt the data. DES is a block encryption algorithm and the authors must cope with the block size constraint on the size of the ROIs. The encryption key is protected by public-key cryptography. However, the encryption decreases the efficiency of the entropic compression of video. The authors claim that the loss of efficiency of entropy coding due to encryption could be avoided by applying encryption during or after the entropy coding step. However, the authors do not present any relevant test on this important issue.

This shortcoming is tackled in Shahid et al. (2011), where an algorithm for the protection of H.264/AVC video streams focused on the entropy coding is presented. The encryption is performed using the AES (Advanced Encryption Standard) cryptosystem. The authors claim that by encrypting in the entropy coding step, the compression efficiency is not altered and, consequently, the resulting bitrate is not modified. Unfortunately, the system is valid for I and P-frame streams

only and all the frame is encrypted, without taking into account ROIs detection and protection.

In Yabuta et al. (2005), an architecture to encrypt moving objects over a JPEG stream is presented. AES is used for encryption, and a given password for the image viewer has to be used to show the original JPEG. The authors do not test the system with video streams but with a 320×240 pixel JPEG stream. Moreover, with regard to the processor architecture that runs the protection algorithms, they presented two approaches: a sequential one with a performance of 9.2 fps and a distributed one with a performance of 16.2 fps. Unfortunately, this architecture does not work on video streams and encrypts all moving objects instead of detecting ROIs.

In Martin and Plataniotis (2008), a secure visual object coder that focuses on the shape and texture of visual objects is proposed by using a Shape-Adaptive Discrete Wavelet Transform variant Said and Pearlman (1996) and offering embedded bitrate output. This proposed method uses a selective encryption algorithm, utilizing a stream cipher to encrypt a portion of the output bit stream related to ROIs. The proposal only works for still JPEG frames.

Finally, Cheung (2008) proposes an architecture to exchange key information and retrieve data. ROIs are protected by encrypting with AES the coefficients of the luminance channel. The video is stored as a set of still frames.

Scrambling-based

There is a plethora of proposals based on scrambling the data to produce a privacy-aware video. For the sake of briefness, we only address the proposals working with video streams instead of merely still frames.

Two interesting proposals are Sohn et al. (2009), Dufaux and Ebrahimi (2008), which are quite similar. They present a privacy-aware surveillance system for H.264/AVC video. The VSS described in Sohn et al. (2009) considers human faces as ROIs, whereas Dufaux and Ebrahimi (2008) considers moving objects. The technique scrambles those detected regions in the compression domain using a pseudo-random sign inversion applied to the coefficients of the luminance component in order to scramble ROIs. Authors use different combinations of security keys in order to produce a protected video that is robust against brute force attacks.

Information hiding

A final trend in the literature is information hiding. In Martínez-Ponte et al. (2005), the authors use the JPEG 2000 standard to protect frames, instead of working with video flows. A JPEG 2000 frame consists of a set of quality layers

(each one providing more or less details depending on a quality value), see Figure 2.10. Hence, the method provides authorized users with access to all layers of the picture. On the contrary, unauthorized users would only be able to decode the lowest quality layers. In this proposal, the authors do not deal with the access control to the high quality layers of the frame by means of security techniques. In addition, Fukuoka et al. (2012) proposed a method for producing different level of privacy protected videos from a delivered video at the client side according to the client authority level by implying a way of protection, such as, box, mosaic or transparency.

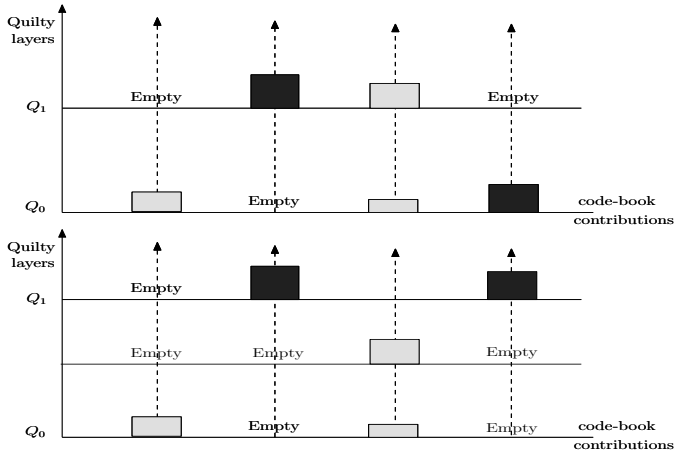


Figure 2.10: Protection based information hiding technique (top) JPEG 2000 quality layers. (Down) JPEG 2000 quality layers with isolated ROIs.

2.3 Discussion on the existing proposals

In this section, we review the cited detection and protection tools, according to their suitability for our concept of VSS. First, we summarize in Table 2.4 the most significant advantages and disadvantages of the methods involved in a VSS. Finally, we elaborate on the lacks we have found upon reviewing the literature.

With respect to the ROI protection, we have addressed face detection and general motion detection. Regarding the tools for face detection, efficiency and accuracy is a constraint. However, we discourage the use of the face as ROI because the identity of individuals could be disclosed via other techniques. The tools for motion detection (considering as ROI any moving object) may fail in scenes with moving backgrounds or dynamic textures such as rain or leaves. To cope with this problem, the use of optical flow is highly recommended.

With respect to the ROI protection, we have observed that systems that protect ROIs in the pixels domain do not cope with our definition of trustworthy privacy:

since protection is not reversible, these systems must keep an unprotected version of the video that could be exposed in case of attacks leading to privacy leaks. Regarding the protection in the compression domain, there is no proposal that entirely takes into account all the aspects needed to asses the trust on the system. However, we encourage using tools that only protect the ROIs in the frames, and perform the protection procedures without significantly increasing the size of the protected video.

Stage		Method		Advantages	Disadvantages	Recommended for a trustworthy system?
ROI detection (Trust is based on accuracy and real-time performance)	Face detection	Haar-features		Good face detection rate	Medium performance	High trust in implementations on fast hardware
	Motion detection	Local Binary Pattern		Good performance	Average face detection rate	Average trust in power constrained systems
		Background Subtraction	Mixture of Gaussians	Simplicity	Average accuracy. Low performance. Only for fixed cameras	Not recommended
			Kernel Density Estimator	Simplicity	Average accuracy. Low performance. Only for fixed cameras	Not recommended
			Codebook Construction	Good accuracy, good performance	Only for fixed cameras	High trust with fixed cameras
		Optical Flow	Lucas/Kanade	Good performance	Low accuracy	Not recommended
			Horn/Shrunk	Good performance	Low accuracy	Not recommended
			Farnebäck	Good accuracy. Robust against scene conditions	Average performance	High trust in implementations on fast hardware
			Bruhn/Weickert	Good accuracy	Very low performance	Not recommended
ROI protection (Trust is based on reversibility and no need of storing the original video)	Pixel domain	Blurring/Pixelization		Simplicity	Transformation not reversible. The utility of the unprotected video decreases. A copy of the original video is needed	Not recommended
		Encryption		Robustness against brute force attacks	Permutation results in noise, that suffers from information loss during the compression process. The utility of the unprotected video decreases	Not recommended
		Abstraction			A copy of the original video is needed	Not recommended
	Compression domain	Encryption over the frequency components		Robustness against brute force attacks	The efficiency of the entropy encoder decreases. Difficulty to accomodate block encryption	Not recommended
		Encryption during the entropy encoding		Robustness against brute force attacks	The bitrate is not modified	Recommended
		Scrambling		Simplicity in random permutation of the sign of the frequency components. The bitrate is not modified	The robustness of the security depends on the generation of the permutation	Recommended
		Information hiding		High quality layers are only available to authorized users	Only valid for JPEG2000 standard	Not recommended

Table 2.4: Comparison of the reviewed tools according to its suitability for a trustworthy privacy-aware VSS.

Chapter 3

Improving the Robustness of Variational Optical Flow Through Tensor Voting

Video surveillance systems typically consist of several functional modules working in concert. The main module of video surveillance analysis performs a motion detection. One way of detecting motion is using optical flow that expresses about the distribution of apparent velocities of movement of brightness patterns in an image. Differential optical flow methods allow the estimation of optical flow fields based on the first-order and even higher-order spatio-temporal derivatives (gradients) of sequences of input images. If the input images are noisy, for instance because of the limited quality of the capturing devices or due to poor illumination conditions, the use of partial derivatives will amplify that noise and thus end up affecting the accuracy of the computed flow fields. The typical approach in order to reduce that noise consists of smoothing the required gradient images with Gaussian filters, for instance by applying structure tensors. However, that filtering is isotropic and tends to blur the discontinuities that may be present in the original images, thus likely leading to an undesired loss of accuracy in the resulting flow fields.

This chapter proposes the use of tensor voting that is a powerful tool in computer vision field as an alternative to Gaussian filtering, and shows that the discontinuity preserving capabilities of the former yield more robust and accurate results. In particular, a state-of-the art variational optical flow method has been adapted in order to utilize a tensor voting filtering approach. The proposed technique has been tested upon different datasets of both synthetic and real image sequences, and compared to both well known and state-of-the-art differential optical flow methods.

The rest of the chapter is organized as follows. Section 3.1 introduces to the state-of-the-art optical flow methods. Related work is discussed in section 3.2. Section 2.3 summarizes the proposed approach in this chapter. Section 3.4 gives an

overview of tensor voting and discusses its relationship with the structure tensor applied in Zimmer et al. (2009) and Bruhn et al. (2005). Sections 3.5 to 3.7 describe the proposed adaptation of tensor voting to the optical flow problem. In particular, Section 3.5 describes the pre-segmentation of the input images based on their spatio-temporal gradients. Section 3.6 describes how image gradients are filtered with tensor voting. The adapted variational optical flow model is then detailed in Section 3.7. Finally, experimental results with both synthetic and real image sequences are shown and discussed in Section 3.8, including a comparison with both well-known and state-of-the-art differential optical flow methods.

3.1 Introduction

The video surveillance systems (VSS) interest in using of imaging sensors to monitor the activity of objects in a video stream Connell et al. (2004). The primary aims of these systems are to provide an automatic interpretation of scenes and to understand the actions of the observed objects through the information taken by special cameras. The main task of video surveillance analysis is detecting moving objects.

Motion detection has a great importance in the analysis of dynamic scenes of VSS. Many techniques have been proposed in order to estimate motion from a given sequence of images. One of those approaches is optical flow, which aims at estimating the spatial displacement of every image pixel between two adjacent images at time t and $t + dt$ respectively. In other words, optical flow is an approximation of the local image motion based on local derivatives given consecutive images. It is assumed that intensity variations in the images are only due to the motion of the objects present in the depicted scenes, not to illumination changes.

Five families of optical flow methods have been proposed in the literature Barron et al. (1992), Baker et al. (2010): correlation-based, energy-based, discrete-optimization, differential and phase-based methods. Correlation-based approaches are suitable for matchable features, such as corners, whereas they are inaccurate for other cases. In turn, energy-based methods, which apply continuous optimization in the frequency domain, yield good estimations of the local orientation of 2D patterns together with corresponding confidence measures, but they solve the optical flow problem locally, not guaranteeing an optimal global solution. On the other hand, discrete optimization schemes, such as graph-cuts, belief propagation and dynamic programming have gained more popularity than the continuous counterparts due to their better ability to minimize non-convex energy functions. Nevertheless, those discrete optimization methods suffer from the problem of label discretization, that is, the difficulty to properly discretize 2D flow fields.

Differential techniques compute image velocities from spatio-temporal deriva-

tives of image intensities. The image domain is therefore assumed to be continuous (or differentiable) in space and time. In contrast, phase-based methods solve the optical flow problem depending on the change of phase of the signal instead of on the change of amplitude of the signal and its derivatives. However, phase correlation may yield ambiguous results with several peaks in the resulting output.

Among the aforementioned techniques, differential and phase based approaches yield the best performance for estimating the optical flow field as mentioned in Barron et al. (1992), Baker et al. (2010). However, differential methods are the most widely used techniques since they allow the estimation of dense optical flow fields, in many cases even in regions without distinctive features, where other techniques would generate voids (see Barron et al., 1992, Baker et al., 2010).

Differential methods estimate optical flow based on the first-order and even higher-order partial derivatives (gradients) of the input images. These approaches can be further classified into local and global methods. Local methods, such as Lucas and Kanade (1981), Bigun et al. (1991), estimate the optical flow at every pixel based on the filtered image gradients in a local neighborhood around that pixel. Alternatively, global methods, also known as variational optical flow methods, such as Horn and Schunck (1981) and its numerous discontinuity-preserving variations (Nagel, 1983, Black and Anandan, 1991, Schnorr, 1994, Weickert and Schnórr, 2001, Bruhn and Weickert, 2005), apply a global optimization procedure based on regularization that determines the optical flow at every pixel from the image gradients of the whole image.

Local methods assume that the optical flow in the local neighborhood of every pixel is uniform and estimate it by applying least squares. They yield flow fields except in homogeneous image regions, in which gradients are null. Since they filter out the input gradients, these approaches have a good noise tolerance. On the other hand, global methods minimize an error functional by forcing the smoothness of the resulting flow field over the whole image. Therefore, these methods yield dense flow fields even in homogeneous image regions. However, they are more sensitive to noise since they do not apply any kind of local filtering to the input gradients.

Trying to overcome the aforementioned drawbacks, Zimmer et al. (2009), Bruhn et al. (2005) proposed a combined approach that merges both local and global methods through Gaussian filtering based on structure tensors. However, Gaussian filtering is isotropic, leading to an undesired blurring of the discontinuities present in the scenes, thus likely yielding an undesired loss of accuracy in the resulting flow fields. The main contribution of this chapter is replacing that Gaussian filtering based on structure tensors by a discontinuity-preserving filtering stage based on tensor voting. This leads to a robust algorithm for estimating an accurate dense optical flow field given a pair of color images, merging the benefits of both local and global differential methods.

Different anisotropic filtering methods have been proposed in the literature to preserve image discontinuities, such as the bilateral filter (Tomasi and Manduchi, 1998) and non-local means (NLM) (Kervrann and Boulanger, 2008). The bilateral filter extends the concept of Gaussian filtering by adding a Gaussian weighting function that depends on the difference between pixel intensities. However, it is unable to filter very noisy images as mentioned in Moreno et al. (2011a). In turn, NLM is an extension of the bilateral filter that uses the spatial distance between pixel neighborhoods instead of pixel intensities. However, an experimental analysis conducted in Moreno et al. (2011a) showed that NLM tends to generate colored spots and undesirable quantization effects, not being satisfactory enough with real noise. Alternatively, Moreno et al. (2011a) proposed the tensor voting framework as a more robust methodology for anisotropic filtering of color images.

Related work

Both the sensitivity and the effect of noise in local and global optical flow differential methods have widely been analyzed in the literature, Bainbridge-Smith and Lane (1997), Fermüller et al. (2001) and Galvin et al. (1998). The conclusion is that global methods are more sensitive to noise than local methods. In order to take advantage of the complementary benefits of both local and global methods, some researchers have also proposed their combination. For instance, Schnorr (1993) proposed such a combined technique by applying Gaussian filters shifted in the frequency space or local methods integrating second-order derivatives instead of the solutions proposed in Lucas and Kanade (1981), Bigun et al. (1991). More recently, Bruhn et al. (2005) analyzed the smoothing effects in local and global differential methods for optic flow computation. In particular, Bruhn et al. (2005) applied the 2D Gaussian filtering suggested in Lucas and Kanade (1981), Bigun et al. (1991) to the global method originally proposed in Horn and Schunck (1981) in order to obtain dense flow fields less sensitive to image noise.

Unfortunately, Gaussian filters are isotropic and do not preserve discontinuities. Therefore, their application may lead to the propagation of incorrect information to pixels located between different image regions, such as object boundaries, or between objects that move along different directions. As a result, the computation of the optical flow field may be seriously affected at those regions.

Alternatively, Little et al. (1988) proposed a framework for calculating the optical flow field through a local voting scheme based on the similarity of planar patches. However, this approach can not prevent motion boundary blurring due to over-smoothing and it is restricted to short-range motion.

Following a voting scheme, Gaucher and Medioni (1999) is the first work that proposed tensor voting for solving the motion flow estimation problem. Tensor voting is a perpetual organization technique originally proposed in Tong et al. (2001),

Medioni et al. (2000). Two separate voting processes are defined in Gaucher and Medioni (1999): one to determine boundary points as the pixels with maximum motion uncertainty, and another to locally refine velocities near boundaries by allowing voting only between pixels placed at the same side of a boundary. However, the voting process between pixels is essentially a $2D$ process that does not reduce the influence of different velocities of neighbors upon each other.

In addition, a visual motion analysis and interpretation framework was presented in Nicolescu and Medioni (2003). That work proposed an approach for motion segmentation from two images using a 4D framework in order to handle real data, and integrated it with a 2D voting-based method for accurate inference of motion boundaries. This approach consists of a correlation-based matching process that recovers feature correspondences as a sparse velocity field, followed by a motion capture process that infers motion boundaries and regions, and an interpretation process that determines the $3D$ structure and motion of the defined objects using $4D$ tensor voting (image position and velocity) and taking into account the constancy velocity assumption over an image area at small time intervals.

In this chapter, a new approach for estimating motion detection based on optical flow for video surveillance sequences is presented. Our method proposes a combined local and global optical flow method based on an adaptation of the approaches described in Bruhn et al. (2005) and more recently in Zimmer et al. (2009), by replacing the isotropic Gaussian filtering based on the structure tensor utilized in Zimmer et al. (2009) and Bruhn et al. (2005) by a discontinuity-preserving filtering stage based on tensor voting.

3.2 Approach overview

The proposed approach estimates the optical flow field given a pair of consecutive images in several stages shown in Figure 3.1.

Let $I(p)$ be an image sequence with $p = (x, y, t)$, where x and y denote the position in the image domain and t denotes time. The optical flow field, $w = (u, v, 1)$, represents the displacement vector field between two frames at times t and $t + dt$. The first stage determines the first spatio-temporal image derivatives, (I_x, I_y, I_t), and the second spatio-temporal image derivatives, (I_{xx}, I_{xy}, I_{xt}) and (I_{yx}, I_{yy}, I_{yt}), for a pair of consecutive images in the image sequence $I(p)$. In the second stage, the image pixels are classified into homogeneous-moving regions (HM), textured-moving regions (TM) and stationary (not moving) regions (NM) based on the spatio-temporal image derivatives. This stage will be explained in detail in Section 5. In the third stage, a classical tensor voting approach described in Tong et al. (2001), Medioni et al. (2000) is used to independently filter the spatio-temporal image derivatives of the first two classes (HM and TM). The result of this stage

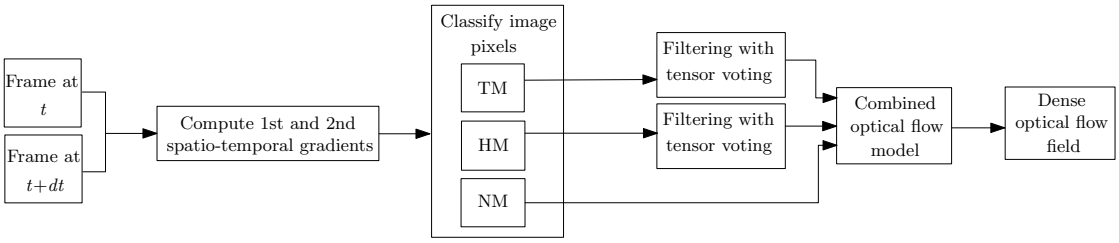


Figure 3.1: Overview of the proposed approach for estimating a dense optical flow field from a pair of consecutive images.

is a smoothing of the spatio-temporal image gradients that preserves discontinuities. The fourth stage solves the optical flow problem using the combined global differential method described in Zimmer et al. (2009), Bruhn et al. (2005) based on the results of the tensor voting stage. Finally, the results are integrated in a dense optical flow field.

3.3 Tensor voting as an alternative to the structure tensor

Tensor voting is a robust methodology for propagating and fusing both 2D and 3D information in the presence of noise, (see Tong et al., 2001, Medioni et al., 2000). In 3D, the information associated with every data point is encoded as a tensor and propagated to its neighboring points through a convolution-like process. Afterwards, the analysis of the resulting tensors leads to the location of surfaces, edges and junctions. This approach takes advantage of the Gestalt principles of proximity, similarity and good continuation in order to estimate perceptual saliency.

In particular, the result of applying tensor voting at point (pixel) p is a tensor, $TV(p)$, defined as:

$$TV(p) = \sum_{q \in \Theta(p)} SV(v, S_q) + PV(v, P_q) + BV(v, B_q), \tag{3.1}$$

where q represents every point belonging to the neighborhood Θ of p. SV, PV and BV are the stick, plate and ball tensor votes cast to p by every component of q, and $v = p - q$. S_q, P_q and B_q are the stick, plate and ball components of the

tensors at q, respectively:

$$\begin{aligned} S_q &= (\lambda_1 - \lambda_2)e_1e_1^T \\ P_q &= (\lambda_2 - \lambda_3)(e_1e_1^T + e_2e_2^T) \\ B_q &= \lambda_3(e_1e_1^T + e_2e_2^T + e_3e_3^T), \end{aligned} \tag{3.2}$$

where λ_i and e_i are the i-th eigenvalue and its corresponding eigenvector of the tensor at q, respectively. Saliency measurements can be estimated from an analysis of the eigenvalues of the resulting tensor in (3.1). Thus, $s_1 = \lambda_1 - \lambda_2$, $s_2 = \lambda_2 - \lambda_3$ and $s_3 = \lambda_3$ can be used as measurements of *surfaceness*, *edginess* and *junctionness*, respectively (Figure 3.2). Small eigenvalues imply noisy points. Moreover, eigenvector $e1$ represents the estimated normal for points lying on a surface, while $e3$ represents the most likely tangent direction of a curve for points belonging to that curve.

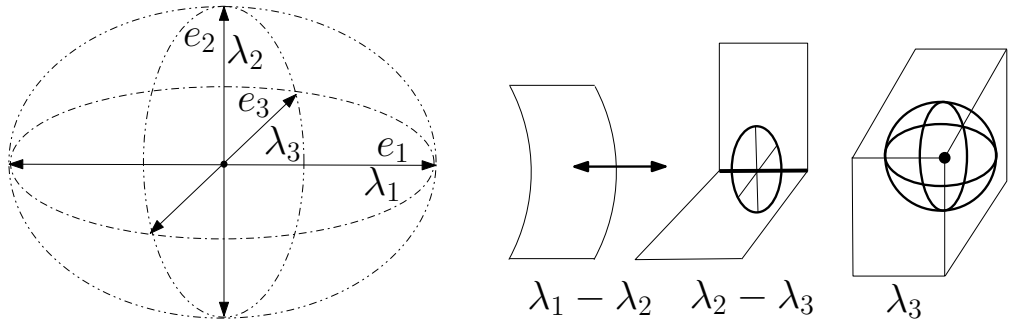


Figure 3.2: Geometrical interpretation of tensor voting.

A stick tensor encodes the orientation of the surface normal at a specific 3D point. Stick tensor voting aims at propagating surfaceness in a neighborhood by using the perceptual principles of proximity, similarity and good continuation. Given a known orientation of the normal at a point q, which is encoded by Sq, the orientation of the normal at a neighboring point p can be inferred by tracking the change of the normal on a joining arc of a circle, Figure 3.3(left). Thus, the stick tensor voting can be written as:

$$SV(v, S_q) = f_s[R_{2\theta}S_qR_{2\theta}^T], \tag{3.3}$$

where f_s is a decaying function, θ is the angle shown in Figure 3.3(right) and $R_{2\theta}$ a rotation with respect to the axis $v \times (S_qv)$. Function f_s was defined in Tong et al. (2001) as:

$$f_s(v, s_q) = \begin{cases} e^{-\frac{l^2+bk^2}{\sigma^2}} & if\ \theta \leq \pi/4 \\ 0 & otherwise, \end{cases} \tag{3.4}$$

where σ is the standard deviation of a Gaussian function that modulates the influence of q over p based on their Euclidean distance, l is the length of the curve, k is the curvature of the path and b is a function of σ (see Medioni et al., 2000). In turn, a plate tensor encodes an edge, whereas a ball tensor encodes either a junction or noise. Plate and ball fields are obtained by integrating stick spanning disks and spheres respectively (see Tong et al., 2001, Medioni et al., 2000).

On the other hand, the structure tensor, which is applied in Zimmer et al. (2009), Bruhn et al. (2005), makes the assumption that gradients change in a neighborhood slowly. Thus, the geometrical structure in that neighborhood can be estimated through a weighted sum of the gradients belonging to the neighborhood. In particular, given two consecutive image frames, $I(x, y, t)$ and $I(x, y, t + dt)$, the structure tensor, J, is defined as the convolution of a Gaussian G_ρ with the tensor of the image gradient (Liou and Jain, 1989):

$$J_\rho = G_\rho * \nabla_3 I (\nabla_3 I)^T, \tag{3.5}$$

where ∇_3 is the spatio-temporal gradient operator, $\nabla_3 = (\partial_x, \partial_y, \partial_t)^T$.

Although both tensor voting and the structure tensor can be utilized for estimating the geometrical structure of images from their gradients, they have significant differences due to their particular assumptions. On the one hand, the structure tensor can be interpreted as a voting process in which the voter q propagates the orientation of its gradient to the *votee* p whenever the former is in a neighborhood of the latter, as shown in Figure 3.3(right).

In turn, tensor voting makes the additional assumption that both the voter and the *votee* must lie along a smooth curve (*i.e.*, an image contour). Under this assumption, the voter propagates its gradient to the *votee* if the angle θ between them is lower than or equal to 45^o, Figure 3.3(left). If this condition is satisfied, the gradient propagated to the *votee* is not the one at the voter, as in the structure tensor, but a rotated version of it perpendicular to the aforementioned smooth curve that hypothetically joins both points.

Both the 3D structure tensors and 3D tensor voting have experimentally been tested in order to estimate the geometrical structures (surfaces, edges, junctions) present in consecutive images by considering their 3D spatio-temporal gradients (see Section 4). Figure 3.4 shows the map of $(\lambda_2 - \lambda_3)$ that can be used to extract edginess. It can be observed that the structure tensor causes blurring, whereas tensor voting tends to preserve discontinuities.

Therefore, tensor voting propagates image gradients more consistently from a geometrical point of view than the structure tensor. As a consequence, the result of filtering image gradients with tensor voting is likely to yield more accurate results than by applying the structure tensor.

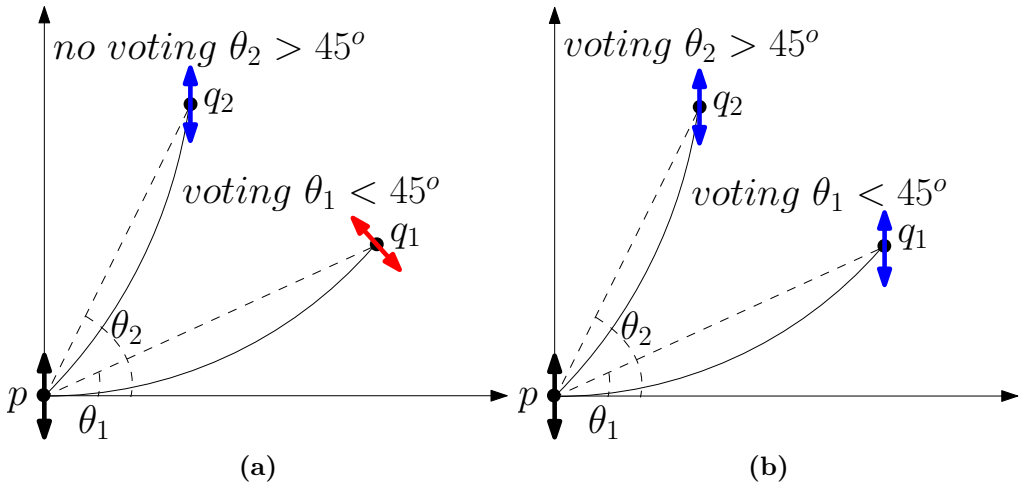


Figure 3.3: (left) Tensor voting propagates rotated versions of the original gradients to neighboring points if θ is less than or equal to 45^o. (right) The structure tensor interpreted as a voting process propagates the original gradients to all neighboring points.

3.4 Pre-segmentation of image pixels based on image gradients

In order to apply tensor voting to the problem of optical flow estimation, it is necessary to include additional constraints beyond the original restrictions regarding the value of θ and the size of the local neighborhood. In particular, it is necessary to ensure that pixels only propagate their gradients to other pixels that are likely to belong to the same region. Otherwise, that propagation is likely to blur discontinuities and, hence, to introduce undesired artifacts. Two additional discontinuity-preservation constraints have thus been enforced. The first constraint prevents the voting process if one of the pixels belongs to a textured region and the other to a homogeneous region. In turn, the second constraint prevents the voting process in case of two pixels that belong to the same type of region (homogeneous or textured) but one of them being in a moving region and the other in a stationary region.

As a consequence, it is necessary to define a fast and simple preprocessing stage that efficiently segments the original image into both homogeneous and textured regions on the one hand, and into moving and stationary regions on the other hand, in both cases based on the analysis of the spatio-temporal gradients of the image.

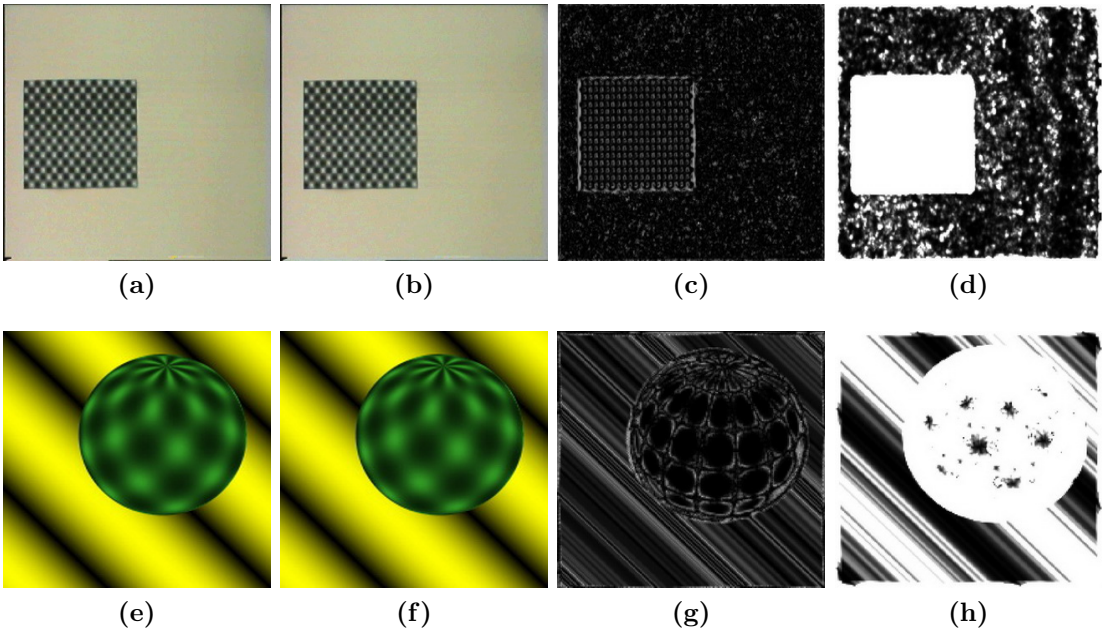


Figure 3.4: (a, b) a pair of consecutive images of a grid that is moving to the right. (c) Map of $\lambda_2 - \lambda_3$ of 3D tensor voting when $\sigma = 1.0$. (d) Map of $\lambda_2 - \lambda_3$ of the 3D structure tensor when $\sigma = 1.0$. (e, f) a pair of consecutive images of a textured ball that is rotating around its axis. (g) Map of $\lambda_2 - \lambda_3$ of 3D tensor voting when $\sigma = 1.0$. (h) Map of $\lambda_2 - \lambda_3$ of the 3D structure tensor when $\sigma = 1.5$.

3.4.1 Classification into homogeneous and textured regions

Let $\nabla_3 I = (I_x, I_y, I_t)^T$ be the spatio-temporal gradient of image $I(x, y, t)$, where $I_x = \partial_x I$, $I_y = \partial_y I$ and $I_t = \partial_t I$. The magnitude of the image gradient is:

$$\|\nabla_3 I\| = \sqrt{I_x^2 + I_y^2 + I_t^2}. \tag{3.6}$$

The signal-to-noise ratio (SNR)is estimated in order to determine what pixels belong to either homogeneous or textured regions:

$$SNR = 20 \log_{10}(\mu/\varsigma), \tag{3.7}$$

where μ is the mean of the gradient magnitudes within a square window (the window size has been set to 11x11 in this chapter) centered at every pixel, and ς is the standard deviation of those gradients. The value of SNR is estimated in order to determine what pixels belong to either homogeneous or textured regions. In particular, the gradients of pixels belonging to homogeneous regions will have a small standard deviation and, hence, a large SNR. In turn, pixels from textured

regions will have a big deviation and low SNR. A threshold τ equal to 25 dB[1], has been applied in order to distinguish between high and low SNR values, which correspond to homogeneous and textured regions respectively.

3.4.2 Classification into moving and stationary regions

In order to discriminate between moving and stationary regions, it is necessary to analyze the variation of image intensity along time, since the only intensity variation is assumed to be due to object motion and not to illumination changes, which are taken into account in the next step of the algorithm. This is done as follows. The angle δ between the spatio-temporal gradient $(I_x, I_y, I_t)^T$ and the temporal unit vector $(0, 0, 1)^T$ gives an indication of the contribution of the temporal gradient to the spatio-temporal gradient. When $(I_x, I_y, I_t)^T$ is parallel to direction t, $|\cos\delta|$ is close to one and the corresponding pixel is likely to belong to a moving region:

$$\cos\delta = \frac{I_t}{\|\nabla_3 I\| + \varepsilon}, \tag{3.8}$$

where $0 < \varepsilon \ll 1$ to avoid division by zero.

Since any small variation in the temporal gradients will cause $|\cos\delta|$ to be close to one, this condition is necessary to detect motion in homogenous regions. However, this condition is not sufficient for textured regions and edges, since the noise and discretization errors will also cause $|\cos\delta|$ to be close to one.

In order to avoid that problem of noise sensitivity, a second angle β is introduced according to the confidence measure proposed in Liou and Jain (1989):

$$\cos\beta = \frac{1}{\sqrt{1 + \|\nabla_3 I\|^2}}. \tag{3.9}$$

When the magnitude of the image gradient is very high, $|\cos\beta|$ is close to zero. However, this condition is not an indication of motion by itself due to the contribution of the spatial gradients. For instance, textured regions and edges will yield values of $|\cos\beta|$ close to zero even through they are still. Thus, the necessary and sufficient condition for a pixel belonging to a moving textured region is that $|\cos\delta|$ be close to one and $|\cos\beta|$ to zero. Based on the above three measures obtained from the spatio-temporal gradients $(SNR, \cos\delta, \cos\beta)$, the pixels of the given input image are classified into three broad classes: textured-moving regions (TM), homogeneous-moving regions (HM) and not moving regions (NM) using

[1]Define minimum SNR values. [Online] available http://www.wireless-nets.com/resources/tutorials/define_SNR_values.html.

3.10. Figure 3.5 shows an example of this segmentation given two consecutive images.

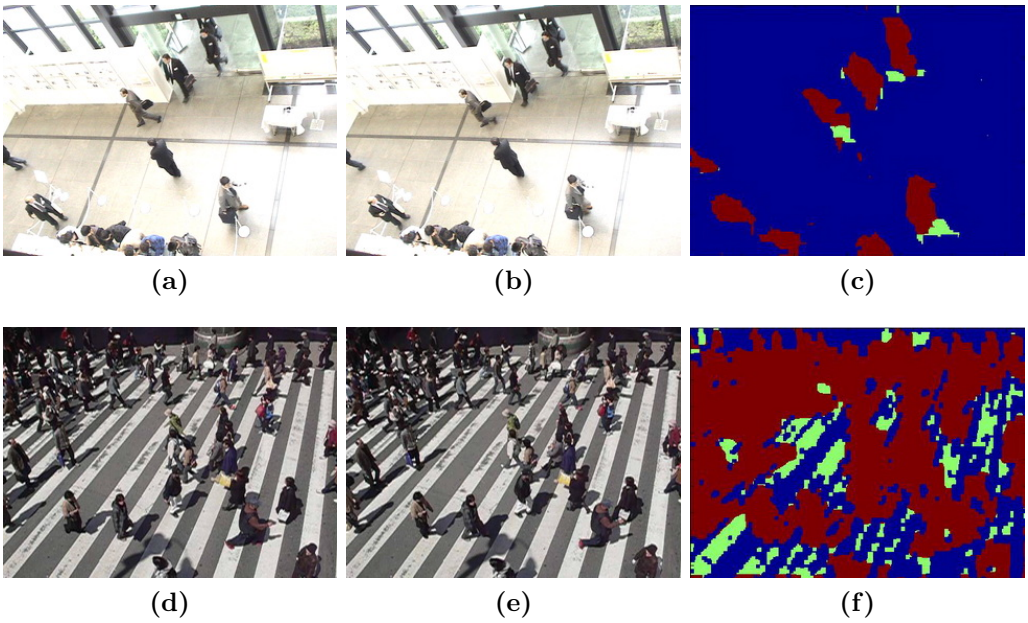


Figure 3.5: (a) Frame at time t in sequence OPEN-HOTEL. (b) Frame at time $t + dt$. (c) Classified pixels: red pixels are textured-moving regions, green pixels are homogeneous-moving regions and blue pixels are stationary (not moving) regions. (d) Frame at time t in sequence STREET-CROSS. (e) Frame at time $t + dt$. (f) Classified pixels: red pixels are textured-moving regions, green pixels are homogeneous-moving regions and blue pixels are stationary (not moving) regions.

$$I(x,y,t) = \begin{cases} TM & SNR \leq \tau, |\cos\delta| \approx 1, |\cos\beta| \approx 0 \\ HM & SNR > \tau, |\cos\delta| \approx 1 \\ NM & \text{otherwise.} \end{cases} \tag{3.10}$$

3.5 Smoothing of image gradients using tensor voting

Once the given image has been segmented as described in the previous section, tensor voting is applied in order to filter the image gradients of the pixels that belong to the TM (texture-moving) class on the one hand, and to the HM (homogeneous-moving) class on the other hand, as all those pixels correspond to moving regions. In textured regions, the necessary filtering window size (3σ) must be small enough

in order to preserve edges and texture. In homogeneous regions, however, the window size will be large enough in order to filter out image noise. In particular, the standard deviation of tensor voting applied to homogenous regions (σ_1) has experimentally been set to twice the standard deviation corresponding to textured regions (σ_2).

Tensor voting is only applied to pixels that belong to the same class. Thus, pixels belonging to textured-moving regions do not propagate their gradients to those belonging to homogeneous-moving regions and vice-versa. In turn, pixels belonging to the NM (not moving) class have gradients whose third component, I_t, is very small. Therefore, tensor voting does not bring any significant improvement on the result of optical flow at those regions, not being applied in order to save computations.

In order to preserve discontinuities within the pixels belonging to a same class and also to prevent the influence of objects belonging to the same class and moving along opposite directions, tensor voting is not applied between any pair of pixels whose gradients are significantly different, in particular, if the angle between both gradient vectors is above a predefined threshold ξ that has experimentally been set to 45 in this chapter, Figure 3.6.

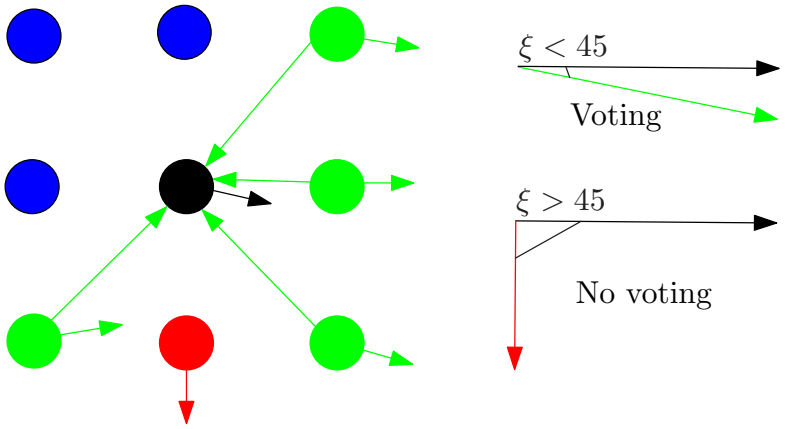


Figure 3.6: Textured voters (green) with $\xi < 45$ will cast votes to textured votee (black). Textured voter (red) with $\xi \geq 45$ will not cast votes to textured votee (black). Homogenous voter (blue) will not cast votes to textured votee (black).

The methodology described in the next section would require the computation of three image gradients for every pixel if the input images were gray-level images: $\nabla_3 I = (I_x, I_y, I_t)^T$, $\nabla_3 I_x = (I_{xx}, I_{xy}, I_{xt})^T$ and $\nabla_3 I_y = (I_{yx}, I_{yy}, I_{yt})^T$. However, when the original images are color images with three channels, which is the assumption in this chapter, the number of gradients is tripled. Thus, the following nine gradients are computed and independently filtered through separate tensor voting

processes: $\nabla_3 I^k = (I^k_x, I^k_y, I^k_t)^T$, $\nabla_3 I^k_x = (I^k_{xx}, I^k_{xy}, I^k_{xt})^T$ and $\nabla_3 I^k_y = (I^k_{yx}, I^k_{yy}, I^k_{yt})^T$, with $k \in \{0, 1, 2\}$ being the color channel.

Each tensor voting process applies the classical methodology introduced in the second section of this chapter and illustrated in Figure 3.3(left). Points p and q correspond to the spatial coordinates of two image pixels. The information associated with each pixel is a gradient vector encoded as a tensor. For instance, in case of $\nabla_3 I^k = (I^k_x, I^k_y, I^k_t)^T$, the initial tensor for a pixel p is defined as $\nabla_3 I^k (\nabla_3 I^k)^T$. After all pixels cast votes to their neighbors, the final tensor is computed for every pixel by applying (3.1). Finally, the filtered gradient $\hat{\nabla_3 I^k} = (\hat{I^k_x}, \hat{I^k_y}, \hat{I^k_t})^T$ corresponding to pixel p is defined as the eigenvector associated with the largest eigenvalue of $TV(p)$. This procedure is also applied to the rest of second-order gradients. Therefore, at the end of this stage, nine filtered gradient vectors are obtained for every pixel: $\hat{\nabla_3 I^k}$, $\hat{\nabla_3 I^k_x}$ and $\hat{\nabla_3 I^k_y}$, with $k \in \{0, 1, 2\}$.

3.6 Adapted optical flow model

This section describes the proposed adaptation to tensor voting of the variational optical flow technique originally proposed in Zimmer et al. (2009) and Bruhn et al. (2005). Variational global optical flow methods estimate the optical flow field by minimizing a functional that is constituted by some data constraints and a smoothness constraint:

$$E_I(u, v) = \int_\Omega [M(w, I) + \alpha V(\nabla_2 u, \nabla_2 v, I)] dx dy. \tag{3.11}$$

The data term $M(w, I)$ takes into account the data constraints, whereas the smoothness term $V(\nabla_2 u, \nabla_2 v, I)$ penalizes deviations from the smoothness of w. ∇_2 is the spatial gradient operator, $\nabla_2 = (\partial_x, \partial_y)^T$. The regularization parameter $\alpha > 0$ determines the weight of the smoothness term. The proposed definition of both the data term and the smoothness term is detailed below.

3.6.1 Data Term

The most common assumption in optical flow estimation is that the grey value of a moving object does not change with the motion of the object. This is refereed to as the gray-value constancy assumption or brightness constraint, which is formulated as:

$$I(x, y, t) - I(x + u, y + v, t + dt) = 0. \tag{3.12}$$

By expanding the second term in (3.12) through its first-order Taylor expan-

sion Brox et al. (2004), it yields the well-known linearized optical flow constraint:

$$I_x u + I_y v + I_t = 0, \tag{3.13}$$

where $u = dx/dt$ and $v = dy/dt$. That equation can be expressed in vector form as:

$$w^T \nabla_3 I = 0, \tag{3.14}$$

where $\nabla_3 I = (I_x, I_y, I_t)^T$. Then, the data term can be penalized in a least squares sense as:

$$M_1 = (w^T \nabla_3 I)^2. \tag{3.15}$$

This equation can be written as:

$$M_1 = w^T \nabla_3 I (\nabla_3 I)^T w = w^T S w, \tag{3.16}$$

where S is the motion tensor for the data term, $S = \nabla_3 I (\nabla_3 I)^T$, which is a positive semi-definite 33 symmetric matrix. S is integrated over a neighborhood of fixed size through a convolution of S with a Gaussian kernel K_ρ of standard deviation ρ. Thus, a modified $S_\rho = K_\rho * S$ is obtained that makes the method more robust against noise Bruhn et al. (2005):

$$M_1 = w^T (K_\rho * S) w = w^T S_\rho w. \tag{3.17}$$

In the present chapter, the above integrated motion tensor S_ρ is replaced by the result of applying tensor voting to the neighborhood of p illustrated in Section 3.5. Thus, a data term that ensures the grey value constancy assumption can be defined as:

$$M_1 = w^T tv(\nabla_3 I) w = w^T \hat{\nabla_3} I (\hat{\nabla_3} I)^T w, \tag{3.18}$$

with $\hat{\nabla_3} I$ being defined as described in the previous Section 3.5 and $tv(\chi) = \hat{\chi}(\hat{\chi})^T$.

As indicated above, the brightness (grey-value) constancy assumption does not cope with illumination changes. If such changes occur in the given image sequence, it is possible to circumvent the problem by considering that the gradient of an object does not change with the motion of the object. This yields the so-called gradient constancy assumption or gradient constraint, which is formulated as:

$$\nabla_3 I(x, y, t) - \nabla_3 I(x + u, y + v, t + dt) = 0. \tag{3.19}$$

Applying the first-order Taylor expression of the second term in (3.19) yields:

$$\begin{aligned} I_{xx} u + I_{xy} v + I_{xt} &= 0 \\ I_{xy} u + I_{yy} v + I_{yt} &= 0. \end{aligned} \tag{3.20}$$

A second data term dependent on the gradient constraint can then be written as:

$$M_2 = w^T \nabla_3 I_x + w^T \nabla_3 I_y. \tag{3.21}$$

After penalizing it in a least squares sense, it becomes:

$$M_2 = w^T \nabla_3 I_x (\nabla_3 I_x)^T w + w^T \nabla_3 I_y (\nabla_3 I_y)^T w. \tag{3.22}$$

The two tensors, $\nabla_3 I_x (\nabla_3 I_x)^T$ and $\nabla_3 I_y (\nabla_3 I_y)^T$, can also be integrated over a neighborhood of fixed size through the convolution with a Gaussian kernel K_ρ Bruhn et al. (2005):

In the present chapter, that Gaussian filtering is replaced by tensor voting:

$$\begin{aligned} M_2 &= w^T tv(\nabla_3 I_x) w + w^T tv(\nabla_3 I_y) w \\ &= w^T \hat{\nabla_3 I_x} (\hat{\nabla_3 I_x})^T w + w^T \hat{\nabla_3 I_y} (\hat{\nabla_3 I_y})^T w. \end{aligned} \tag{3.23}$$

This gradient constraint is robust to translations, whereas the brightness constraint defined above is suitable for more complicated types of motion. Therefore, Brox et al. (2004) proposed the combination of the two constraints in the data term while keeping the linearization. The combined data term using tensor voting is defined as:

$$M = M_1 + \gamma M_2, \tag{3.24}$$

where γ is the weight of the gradient constancy term in the data term. Therefore, the final tensor obtained as a result of the voting processes for $\nabla_3 I$, $\nabla_3 I_x$ and $\nabla_3 I_y$ is:

$$\begin{aligned} T &= tv(\nabla_3 I) + \gamma[tv(\nabla_3 I_x) + tv(\nabla_3 I_y)] \\ &= T_0 + \gamma T_{xy}. \\ T &= \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{12} & t_{22} & t_{23} \\ t_{13} & t_{23} & t_{33} \end{pmatrix}. \end{aligned} \tag{3.25}$$

In order to gain robustness against outliers, it is convenient to define the data term without the quadratic penalization intrinsic to the use of tensors. In particular, the non-quadratic function Ψ_M proposed in Zimmer et al. (2009) is applied:

$$\Psi_M(l^2) = \sqrt{l^2 + \zeta^2}, \tag{3.26}$$

where $\zeta \to 0$ is close to zero.

In addition, the data term is extended in order to be applicable to HSV color images. This copes with illumination changes, highlights, shading and shadow effects, as proposed in Zimmer et al. (2009). The final data term becomes:

$$M(w, I) = \sum_{k=1}^{3} \Psi_M(w^T T^k w), \tag{3.27}$$

where T^k is (3.25) applied to the $k-th$ color channel of I.

3.6.2 Smoothness term

Any value of w that minimizes the data term $M(w, I)$ defined in (3.27) would be a valid optical flow solution in agreement with both the grey-value and the gradient constancy assumptions. However, such a solution is not unique in general. That is, by only considering the spatio-temporal image gradients in a neighborhood of a pixel, it is not possible to estimate the direction and magnitude in which this pixel is moving in general. This is known as the aperture problem of optical flow. In order to obtain a unique solution, it is necessary to introduce some additional constraint in the functional that is minimized, that is, it is necessary to apply regularization. This is the goal of the smoothness term V included in (3.11). In particular, a quadratic smoothness term that penalizes the squared magnitude of the flow gradient was proposed in Horn and Schunck (1981):

$$V(\nabla_2 u, \nabla_2 v) = |\nabla_2 u|^2 + |\nabla_2 v|^2. \tag{3.28}$$

However, this function leads to an isotropic smoothing of the resulting flow field that does not preserve discontinuities. Thus, other smoothness terms more tolerant to discontinuities have been proposed in the literature. Image discontinuities are taken into account by image-driven methods, such as the anisotropic filter proposed by Nagel and Enkelmann (1986). That method regularizes the flow field along image edges, but not across them. Thus, image-driven filters are prone to generating artifacts in textured image regions.

In order to avoid the aforementioned problem, flow-driven regularization methods have been proposed in (Weickert and Schnórr, 2001). They preserve discontinuities in the flow field, not being affected by image textures. In particular, the anisotropic complementary smoothness term proposed in Zimmer et al. (2009) takes into account directional information from the constraints imposed in the data term. A robust penalization is performed across edges in order to reduce the smoothing effect in the direction where the data term gives the most information. Along edges, where the data term gives no information, a strong filling-in is performed by using a quadratic penalization. This smoothness term provides an effective combination of both image-driven and flow-driven behaviors.

Accordingly, the smoothness term proposed in Zimmer et al. (2009) has been adapted to tensor voting by using the directional information resulting from its process that suggested in this work by considering HSV color images.

Let R a regularization tensor defined as:

$$\begin{aligned} R &= \sum_{k=1}^{3}[tv(\nabla_2 I^k) + \gamma(tv(\nabla_2 I_x^k) + tv(\nabla_2 I_y^k))]. \\ R &= \begin{pmatrix} e_1 & e_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}. \end{aligned} \tag{3.29}$$

The smoothness term is finally defined as in Zimmer et al. (2009):

$$\begin{aligned} V(\nabla_2 u, \nabla_2 v, I) = \; & \Psi_V((e_1^T \nabla_2 u)^2 + (e_1^T \nabla_2 v)^2) + \\ & ((e_2^T \nabla_2 u)^2 + (e_2^T \nabla_2 v)^2), \end{aligned} \tag{3.30}$$

where e_1 and e_2 are the eigenvectors of the regularization tensor R in (3.29) corresponding to the eigenvalues $\lambda_1 \geq \lambda_2$, and Ψ_V is the non-convex regularizer proposed in Zimmer et al. (2009):

$$\Psi_V(l^2) = \zeta^2 \log\,(1 + \frac{l^2}{\zeta^2}), \tag{3.31}$$

using $\zeta > 0$ is a contrast parameter.

3.6.3 Implementation

The functional in (3.11) is minimized by solving the corresponding Euler-Lagrange equations:

$$\begin{aligned} \partial_u M - \alpha(\partial_x(\partial_{u_x} V) + \partial_y(\partial_{u_y} V)) &= 0 \\ \partial_v M - \alpha(\partial_x(\partial_{v_x} V) + \partial_y(\partial_{v_y} V)) &= 0, \end{aligned} \tag{3.32}$$

which can be rewritten based on the resulting tensor S of the voting process (3.25) as:

$$\begin{aligned} &\sum_{k=1}^{3} \Psi_M(w^T T^k w)(t_{11}^k + t_{12}^k + t_{13}^k) - \alpha X_{MV}(\nabla_2 u, \nabla_2 v) = 0 \\ &\sum_{k=1}^{3} \Psi_M(w^T T^k w)(t_{21}^k + t_{22}^k + t_{23}^k) - \alpha X_{MV}(\nabla_2 v, \nabla_2 u) = 0, \end{aligned} \tag{3.33}$$

where $X_{MV}(\nabla_2 a, \nabla_2 b)$ is:

$$X_{MV}(\nabla_2 a, \nabla_2 b) = divergence(D(e_1, e_2, \nabla_2 a, \nabla_2 b)\nabla_2 a),$$

with $D(e_1, e_2, \nabla_2 u, \nabla_2 v)$ defined in Zimmer et al. (2009) as:

$$D = \begin{pmatrix} e_1 \; e_2 \end{pmatrix} \begin{pmatrix} \Psi'_V((e_1^T \nabla_2 u)^2 + (e_1^T \nabla_2 v)^2) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}, \tag{3.34}$$

where Ψ'_V is the differential of Ψ_V.

The multi-scale, coarse-to-fine warping approach described in Brox et al. (2004) is used by most modern algorithms for estimating optical flow in order to support large displacements while keeping a good accuracy. This approach relies on estimating the optical flow in a Gaussian pyramid, where the top images are the original images after having been rescaled to a coarse scale. At each pyramid level, small flow increments are computed by solving the linear system in (3.33) through the Successive Over-Relaxation (SOR) solver with alternating liner relaxation[2].

In practical, the levels below are warped representations of the images based on the flow estimated at the preceding scale. This ensures that the small motion assumption considered in 3.11 remains valid, Figure 3.7. At each warping level, small flow increments are computed by solving the linear system in 3.27. Once a flow field is estimated (v_0), the past frames used (F_{1_0}, F_{2_0}) and the flow field calculated (v_0) in the coarse level are rescaled to the finer level. Then in this finer level, the second frame rescaled (F_{2_1}) warps towards the first frame (F_{1_1}). And, a new flow field is calculated between the first frame (F_{1_1}) and the warped frame (Fw_{2_1}). This value is then added to the previous flow rescaled (v_0) to get a new flow field (v_1) and the process is repeated until the maximum number of iterations is met.

To obtain the coarse representation of the pyramid, the input images have been rescaled by a factor ι. A standard image pyramid uses $\iota = 0.5$, whereas a larger factor $\iota = 0.9$ is actually used (according to Brox et al. (2004)) to obtain better results at the expense of an increased computational time. In this work, the number of pyramid levels is calculated as:

$$n \approx log_{10}(30/min(ht, wt))/log_{10}(0.9), \tag{3.35}$$

where constant 30 indicates the minimum image width or height in the pyramid and parameters ht and wt are the height and width of the original image, respectively. Moreover, in order to avoid aliasing, a low pass filter is applied to each level of the pyramid through a Gaussian convolution with a standard deviation equal to $0.5/\iota$.

[2]Black, Noel, Moore, Shirley, Successive over-relaxation method. Available: http://mathworld.wolfram.com/SuccessiveOverrelaxationMethod.html

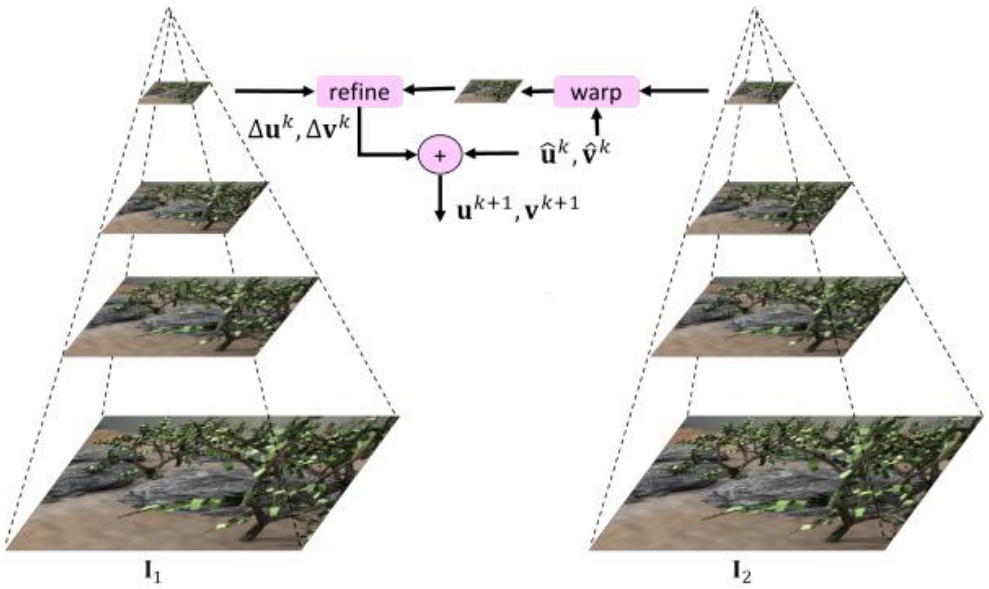


Figure 3.7: Coarse-to-fine approach.

3.7 Experimental results

The proposed technique, referred to as $IROF-TV$, has been implemented in Matlab and compared to both classical and state-of-the-art optical flow estimation methods. The results have been submitted to the Middlebury optical flow benchmark for external evaluation[3], (see Baker et al., 2010). The parameters of the proposed method for the Middlebury database have been set to: $\sigma_1 = 1.3$ for homogeneous regions, $\sigma_2 = 0.5$ for textured regions, $\alpha = 20$ and $\gamma = 100$. Moreover, the SNR threshold τ has been set to 25. The Matlab execution time for the Middlebury training sequences is around 250 seconds on a 3.2 GHZ Dual Core Pentium, by considering 640×480 images. At the time of submission (June 2011), the proposed method was ranked in the sixth position with respect to the Average End-Point Error (AEE), in the eighth position with respect to the Average Angular Error (AAE) and in the fourth position with respect to both the Average Interpolation Error (AIE) and the Average Normalized Interpolation Error (ANIE), Figure 3.8. Figure 3.9 shows some of the results for several Middlebury sequences.

The proposed technique has also been compared with other 12 datasets from the Middlebury database and 6 datasets from MIT datasets Liu et al. (2008), all of them with ground truth, see Figure 3.10. The AAE and AEE Baker et al. (2010)

[3]Middlebury datasets. Available: http://vision.middlebury.edu/flow/data

Average endpoint error	avg.	Army (Hidden texture) GT			im0			im1			Mequon (Hidden texture) GT			im0			im1		
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	5.2	0.09	6	0.23	3	0.07	3	0.16	2	**0.52**	1	0.13	3	0.22	2	0.46	3	0.17	4
Layers++ [33]	6.1	0.08	2	**0.21**	1	0.07	3	0.19	5	0.56	3	0.17	13	**0.20**	1	**0.40**	1	0.18	8
TC-Flow [34]	9.0	**0.07**	1	**0.21**	1	**0.06**	1	**0.15**	1	0.59	4	**0.11**	1	0.31	13	0.78	14	**0.14**	1
LSM [35]	9.0	0.08	3	0.23	3	0.07	3	0.22	13	0.73	15	0.18	15	0.28	8	0.64	6	0.19	13
Classic+NL [36]	9.9	0.08	2	0.23	3	0.07	3	0.22	13	0.74	16	0.18	13	0.29	10	0.65	10	0.19	13
IROF-TV	10.8	0.09	5	0.25	7	0.08	4	0.22	11	0.77	7	0.19	7	0.30	7	0.70	7	0.19	7

Average endpoint error	avg.	Wooden (Hidden texture) GT			im0			im1			Grove (Synthetic) GT			im0			im1		
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	5.2	0.17	11	0.93	12	0.09	4	0.65	4	0.98	5	0.43	4	**0.29**	1	**0.91**	1	0.26	3
Layers++ [33]	6.1	**0.13**	1	**0.58**	1	**0.07**	1	**0.48**	1	**0.70**	1	0.33	2	0.47	12	1.01	3	0.33	11
TC-Flow [34]	9.0	0.16	6	0.86	12	0.08	2	0.75	11	1.11	12	0.54	10	0.42	6	1.40	16	0.25	2
LSM [35]	9.0	0.14	2	0.70	2	0.09	4	0.66	6	0.97	4	0.48	7	0.50	15	1.06	5	0.33	11
Classic+NL [36]	9.9	0.15	3	0.73	4	0.09	4	0.64	3	0.93	3	0.47	5	0.52	17	1.12	7	0.33	11
IROF-TV	10.8	0.18	93	0.11	73	1.04	56	0.44	69	0.31	09	0.11	50	1.08	73				

Average endpoint error	avg.	Urban (Synthetic) GT			im0			im1			Yosemite (Synthetic) GT			im0			im1		
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	5.2	**0.29**	1	**0.91**	1	0.26	3	0.11	7	0.13	9	0.17	6	**0.46**	1	**0.88**	1	0.72	7
Layers++ [33]	6.1	0.47	12	1.01	3	0.33	11	0.15	25	0.14	18	0.24	24						
TC-Flow [34]	9.0	0.42	6	1.40	16	0.25	2	0.11	7	0.12	4	0.29	38	0.62	13	1.35	14	0.93	25
LSM [35]	9.0	0.50	15	1.06	5	0.33	11	0.15	25	0.12	4	0.29	38	0.50	4	0.99	3	0.73	10
Classic+NL [36]	9.9	0.52	17	1.12	7	0.33	11	0.16	31	0.13	9	0.29	38	0.49	3	0.98	2	0.74	12
IROF-TV	10.8	0.44	69	1.69	81	0.31	09	0.09	1	**0.11**	1	0.12	1	0.50	1	1.08	1	0.73	1

(a)

Average angle error	avg.	Army (Hidden texture) GT			im0			im1			Mequon (Hidden texture) GT			im0			im1		
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	6.5	3.32	7	8.76	5	2.85	8	**2.18**	1	7.47	3	1.85	4	2.77	3	6.95	3	2.06	9
Layers++ [33]	6.6	3.11	2	8.22	2	2.79	6	2.43	5	**7.02**	1	2.24	11	**2.43**	1	**5.77**	1	2.18	12
LSM [35]	8.0	3.12	3	8.62	3	2.75	4	3.00	10	10.5	13	2.44	16	3.43	7	8.85	8	2.35	14
TC-Flow [33]	8.6	**2.91**	1	**8.00**	1	2.34	2	**2.18**	1	8.77	7	**1.52**	1	3.84	13	10.7	14	**1.49**	1
Classic+NL [36]	9.5	3.20	5	8.72	4	2.81	7	3.02	17	10.6	14	2.44	16	3.46	9	8.84	7	2.38	16
SimpleFlow [37]	11.2	3.35	8	9.20	9	2.98	13	3.18	19	10.7	16	2.71	24	5.06	20	12.6	15	2.70	20
OF-MoI [38]	12.8	3.19	4	8.76	5	2.77	5	3.84	28	14.0	31	2.69	23	3.44	8	8.78	6	2.39	17
MDP-Flow [39]	13.6	3.48	12	9.46	14	3.10	16	2.45	6	7.36	2	2.41	14	3.21	5	8.31	5	2.78	22
IROF-TV	13.6	3.40	11	9.29	11	2.95	11	2.99	11	11.1	11	2.53	11	3.81	11	9.81	11	2.44	11

Average angle error	avg.	Wooden (Hidden texture) GT			im0			im1			Grove (Synthetic) GT			im0			im1		
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	6.5	3.25	14	17.3	19	1.59	12	2.87	6	3.73	8	2.32	5	3.15	3	11.1	8	2.65	3
Layers++ [33]	6.6	**2.13**	1	**9.71**	1	**1.15**	1	**2.35**	1	**3.02**	1	**1.96**	1	3.81	12	11.4	9	3.22	15
LSM [35]	8.0	2.66	3	13.6	3	1.44	2	2.82	4	3.68	3	2.36	7	3.38	6	9.41	3	2.81	9
TC-Flow [33]	8.6	3.13	10	16.6	14	1.46	4	2.78	3	3.73	5	**1.96**	1	3.08	2	11.4	9	2.66	4
Classic+NL [36]	9.5	2.78	4	14.3	4	1.46	4	2.83	5	3.68	3	2.31	4	3.40	7	**9.09**	1	2.76	8
SimpleFlow [37]	11.2	2.95	6	15.1	7	1.58	10	2.91	7	3.79	7	2.47	11	3.59	11	9.49	4	2.99	11
OF-MoI [38]	12.8	2.98	7	15.8	10	1.53	7	2.96	8	3.89	9	2.34	6	3.40	7	9.30	2	2.73	6
MDP-Flow [39]	13.6	3.18	13	17.8	20	1.70	14	3.03	9	3.87	8	2.60	15	3.43	9	12.6	13	2.81	9
IROF-TV	13.6	3.25	14	16.9	18	1.78	17	3.27	14	4.10	14	2.93	14	4.47	20	16.0	20	3.53	17

Average angle error	avg.	Urban (Synthetic) GT			im0			im1			Yosemite (Synthetic) GT			im0			im1		
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	6.5	3.81	12	11.4	9	3.22	15	2.74	21	4.01	25	2.35	20	1.45	2	**3.05**	1	1.79	6
Layers++ [33]	6.6	3.81	12	11.4	9	3.22	15	2.74	21	4.01	25	2.35	20	1.45	2	**3.05**	1	1.79	6
LSM [35]	8.0	3.38	6	9.41	3	2.81	9	2.69	20	3.52	9	2.84	27	1.59	3	3.38	3	1.80	7
TC-Flow [33]	8.6	3.08	2	11.4	9	2.66	4	1.94	6	3.43	5	3.20	40	3.06	15	7.04	15	4.08	33
Classic+NL [36]	9.5	3.40	7	**9.09**	1	2.76	8	2.87	26	3.82	15	2.86	31	1.67	5	3.53	4	2.26	11
SimpleFlow [37]	11.2	3.59	11	9.49	4	2.99	11	2.39	13	3.46	6	2.24	18	1.60	4	3.56	5	1.57	3
OF-MoI [38]	12.8	3.40	7	9.30	2	2.73	6	2.83	24	3.92	20	2.98	37	2.46	10	4.98	10	2.89	16
MDP-Flow [39]	13.6	3.43	9	12.6	13	2.81	9	2.19	10	3.88	18	1.60	6	4.13	27	9.96	29	3.86	30
IROF-TV	13.6	4.47	20	16.0	20	3.53	17	1.70	1	3.21	1	1.12	1	1.91	1	4.75	1	2.19	1

(b)

Average interpolation error	avg.	Mequon (Hidden texture) im0		GT		im1		Schefflera (Hidden texture) im0		GT		im1		Urban (Synthetic) im0		GT		im1	
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	7.0	2.86	2	5.31	2	1.20	2	3.46	3	5.07	3	1.31	4	**3.49**	1	**5.34**	1	1.47	3
CBF [40]	9.8	**2.83**	1	**5.20**	1	1.23	17	3.97	19	5.79	19	1.56	21	3.62	3	5.47	2	1.60	9
Aniso. Huber-L1 [41]	11.8	2.95	7	5.44	6	1.24	22	4.42	34	6.27	35	1.67	33	3.79	11	5.70	7	1.50	4
CLG-TV [42]	12.0	2.94	6	5.45	7	1.25	27	4.26	29	6.17	27	1.60	28	3.68	7	5.73	8	1.73	13
IROF-TV	12.1	3.07	11	5.91	11	1.23	1	3.71	11	5.47	11	1.40	1	3.70	11	6.27	11	1.58	1

Average interpolation error	avg.	Teddy (Stereo) im0		GT		im1		Backyard (High-speed camera) im0		GT		im1		Basketball (High-speed camera) im0		GT		im1	
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	7.0	5.40	8	7.95	14	3.41	31	10.2	2	12.7	2	3.61	14	6.12	8	11.8	7	2.38	19
CBF [40]	9.8	**5.21**	1	**7.12**	1	3.29	17	**10.1**	1	**12.6**	1	3.62	18	5.97	4	11.5	4	2.31	7
Aniso. Huber-L1 [41]	11.8	5.31	9	7.42	4	3.24	15	11.1	18	14.0	22	3.61	14	5.91	3	11.4	3	**2.24**	1
CLG-TV [42]	12.0	5.36	5	7.41	3	3.32	24	11.1	18	14.0	22	3.57	5	5.88	2	11.3	2	2.26	2
IROF-TV	12.1	5.25	7.60		3.17		11.0	13.9		4.47		6.37	12.4		2.30				

Average interpolation error	avg.	Dumptruck (High-speed camera) im0		GT		im1		Evergreen (High-speed camera) im0		GT		im1	
	rank	all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	7.0	7.48	3	17.1	3	1.51	2	**7.32**	1	**11.4**	1	1.75	31
CBF [40]	9.8	7.76	14	17.8	13	1.61	14	7.60	5	11.9	5	1.76	36
Aniso. Huber-L1 [41]	11.8	7.60	6	17.3	6	1.51	2	7.62	7	11.9	5	1.73	18
CLG-TV [42]	12.0	7.58	5	17.0	2	1.57	12	7.75	11	12.1	11	1.72	13
IROF-TV	12.1	7.79	17.9		**1.50**		7.63	11.9		**1.66**			

(c)

Average normalized interpolation error	avg.	Mequon (Hidden texture) im0		GT		im1		Schefflera (Hidden texture) im0		GT		im1		Urban (Synthetic) im0		GT		im1	
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	7.4	**0.58**	1	**0.71**	1	**0.64**	1	0.63	3	0.87	3	**0.59**	1	0.92	5	1.37	7	0.85	7
CLG-TV [42]	13.3	0.63	23	0.86	21	0.66	22	0.81	31	1.12	36	0.66	22	0.96	8	1.43	9	0.96	15
Aniso. Huber-L1 [41]	14.4	0.62	13	0.80	10	0.66	22	0.84	34	1.13	32	0.66	25	1.03	14	1.44	10	0.93	14
IROF-TV	14.9	0.62	0.84		0.65		0.67	0.92		0.60		0.92	1.49		0.79				

Average normalized interpolation error	avg.	Teddy (Stereo) im0		GT		im1		Backyard (High-speed camera) im0		GT		im1		Basketball (High-speed camera) im0		GT		im1	
	rank	all		disc		untext		all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	7.4	0.98	1	**0.95**	1	1.15	4	1.13	18	1.60	21	1.08	15	0.68	5	1.23	5	0.68	8
CLG-TV [42]	13.3	0.97	8	1.03	5	1.25	15	1.06	19	1.08	20	1.15	4	**1.02**	1	**1.25**	1	1.04	2
Aniso. Huber-L1 [41]	14.4	0.97	8	1.03	5	1.26	20	1.06	19	1.09	21	1.15	4	1.08	8	1.46	10	**1.03**	1
IROF-TV	14.9	**0.94**	1.02		1.22		1.18	1.28		1.70		1.12	1.58		1.05				

Average normalized interpolation error	avg.	Dumptruck (High-speed camera) im0		GT		im1		Evergreen (High-speed camera) im0		GT		im1	
	rank	all		disc		untext		all		disc		untext	
MDP-Flow2 [32]	7.4	0.68	5	1.23	5	0.68	8	**0.75**	1	**1.06**	1	0.64	18
CLG-TV [42]	13.3	0.63	1	**1.09**	1	0.66	3	0.97	23	1.45	23	0.63	2
Aniso. Huber-L1 [41]	14.4	0.64	2	1.12	2	0.66	3	0.99	33	1.48	36	0.63	2
IROF-TV	14.9	0.79	1.57		0.70		0.85	1.24		0.64			

(d)

Figure 3.8: Results of Middlebury benchmark. The proposed method (IROF-TV) is highlighted. (a) Topmost methods according to the Average End-Point Error (AEE). (b) Topmost methods according to the Average Normalized Interpolation Error (AAE). (c) Topmost methods according to the Average Normalized Interpolation Error (AIE). (d) Topmost methods according to the Average Normalized Interpolation Error (ANIE). Snapshots from Middlebury benchmark website.

between the ground-truth and the results obtained by Brox et al. (2004), Bruhn et al. (2005), Zimmer et al. (2009) and the proposed method have been calculated. Table 7.1 shows the *AAE*, in turn table 3.2 shows *AEE* for some of the image se-

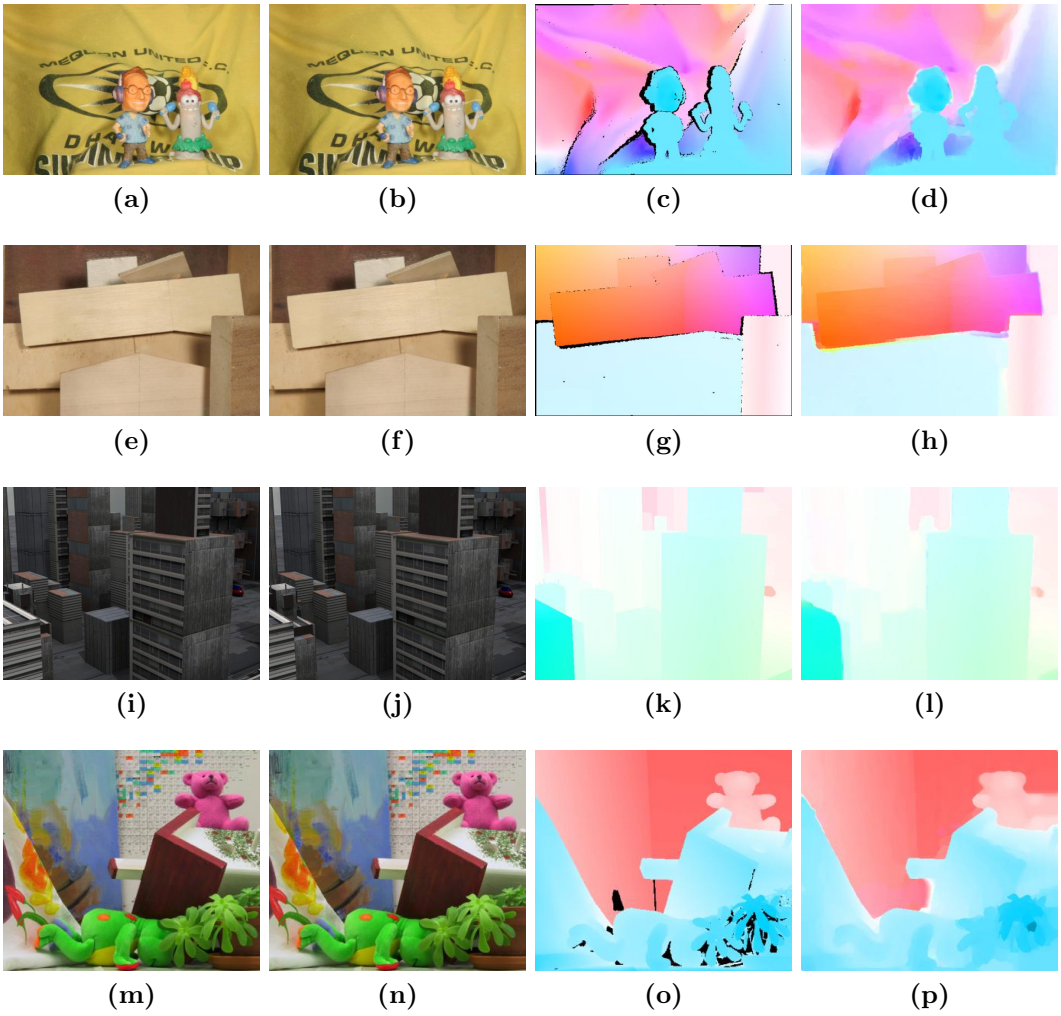


Figure 3.9: Results for some Middlebury sequences with corresponding ground-truth. (1st column) and (2nd column) Frames 10 and 11. (3rd column) Ground-truths (black points correspond to pixels without available ground-truth). (4th column) Optical flow fields obtained with the proposed approach.

quences from Middlebury datasets (Urban3, Dimetrodon) and from MIT datasets (Car, Table). Qualitative results are shown in Figure 3.10. Parameter $\gamma = 10$ is a constant for all the compared methods, while α, σ_1 and σ_2 are experimentally tuned parameters. The experimental parameters of the proposed approach have been set to: $\alpha = 15$, $\sigma_1 = 2.0$ and $\sigma_2 = 0.5$ for the Middlebury sequences, and $\alpha = 25$, $\sigma_1 = 1.5$ and $\sigma_2 = 0.6$ for the MIT sequences. In addition, the SNR threshold τ has been set to 25.

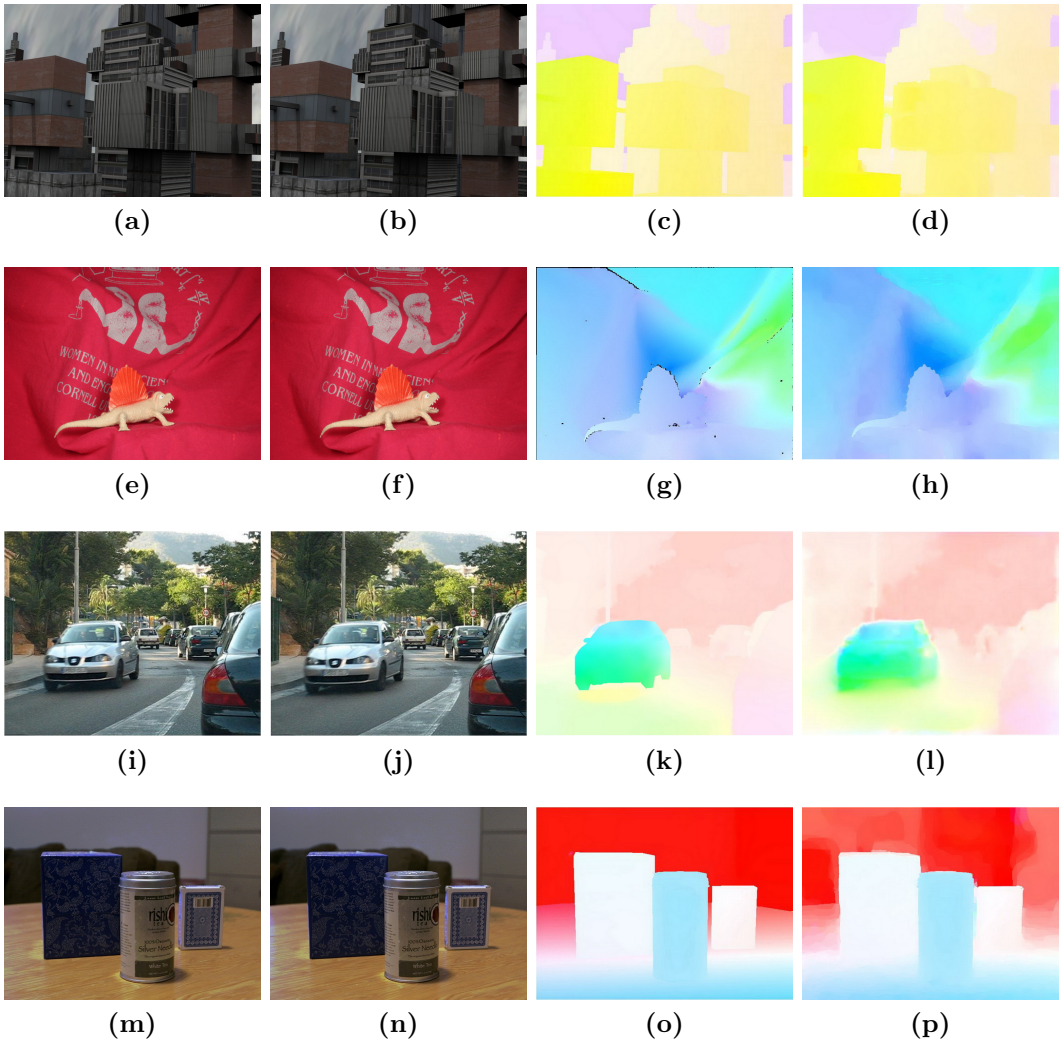


Figure 3.10: Results for some Middlebury and MIT sequences with associated ground-truths. (1st column) and (2nd Column) Two consecutive frames. (3th Column) Ground-truths. (4th Column) Optical flow fields obtained with the proposed approach.

The consistency of the computed optical flow has been tested by calculating the histogram of Angular Error (AE) and the histogram of Endpoint Error (EE) between the computed optical flow and the ground-truth. Figure 3.11 shows the histograms of AE and EE for two different sequences from MIT datasets: Car and Table. The histograms of AE and EE have been calculated for both Zimmer et al. (2009) and the proposed method. As can be observed in Figure 3.11, the proposed method yields a result more similar to the ground-truth than Zimmer

Methods	URBAN	Dim	CAR	TABLE
Brox et al. (2004)	0.259	0.232	0.143	0.353
Bruhn et al. (2005)	0.565	0.352	0.226	0.877
Zimmer et al. (2009)	0.144	0.195	0.209	0.347
Proposed	**0.141**	**0.125**	**0.094**	**0.225**

Table 3.1: AEE for some sequences from the Middlebury and MIT databases.

Methods	URBAN	Dim	CAR	TABLE
Brox et al. (2004)	03.882	04.487	04.871	04.121
Bruhn et al. (2005)	4.264	05.270	04.659	04.415
Zimmer et al. (2009)	03.105	03.358	03.153	03.537
Proposed	**02.824**	**02.934**	**02.415**	**03.4721**

Table 3.2: AAE for some sequences from the Middlebury and MIT databases.

et al. (2009).

Another qualitative comparison has been carried out by adding Gaussian noise with zero mean and different standard deviations ($\sigma_n = 0 : 25$). Figure 3.12 shows the AAE and AEE between the ground-truth and the flow fields obtained with both Zimmer et al. (2009) and the proposed technique for the Urban3 sequence under different noise levels. As can be seen in Figure 3.12, the AEE does not experimentally increase with the different levels of noise. In turn, the AAE is not affected at low levels of noise, $\sigma_n < 10$, but it deteriorates at the highest levels, $\sigma_n > 10$. In addition, the result obtained with Zimmer et al. (2009) is much more sensitive to noise than the one obtained with the proposed method.

Figure 3.13 visually compares the results for three examples of small objects in the Army sequence with the optical flow methods that have a rank in AEE better than the one achieved by the proposed method in Middlebury benchmark (Figure 3.8): Xu et al. (2012), Sun et al. (2010b), Volz et al. (2011), Jia et al. (2011), Sun et al. (2010a). Figure 3.14 shows that the flow estimated with the proposed method contains more motion details than the other techniques, and detects the contours of small objects better than them. Furthermore, Figure 3.16 visually compares the flow fields estimated for two regions within the Yosemite sequence with the same aforementioned optical flow methods, whose results are shown in Figure 3.15. The flow field estimated with the proposed technique has a smoother transition between different regions in the flow field.

The proposed method has also been tested upon two real image sequences: OPEN-HOTEL (Figure 3.17) and STREET-CROSS (Figure 3.18). Both sequences

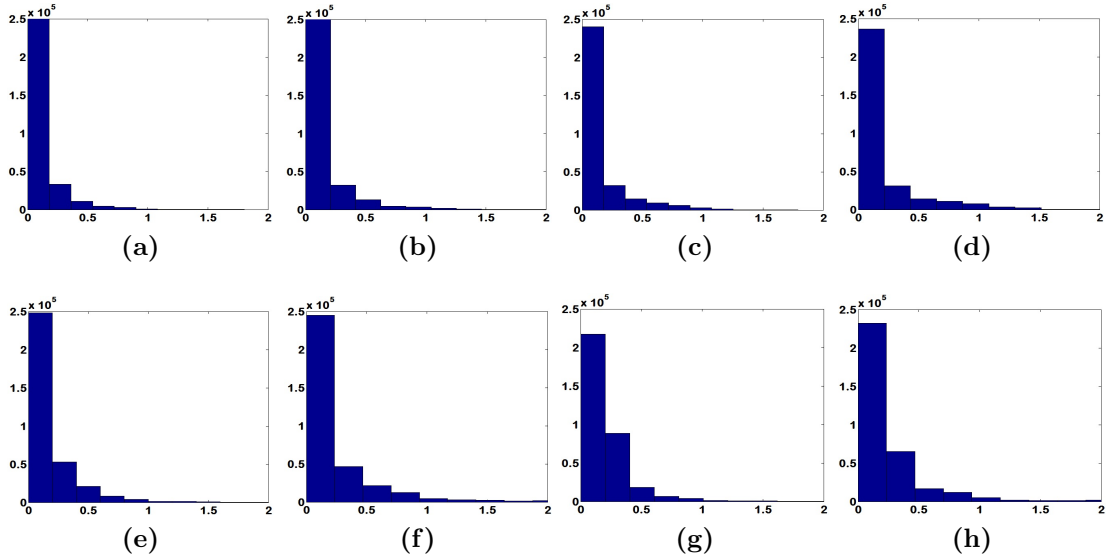


Figure 3.11: (a) Histogram of AE for the CAR sequence with the proposed method. (b) Histogram of EE for the CAR sequence with the proposed method. (c) Histogram of AE for the CAR sequence with Zimmer et al. (2009). (d) Histogram of EE for the CAR sequence with Zimmer et al. (2009). (e) Histogram of AE for the TABLE sequence with the proposed method. (f) Histogram of EE for the TABLE sequence with the proposed method. (g) Histogram of AE for the TABLE sequence with Zimmer et al. (2009). (h) Histogram of EE for the TABLE sequence with Zimmer et al. (2009).

are unstructured crowded scenes. The parameters of the proposed method have experimentally been set to: $\sigma_1 = 1.5$, $\sigma_2 = 0.5$, $\alpha = 20$, $\gamma = 75$ and $\tau = 25$. Figure 3.17 (row 1) shows four consecutive frames from the OPEN-HOTEL sequence. In Figure 3.17(row 2), the resulting dense optical flow fields for those frames are shown. In order to visualize the flow fields, a color coding has been used such that color encodes the flow direction, while brightness indicates the magnitude, as shown in Figure 3.17 (h). Figure 3.18 shows four consecutive frames from the STREET-CROSS sequence and their resulting optical flow fields. As can be appreciated, the optical flow fields obtained with the proposed technique are very accurate both in direction and magnitude. Although, some artifacts are present, the structure and appearance of the individuals in the scene is remarkable. Furthermore, the motion boundaries are fairly sharp. Accordingly, those flow fields could directly be used for image motion segmentation as a realistic application.

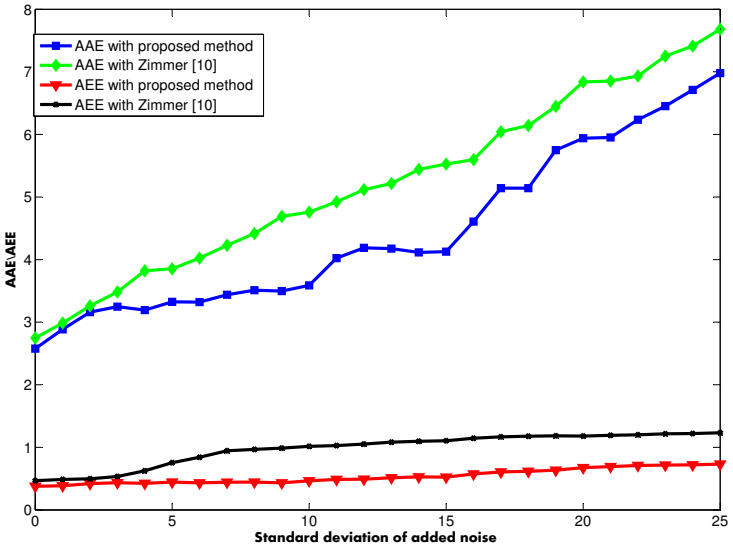


Figure 3.12: Stability of the proposed method for different noise levels.

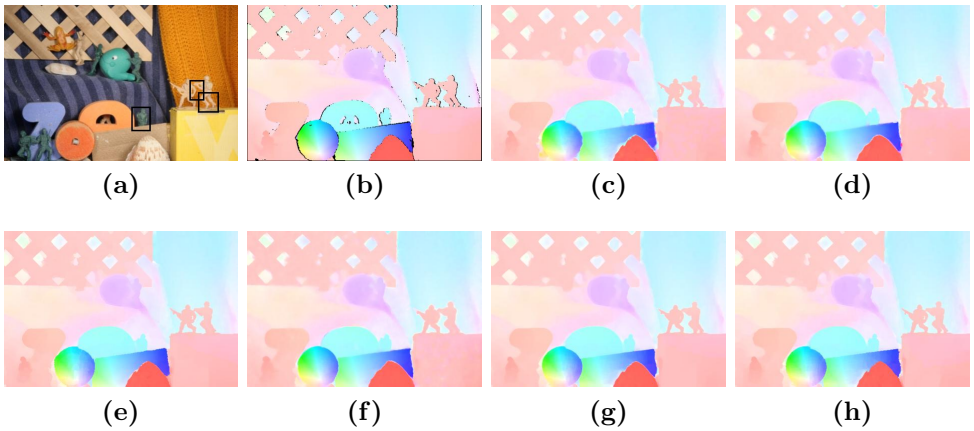


Figure 3.13: Resulting flow fields with the proposed method and the techniques with better AEE according to Middlebury benchmark. (a) Reference image in the Army sequence. (b) Ground-truth. (c) Flow field obtained with the proposed method. (d) Flow field obtained with Xu et al. (2012). (e) Flow field obtained with Sun et al. (2010b). (f) Flow field obtained with Volz et al. (2011). (g) Flow field obtained with Jia et al. (2011). (h) Flow field obtained with Sun et al. (2010a).

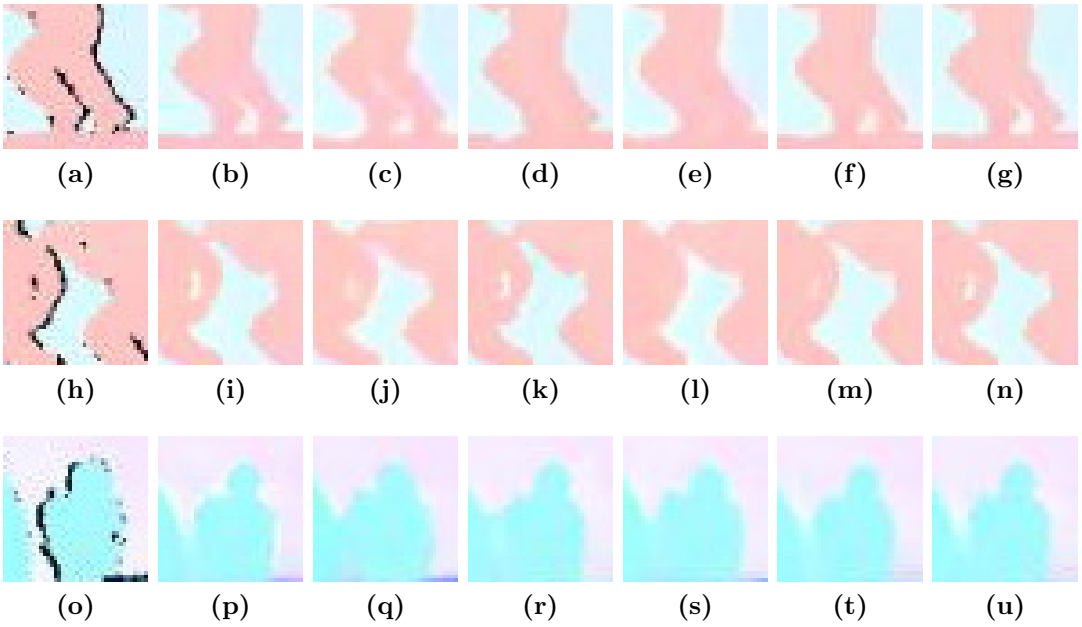


Figure 3.14: Detail images of the resulting optical flow for the Army sequence with the proposed method and the techniques with better AEE according to Middelbury benchmark. (a, h, o) Ground-truth (b, i, p) Proposed method. (c, j, q) Method proposed in Xu et al. (2012). (d, k, r) Method proposed in Sun et al. (2010b). (e, l, s) Method proposed in Volz et al. (2011). (f, m, t) Method proposed in Jia et al. (2011). (g, n, u) Method proposed in Sun et al. (2010a).

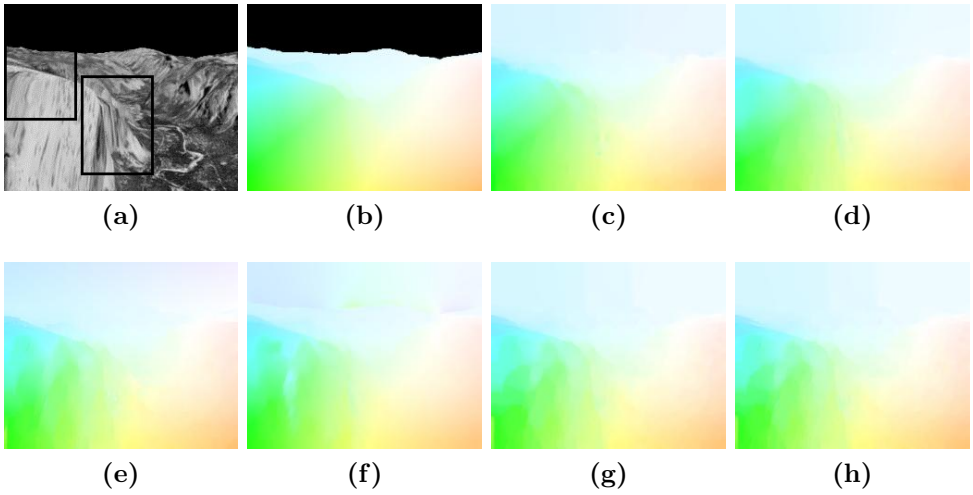


Figure 3.15: Resulting flow fields with the proposed method and the techniques with better AEE according to Middlebury benchmark. (a) Reference image in the Yosemite sequence. (b) Ground-truth. (c) Flow field obtained with the proposed method. (d) Method proposed in Xu et al. (2012). (d, k, r) Method proposed in Sun et al. (2010b). (e, l, s) Method proposed in Volz et al. (2011). (f, m, t) Method proposed in Jia et al. (2011). (g, n, u) Method proposed in Sun et al. (2010a).

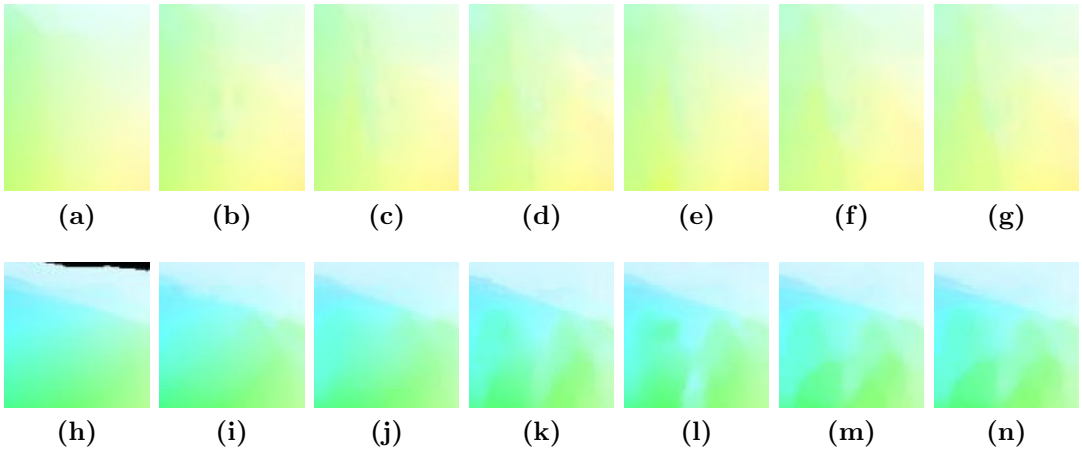


Figure 3.16: Detail images of the resulting optical flow for the Yosemite sequence with the proposed method and the techniques with better AEE according to Middlebury benchmark. (a, h) Ground-truth (b, i) Proposed method. (c, j) Method proposed in Xu et al. (2012). (d, k) Method proposed in Sun et al. (2010b). (e, l) Method proposed in Volz et al. (2011). (f, m) Method proposed in Jia et al. (2011). (g, n) Method proposed in Sun et al. (2010a).

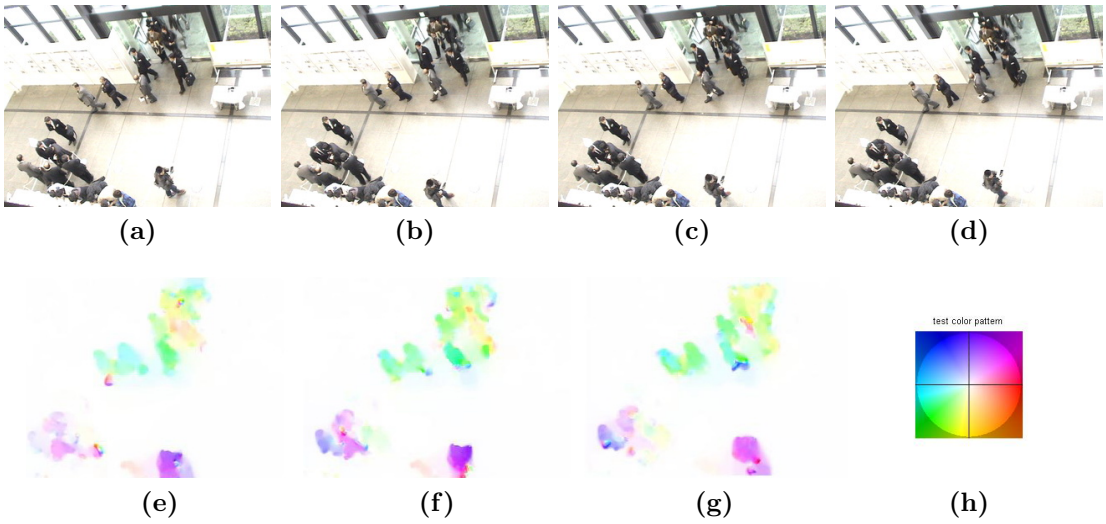


Figure 3.17: (a, b, c, d) Application of the proposed method to four consecutive frames (625, 626, 627 and 628) of the OPEN-HOTEL sequence. (e) Resulting optical flow field between frames 625 and 626. (f) Resulting optical flow field between frames 626 and 627. (g) Resulting optical flow field between frames 627 and 628. (h) Color coding chart.

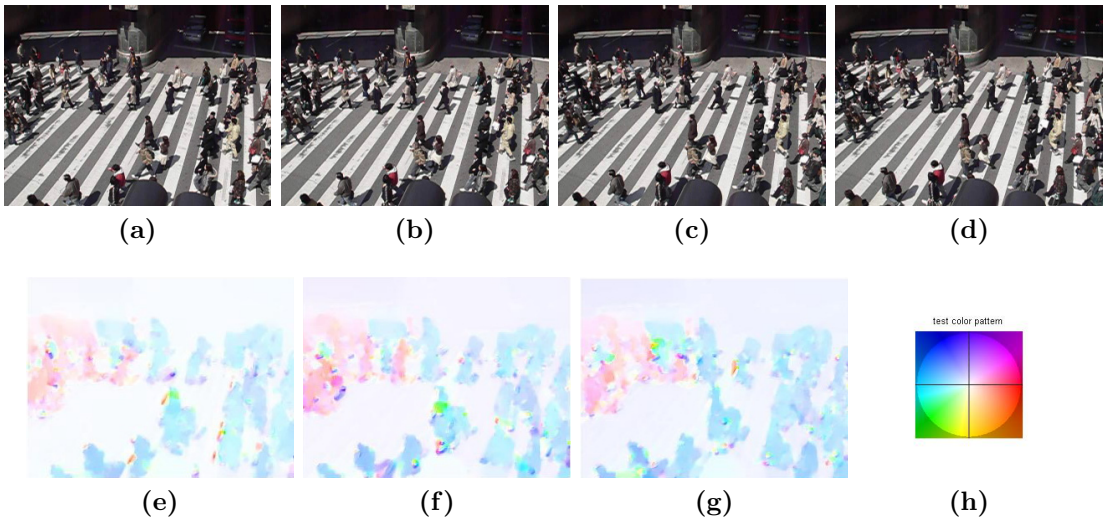


Figure 3.18: (a, b, c, d) Application of the proposed method to four consecutive frames (63, 64, 65 and 66) of the STREET-CROSS sequence. (e) Resulting optical flow field between frames 63 and 64. (f) Resulting optical flow field between frames 64 and 65. (g) Resulting optical flow field between frames 65 and 66. (h) Color coding chart.

Chapter 4

Robust Optical Flow Estimation Based on Stick Tensor Voting

In order to get a robust video surveillance systems, a fast and robust optical flow approach used for a motion detection is required. Variational optical flow techniques allow the estimation of flow fields from spatio-temporal derivatives. They are based on minimizing a functional that contains a data term and a regularization term. Recently, numerous approaches have been presented for improving the accuracy of the estimated flow fields. Among them, tensor voting has been shown to be particularly effective in the preservation of flow discontinuities. This chapter presents an adaptation of the data term by using anisotropic stick tensor voting in order to gain robustness against noise and outliers with significantly lower computational cost than (full) tensor voting that proposed in Chapter 3. In addition, an anisotropic complementary smoothness term depending on directional information estimated through stick tensor voting is utilized in order to preserve discontinuity capabilities of the estimated flow fields. Finally, a weighted non-local term that depends on both the estimated directional information and the occlusion state of pixels is integrated during the optimization process in order to denoise the final flow field. The proposed approach yields state-of-the-art results on the Middlebury benchmark.

The rest of this chapter is organized as follows. Section 4.1 introduces to the related work. The complexity of using full tensor voting is discussed in Section 4.2. The adapted variational optical flow model based on stick tensor voting is detailed in Section 4.3. The improved model based on a weighted non-local term using the saliency of image gradients and the occlusion state of pixels is described in Section 4.4. Finally, experimental results are shown and discussed in Section 4.5, including a comparison with state-of-the-art optical flow methods using the Middlebury benchmark.

4.1 Introduction

Optical flow is an important visual cue for representing the motion information associated with the objects present in a given sequence of images. Many methods have been proposed for estimating an optical flow field based on differential techniques Baker et al. (2010). They are able to estimate flow fields even in regions where other techniques would generate voids. These techniques can be classified into local and global methods. Local methods (*e.g.*, Lucas and Kanade (1981)) assume a uniform optical flow around each pixel and estimate it by applying least squares. These methods yield flow fields except in homogeneous regions with null gradients. In turn, global methods (*e.g.*, Horn and Schunck (1981)) minimize a function that forces the smoothness of the resulting flow field over the whole image, thus yielding dense flow fields even in homogeneous regions. However, they are more sensitive to noise since they do not filter the input gradients.

Furthermore, the most recent schemes apply a coarse-to-fine approach in order to cope with large motions and to improve the accuracy of the estimated flow fields as explained in Chapter 3. This approach estimates the optical flow fields using Gaussian pyramids corresponding to the input images, the latter being the fine scale images in those pyramids. Most top ranking methods in the Middlebury benchmark[1] apply differential techniques with a coarse-to-fine approach.

Related work

Recently, Brox et al. (2004) combined the classical brightness constancy assumption introduced in Horn and Schunck (1981) with the higher-order gradient constancy assumption Schnorr (1994) and a coarse-to-fine approach to improve the accuracy of the estimated flow fields, as well as to cope with illumination changes. However, the method proposed in Brox et al. (2004) does not filter the input gradients, yielding flow fields sensitive to noise. Alternatively, Bruhn et al. (2005) suggested a combination of the local method proposed in Lucas and Kanade (1981) by applying the 2D Gaussian filtering with structure tensors suggested in Lucas and Kanade (1981)Bigun et al. (1991) with the global method proposed in Horn and Schunck (1981) in order to obtain accurate flow fields less sensitive to image noise. Unfortunately, Gaussian filters are isotropic and do not preserve discontinuities. This may lead to the propagation of incorrect information at pixels located between different image regions, such as object boundaries, or between objects that move along different directions.

More recently, Zimmer et al. (2009) presented a robust data term that uses the HSV color space to avoid illumination, shading and shadow conditions. Moreover,

[1] Middlebury's website: vision.middlebury.edu/flow.

they suggested an anisotropic complementary regularization term that merges an image-driven and a flow-driven regularizer Nagel and Enkelmann (1986) and Weickert and Schnórr (2001) in order to preserve flow discontinuities. In addition, Zimmer et al. (2011) combined the variational optical flow technique proposed in Zimmer et al. (2009) with a simple method for automatically determining the optimal smoothness weight. The approaches proposed in Zimmer et al. (2009)Zimmer et al. (2011) generate accurate and dense optical flow fields, but they still suffer from inaccurate object boundaries, since the regularization term introduced in Zimmer et al. (2009)Zimmer et al. (2011) depends on the integration of local information through Gaussian convolution, which blurs object boundaries and small details.

Furthermore, Sand and Teller (2008) proposed to combine long-term feature tracking with dense flow fields. Their optical flow approach uses the same global smoothness value suggested in Brox et al. (2004) and adds a local parameter that specifies the smoothness criterion in a gradient dependent manner, in such a way that image regions with edges and texture will have lower local smoothness than textureless regions, thus preserving flow field discontinuities.

In turn, Chapter 3 (Rashwan et al. (2011) and Rashwan et al. (2012)) proposed a discontinuity-preserving filtering stage based on tensor voting with an adaptation of the complementary regularization term proposed in Zimmer et al. (2009) based on directional information obtained from tensor voting. This technique illustrated in Chapter 3 estimates accurate dense optical flow fields by merging the benefits of both local and global differential methods using robust tensor voting. However, tensor voting is a time consuming process due to its three constituent stages: stick, plate and ball tensor voting. Plate and ball tensor voting are the most expensive stages and responsible for dealing with image discontinuities. Both stages must be applied to all pixels, despite only a fraction of them usually belong to discontinuities.

In addition, Arredondo et al. (2004) proposed an approach to estimate optical flow fields using textural image information estimated using matrices designed to act as matched filters for certain types of quasiperiodic variations. Optical flow fields are then independently estimated in both intensity and textural images and then combined by weighting them according to the strength of the gradients in the neighborhood used to estimate the flow fields. Furthermore, Xu et al. (2012) presented an accurate optical flow estimation method that computes extensive initial flow vectors at each image level thus making the optimization process less dependent on the results from the coarser levels.

Moreover, Werlberger et al. (2009) proposed an anisotropic image-driven regularization based on the Huber norm. Image-driven diffusion filters regularize the flow field along image edges but not across them Weickert and Schnórr (2001).

Thus, they are prone to generating artifacts in textured image regions. In turn, Werlberger et al. (2010) applies non-local total variation regularization. Furthermore, their data term is based on patch-based normalized cross-correlation in order to gain robustness against illumination changes. However, their optimization operates on each individual pairwise term (data term and regularization term) yielding a high computational complexity, and their non-local term depends on color-similarities and on spatial distances between pixels. Unfortunately, the reliance on color similarities of the non-local term make the latter being influenced by textured, noisy pixels and illumination changes, leading to inaccurate flow fields and to blurred object boundaries.

Alternatively, Krähenbühl and Koltun (2012) introduces long-range temporal constraints in order to improve the scene flow consistency both visually and quantitatively. However, it depends on the image-driven diffusion tensor proposed in Werlberger et al. (2009), which yields artifacts with textured images. In turn, Hung et al. (2012) incorporated the traditional optimization model proposed in Werlberger et al. (2010) with an accelerated non-local regularization term that also depends on the colors and positions of pixels. In addition, Sun et al. (2010a) proposed an algorithm based on the classical optimization function introduced in Horn and Schunck (1981) with a weighted non-local term dependent on the color distance and spatial distance between pixels, which can be minimized as suggested in Li and Osher (2009) in order to denoise the resulting flow field while preserving object details. However, that non-local term dependent on the difference of intensity values is again influenced by textured, noisy pixels and illumination changes.

This chapter presents a robust algorithm for estimating accurate flow fields. The first contribution consists of replacing the discontinuity-preserving filtering stage based on tensor voting previously proposed in Chapter 3 by a similar stage exclusively based on stick tensor voting in order to reduce computational cost. The anisotropic stick tensor is used in the data term to make it robust against noise and outliers, as well as in the smoothness term in order to preserve the discontinuities of the estimated flow field.

Furthermore, the second contribution of this chapter aims at compensating for the loss of accuracy due to the suppression of both the plate and ball tensor voting. This is done by modifying the optimization function with an additional weighted non-local term that is similar to the one proposed in Sun et al. (2010a), although with its weights defined according to both saliency information obtained after the stick tensor voting process and the occlusion state of pixels, the latter as proposed in Sand and Teller (2008).

4.2 Complexity of tensor voting

The tensor voting process consists of stick, plate and ball tensor voting stages. Stick tensors are used in tensor voting to encode the orientation of the surface normal at every point. Tensor voting handles stick tensors through the so-called stick tensor voting, $SV(v, S_q)$. Stick tensor voting is based on the hypothesis that the normals of neighboring points lying on a same surface change smoothly (see Figure 4.1). Thus, the stick tensor voting proposed in Medioni et al. (2000) can be written as:

$$SV(v, S_q) = f_s[R_{2\theta} S_q R_{2\theta}^T], \tag{4.1}$$

where f_s is a decaying function, θ is the angle shown in Figure 4.1 and $R_{2\theta}$ a rotation with respect to the axis $v \times (S_q v)$. Function f_s was defined in (3.10) (see Tong et al., 2001):

$$f_s(v, S_q) = \begin{cases} e^{-\frac{l^2+bk^2}{\sigma^2}} & if\ \theta \leq \pi/4 \\ 0 & otherwise, \end{cases} \tag{4.2}$$

where σ is the standard deviation of a Gaussian function that modulates the influence of q over p based on their Euclidean distance, l is the length of the arc of circle between p and q, such that its endpoint tangents are orthogonal to $SV(v, S_q)$ and S_q, respectively (see Figure 4.1), k is the curvature of the arc and b is a function of σ as described in Tong et al. (2001). Under this assumption, the voter propagates its gradient to the votee if the angle θ between them is lower than or equal to 45^o (see Rashwan et al., 2012).

As shown in (4.2), the complexity of stick tensor voting mainly comes from the computation of an arcsine required to calculate l and the exponential required by (3.10). In addition, these computations are not necessary for $\theta \leq 45^o$

Furthermore, tensor voting utilizes plate tensors to encode edges. Ideally, if a point belongs to an edge, the third eigenvector of its associated tensor must be aligned with the tangent to the edge at that point, and the corresponding eigenvalue, λ_3, must be zero. Tensor voting handles plate tensors through the so-called plate tensor voting, $PV(v, P_q)$. The plate vote is defined as the aggregation of the stick votes cast by all the stick tensors $S_{p_q}(\phi)$ in which a specific plate P_q can be decomposed, Figure 4.2, where ϕ is a rotation angle with respect to an axis parallel to the third eigenvector of tensor P_q, and λ_1 is the biggest eigenvalue of P_q:

$$PV(v, P_q) = \frac{\lambda_1}{\pi} \int_0^{2\pi} SV(v, S_{P_q}(\phi)) d\phi. \tag{4.3}$$

In turn, ball tensors are utilized by tensor voting to encode either junctions or noise. Tensor voting handles ball tensors through the so-called ball tensor

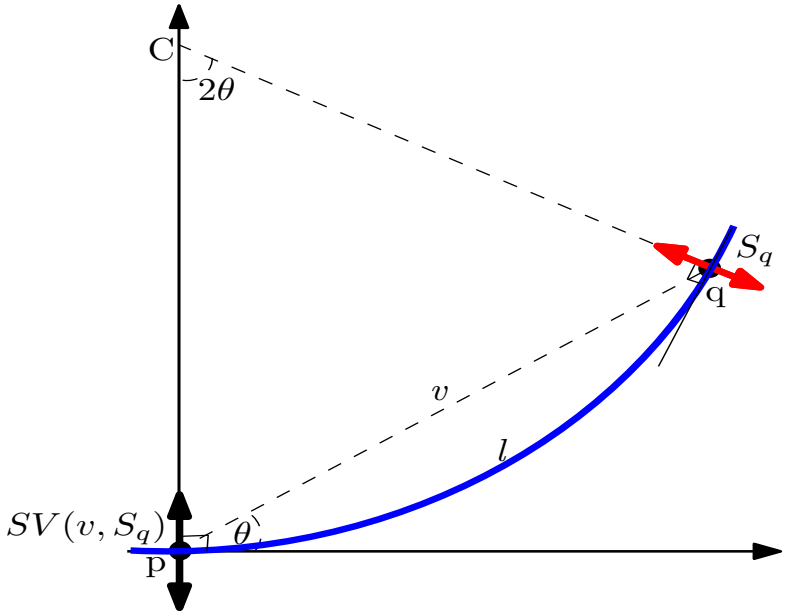


Figure 4.1: Stick tensor voting. A stick Sq casts a stick vote $SV(v, S_q)$ to p.

voting, $BV(v, B_q)$. Ball tensor voting is defined similarly to plate tensor voting as the integration of stick votes cast by all the stick tensors $S_{B_q}(\phi_1, \phi_2)$ in which a specific ball B_q can be decomposed:

$$BV(v, B_q) = \frac{3\lambda_1}{4\pi} \int_{\phi_1,\phi_2} SV(v, S_{B_q}(\phi_1, \phi_2)) d\phi_1 d\phi_2, \tag{4.4}$$

where $S_{B_q}(\phi_1, \phi_2)$ is a unitary stick tensor oriented in the direction $(1, \phi_1, \phi_2)$ in spherical coordinates.

As discussed above, plate and ball fields are respectively obtained by integrating stick spanning disks and spheres Tong et al. (2001) and Medioni et al. (2000). Thus, a significant computation time is necessary for the plate and ball voting stages, which are beneficial for preserving edges and junctions, despite those features usually correspond to a fraction of the image pixels. In Rashwan et al. (2011) and Rashwan et al. (2012), the rotation angle ϕ used for plate tensor voting (4.3) was discretized into 30^o steps. Therefore, plate tensor voting was obtained by integrating 12 rotated stick tensors for every neighbor of a point p. Likewise, the ball tensor voting was aggregated from 144 rotated stick tensors for every neighbor of a point p. For instance, if the window size used for voting is 9x9 pixels, the full tensor voting process requires 80 votes for stick tensor voting, 80x12 votes for plate tensor voting and 80x144 votes for ball tensor voting. As a result, the full tensor voting (stick, plate and ball) is a computationally intensive process. In

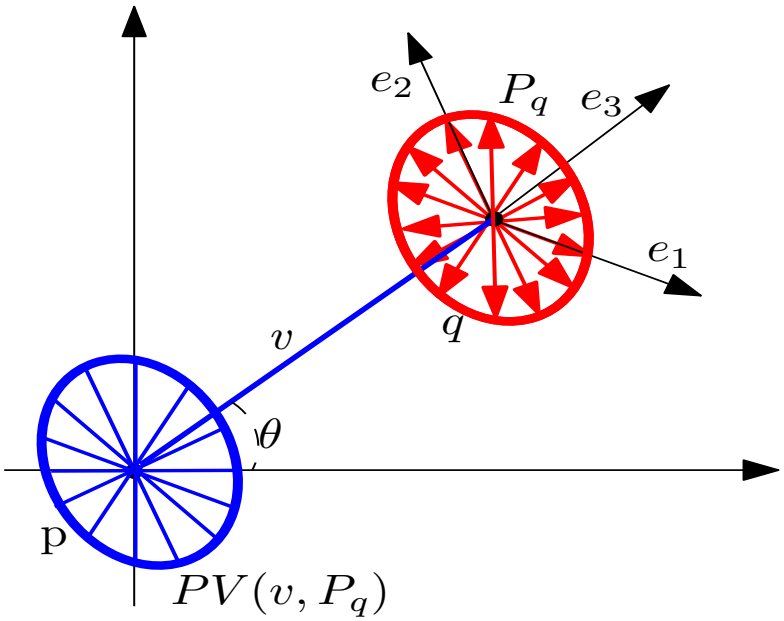


Figure 4.2: Plate tensor voting. Aggregation of stick votes cast to p by all the stick tensors belonging to the plate around a point q.

order to speed up this process, this chapter proposes the exclusive application of stick tensor voting in order to filter and smooth the input image gradients instead of applying full tensor voting as suggested in Rashwan et al. (2011, 2012).

For the aforementioned reasons, (3.1) is simplified in such a way that the result of applying stick tensor voting at pixel p is a tensor defined as:

$$STV(p) = \sum_{q \in \Theta(p)} SV(v, S_q). \tag{4.5}$$

$STV(p)$ can be decomposed into a stick tensor S_p, a plate tensor P_p and a ball tensor B_p, as illustrated in Figure 3.2 using (3.2). As defined in (3.2), eigenvector e_1 represents the estimated normal for points lying on a surface, while e_3 represents the most likely tangent direction of a curve for points belonging to that curve. Furthermore, three saliency measurements are defined: *surfaceness* ($S_1 = \lambda_1 - \lambda_2$) *edginess*, ($S_2 = \lambda_2 - \lambda_3$), and *junctionness* ($S_3 = \lambda_3$) (see Tong et al., 2001, Medioni et al., 2000).

4.3 Adapted variational optical flow model

After applying the pre-segmentation stage of the input spatio-temporal gradients illustrated in Chapter 3.4, the proposed method requires the computation of nine

image gradients for every pixel of a given color image. These gradients are filtered through separate stick tensor voting processes according to (4.5): the first spatio-temporal derivatives $\nabla_3 I^k = (I_x^k, I_y^k, I_t^k)^T$, the second spatio-temporal derivatives for the horizontal direction x, $\nabla_3 I_x^k = (I_{xx}^k, I_{xy}^k, I_{xt}^k)^T$, and the third spatio-temporal derivatives for the vertical direction y, $\nabla_3 I_y^k = (I_{yx}^k, I_{yy}^k, I_{yt}^k)^T$, with $k \in \{0, 1, 2\}$ being the color channel. For every pixel, the nine gradients $\nabla_3 I^k$, $\nabla_3 I_x^k$ and $\nabla_3 I_y^k$ are encoded as nine 3D stick tensors and used as initial input tensors for the stick voting process.

As a result, nine tensors for every pixel are obtained. Furthermore, after analyzing the resulting tensors, nine filtered gradient vectors are obtained for every pixel, which correspond to the eigenvectors associated with the biggest eigenvalues λ_1: $\hat{\nabla_3 I^k}$, $\hat{\nabla_3 I_x^k}$ and $\hat{\nabla_3 I_y^k}$, with $k \in \{0, 1, 2\}$. The nine resulting tensors are aggregated in a joint tensor ST as:

$$ST = \frac{1}{3}\sum_{k=0}^{2}(STV(\nabla_3 I^k) + STV(\nabla_3 I_x^k) + STV(\nabla_3 I_y^k)). \tag{4.6}$$

The surfaceness saliency measure, S_1, for every pixel p is computed as:

$$S_1(p) = \lambda_1(p) - \lambda_2(p), \tag{4.7}$$

where $\lambda_1(p)$ and $\lambda_2(p)$ are the first and second eigenvalues of the resulting tensor $ST(p)$ defined in (4.6), respectively.

The proposed definition of both the adapted data term and the adapted regularization term is described below.

4.3.1 Adapted data term

In this chapter, the motion tensor S in (3.25) is replaced by the result of applying stick tensor voting (4.5) to the neighborhood of p. Thus, a data term combining the brightness and the gradient constancy assumptions using stick tensor voting can be defined as a direct adaptation of (3.23):

$$\begin{aligned} M(w, I) = w^T[&STV(\nabla_3 I) + \gamma(STV(\nabla_3 I_x) \\ &+ STV(\nabla_3 I_y))]w, \end{aligned} \tag{4.8}$$

The symmetric tensor finally obtained as a result of the stick voting processes for $\nabla_3 I$, $\nabla_3 I_x$ and $\nabla_3 I_y$ is the direct adaptation of (3.25):

$$S = STV(\nabla_3 I) + \gamma[STV(\nabla_3 I_x) + STV(\nabla_3 I_y)]. \tag{4.9}$$

The final data term that is applicable to HSV color images is the corresponding adaptation of (3.27):

$$M(w, I) = \sum_{k=0}^{2} \Psi_M(w^S S^k w), \tag{4.10}$$

where S^k is (4.9) applied to the $k-th$ color channel of I.

4.3.2 Adapted regularization term

The complementary regularizer introduced in Zimmer et al. (2009, 2011), and later adapted in Chapter 3.6.2 (see Rashwan et al., 2011, 2012), has been further adapted to stick tensor voting in this chapter. In particular, the regularization term R defined in (3.29) has been reformulated as:

$$R = \sum_{k=0}^{2} [STV(\nabla_2 I^k) + \gamma(STV(\nabla_2 I_x^k) + STV(\nabla_2 I_y^k))]. \tag{4.11}$$

The first two eigenvectors (e_1, e_2) of tensor R are then used to compute the regularization term by applying (3.30) as illustrated in Chapter 3.6.2.

4.4 Improved optical flow model

The only reliance on stick tensor voting in both the data and the smoothness terms causes the loss of some of the benefits of full tensor voting for properly preserving edges and object boundaries and details in the estimated flow fields. In order to diminish the negative impact on flow discontinuities of the exclusive use of stick tensor voting, it is necessary to introduce a non-local term (NL) (a practical median filter) similar to the one proposed in Sun et al. (2010a) to denoise the flow field.

However, applying a median filter in a large neighborhood has negative effects on edges and corners, since they are affected by their surroundings, leading to oversmoothing. Thus, it is necessary to ensure that pixels only propagate their information within their same regions, provided they do not belong to edges and corners. In practice, the non-local term proposed in Sun et al. (2010a) has been modified by introducing a weighting function H, which gives high values for pixels belonging to the same surface (region) and low values for pixels corresponding to edges, corners and thin structures.

A weighting function H was proposed in Sun et al. (2010a) for all image pixels based on their spatial distance and their difference of intensities:

$$H(x,y,\acute{x},\acute{y}) = \exp(\frac{\mid x-\acute{x} \mid^2 + \mid y-\acute{y} \mid^2}{2\sigma_p^2} - \frac{\mid I(x,y) - I(\acute{x},\acute{y}) \mid^2}{2\sigma_I^2}), \tag{4.12}$$

where $(\acute{x},\acute{y})$ is the spatial position of any pixel $\acute{p}$ belonging to a neighborhood of pixel $p=(x,y)$ in a possibly large region $N_{x,y}$, $I(x,y)$ and $I(\acute{x},\acute{y})$ are the intensity values of $(\acute{x},\acute{y})$ and (x,y), respectively, and σ_p and σ_I are standard deviations.

As shown in (4.12), the weighting function H partially depends on the difference of intensity values. As a result, this function can be influenced by textured pixels, noisy pixels, and shadows and illumination changes, as shown in Figure 4.3. Thus, it negatively affects object motion details in the estimated flow fields. In this chapter, the aforementioned weighting function is redefined as the complement of the normalized saliency of *surfaceness*, $1-S_1(p)$, which is obtained from the stick tensor voting stage for every image gradient, as shown in Figure 4.3 and described in Section 4.3 (4.7).

Most current optical flow estimation approaches do not handle occlusions, thus yielding artifacts particularly near moving occlusion boundaries. In this chapter, those occlusion effects have been addressed by using the flow divergence and by considering temporal changes between consecutive frames, as suggested in Sand and Teller (2008). This is beneficial for extracting a set of candidate occluding and disoccluding points, which convey information about boundaries that respectively appear and disappear.

In particular, the flow divergence is defined as:

$$Div(p) = \frac{\partial}{\partial_x}u + \frac{\partial}{\partial_y}v. \tag{4.13}$$

The occlusion state of pixels, $O(p)$, is estimated by combining the flow divergence and the pixel projection difference, as proposed in Sand and Teller (2008), in order to identify occluded pixels. Based on the latter, the occluding boundary function, $d(p)$, is defined in Sand and Teller (2008) as:

$$d(p) = \begin{cases} Div(p) & Div(p) \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.14}$$

In turn, the pixel projection difference, $e(p)$, is defined as:

$$e(p) = I(x,y,t) - I(x+u, y+v, t+dt) \tag{4.15}$$

Finally, the pixel's occlusion state can be expressed as a combination of both $d(p)$ and $e(p)$ by using zero-mean, non-normalized Gaussian functions Sand and Teller (2008):

$$O(p) = NG(e(p),\sigma_e)NG(d(p),\sigma_d), \tag{4.16}$$

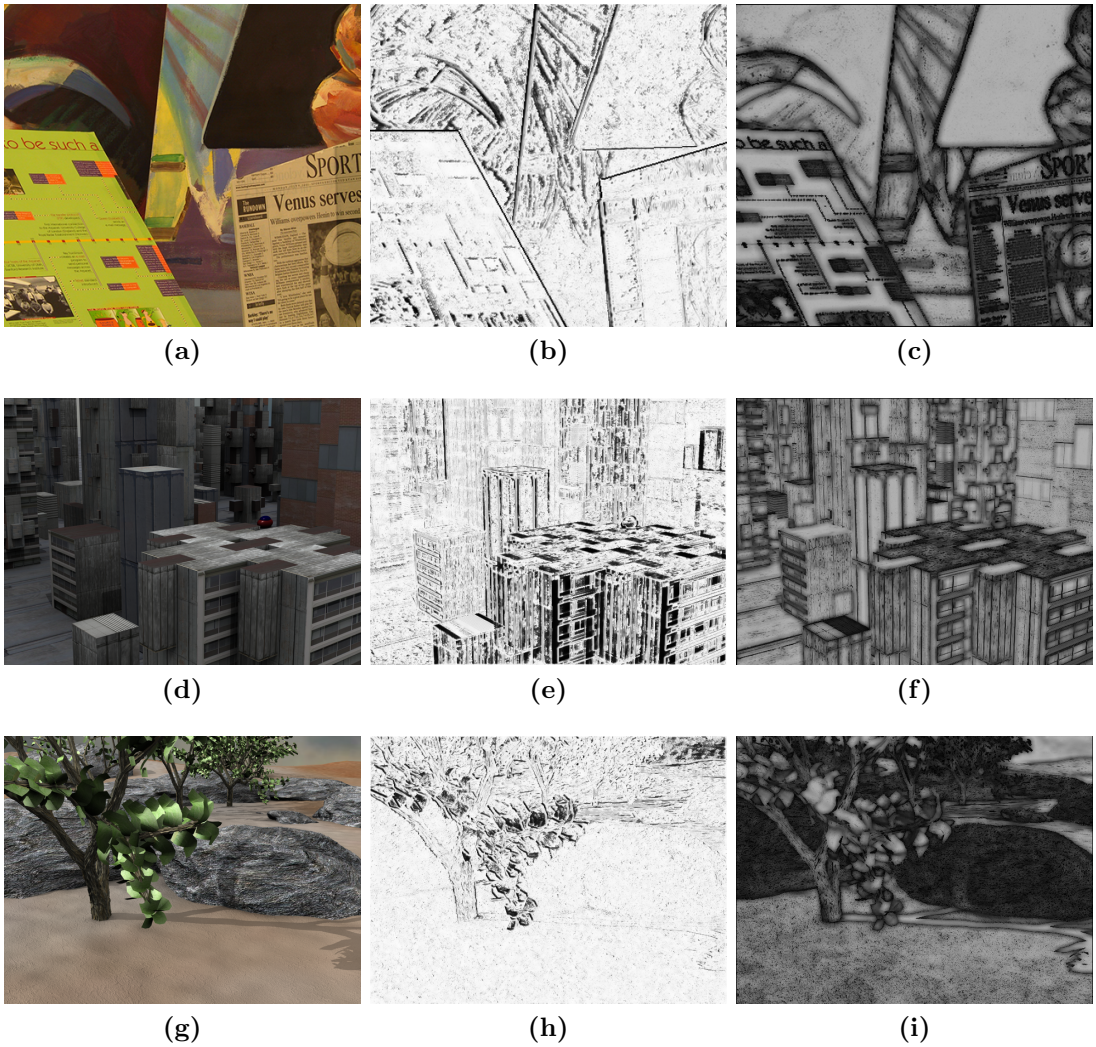


Figure 4.3: (a,d,g) Original frames of Middlebury sequences Venus, Urban2 and Grove2. (b,e,h) Complement of the normalized saliency of *surfaceness*, $1 - S_1$. (c,f,i) Weighting function H suggested in Sun et al. (2010a).

where σ_e and σ_d are standard deviations experimentally set to 0.5 and 10, respectively, and $O(p)$ is close to zero for occluded pixels and close to one for non-occluded pixels (Figure 4.4).

Thus, a weighting function $\varpi_{p,\acute{p}}$ is introduced in the proposed non-local term to take into account the occlusion state of pixels, $O(p)$, and the complement of the

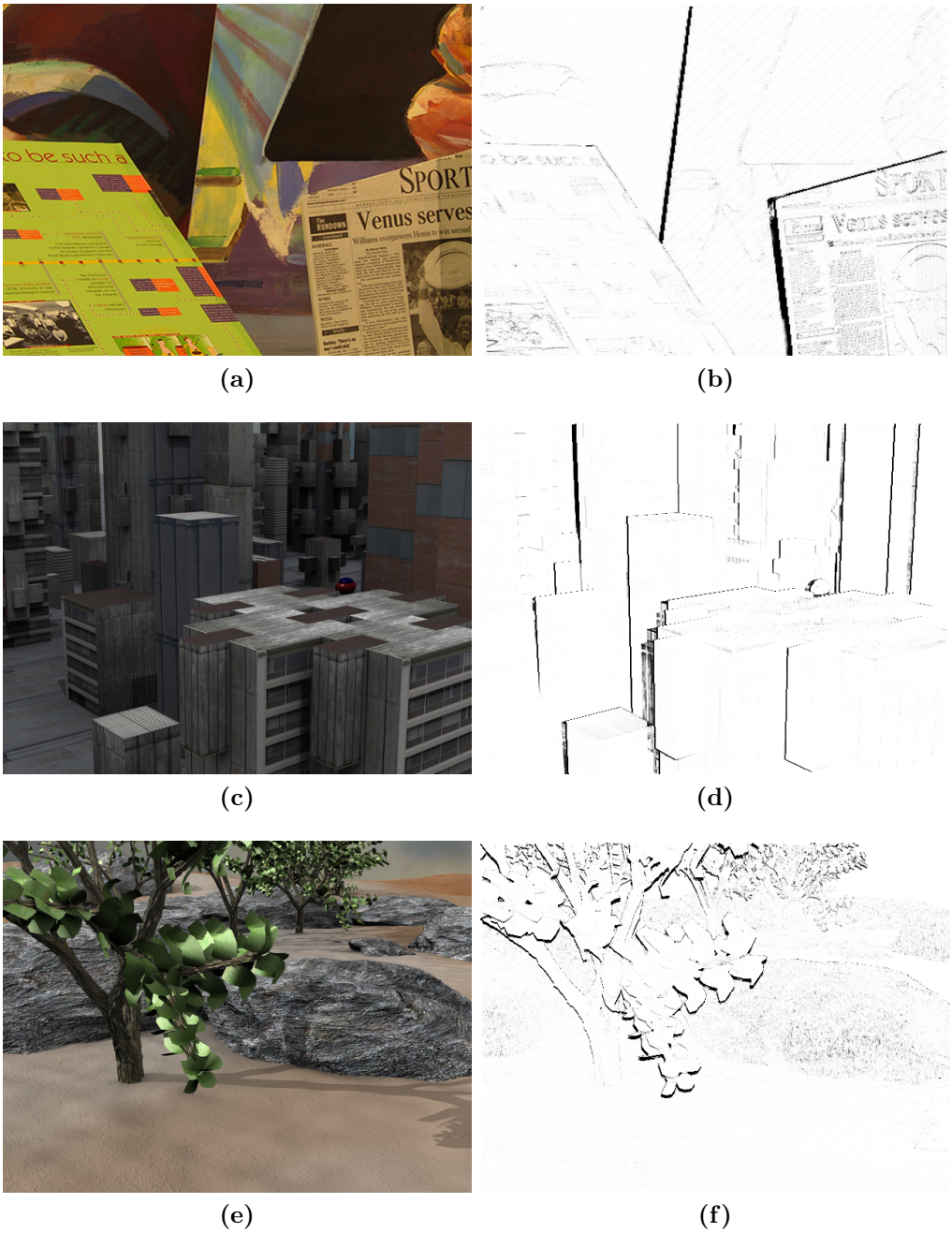


Figure 4.4: (a,c,e) Original frames of Middlebury sequences Venus, Urban2 and Grove2. (b,d,f) Resulting occlusion state $O(p)$ as suggested in Sand and Teller (2008).

normalized saliency of *surfaceness*, $1 - S_1(p)$:

$$\varpi_{p,\acute{p}} = (1 - S_1(p))\frac{O(\acute{p})}{O(p)}. \tag{4.17}$$

Thus, the functional in (3.11) is complemented with a weighted non-local term, which is a particular median filter within a region of an auxiliary flow field $(\hat{u}, \hat{v})$ defined below. The optimization process is thus redefined as:

$$\begin{aligned}\min_{u,v,\hat{u},\hat{v}} \quad E_I(u,v,\hat{u},\hat{v}) = &\sum_{x,y\in\Omega} M(u,v,I) + \alpha V(\nabla_2 u, \nabla_2 v, I) \\ &+ \lambda(\| u - \hat{u} \|^2 + \| v - \hat{v} \|^2) \\ &+ \sum_{(x,y)} \sum_{(\acute{x},\acute{y})\in N_{x,y}} \varpi_{p,\acute{p}}(|\hat{u}_{x,y} - \hat{u}_{\acute{x},\acute{y}}| + \\ &|\hat{v}_{x,y} - \hat{v}_{\acute{x},\acute{y}}|),\end{aligned} \tag{4.18}$$

λ is the weight of the coupling term that, in practice, is small or steadily increased from small values (*i.e.*, changed logarithmically from 10^{-2} to 1) for each step in the alternating optimizations of (4.18) (see Sun et al. (2010a)). In turn, $\hat{u}$ and $\hat{v}$ are the resulting flow fields obtained by minimizing the weighted non-local term as suggested in Sun et al. (2010a). Moreover, $\hat{u}_p$ and $\hat{v}_p$ are the refined horizontal and vertical components of the flow vector at point $p = (x, y)$, which are formulated as described in Li and Osher (2009):

$$\begin{aligned}\hat{u}_p^{(h+1)} &= median\{Neighbours_u^{(h)} \cup Data_u\} \\ \hat{v}_p^{(h+1)} &= median\{Neighbours_v^{(h)} \cup Data_v\},\end{aligned} \tag{4.19}$$

where h is the number of steps of the alternating optimization, $Neighbours_u^{(h)} = \{\hat{u}_{\acute{p}}^{(h)}\}$ and $Neighbours_v^{(h)} = \{\hat{v}_{\acute{p}}^{(h)}\}$ for $\acute{p} \in N_{x,y}$, with $\hat{u}^{(0)} = u$ and $\hat{v}^{(0)} = v$. $Data_u$ is the set of weighted values of u_p within $N_{x,y}$:

$$\begin{aligned}Data_u = \{&u_p, u_p \pm \frac{\varpi_{p,\acute{p}}}{\lambda}, u_p \pm \frac{2\varpi_{p,\acute{p}}}{\lambda}, ... \\ &u_p \pm \frac{|N_{x,y}|\varpi_{p,\acute{p}}}{2\lambda}\},\end{aligned} \tag{4.20}$$

where $|N_{x,y}|$ is the number of neighbours of a certain point p. Similarly, $Data_v$ is defined as:

$$\begin{aligned}Data_v = \{&v_p, v_p \pm \frac{\varpi_{p,\acute{p}}}{\lambda}, v_p \pm \frac{2\varpi_{p,\acute{p}}}{\lambda}, ... \\ &v_p \pm \frac{|N_{x,y}|\varpi_{p,\acute{p}}}{2\lambda}\},\end{aligned} \tag{4.21}$$

In order to minimize (4.18), an alternating optimization process is performed by defining two functionals, E_{I1} and E_{I2}, as proposed in Sun et al. (2010a):

$$E_{I1}(u,v,\hat{u},\hat{v}) = \sum_{x,y\in\Omega} M(u,v,I) + \alpha V(\nabla_2 u, \nabla_2 v, I) + \lambda(\| u-\hat{u} \|^2 + \| v-\hat{v} \|^2). \tag{4.22}$$

$$E_{I2}(u,v,\hat{u},\hat{v}) = \lambda(\| u-\hat{u} \|^2 + \| v-\hat{v} \|^2) + \sum_{x,y} \sum_{(\acute{x},\acute{y})\in N_{x,y}} \varpi_{p,\acute{p}}(|\hat{u}_{x,y} - \hat{u}_{\acute{x},\acute{y}}| + |\hat{v}_{x,y} - \hat{v}_{\acute{x},\acute{y}}|). \tag{4.23}$$

A multi-scale, coarse-to-fine scheme is used by most modern algorithms for optical flow estimation in order to support both small and large motion and to improve the accuracy of flow fields. This approach relies on estimating the optical flow in a Gaussian pyramid, where the bottom image is the original image at the finest scale, and the levels above are warped representations of the images based on the flow estimated at every preceding scale Brox et al. (2004), Bruhn et al. (2005).

At each pyramid level, the alternating optimization process first holds $\hat{u}$ and $\hat{v}$ constant and minimizes the linear system corresponding to the Euler-Lagrange equations of (4.22) with respect to u and v (initially set to zero) by using a SOR type solver with alternating line relaxation Press et al. (1993). Subsequently, by fixing u and v, (4.23) is minimized with respect to $\hat{u}$ and $\hat{v}$ (initially set to u and v) based on the median formulation proposed in Li and Osher (2009) as shown in (4.19). The alternating optimizations are repeated h steps at every pyramid level to denoise the resulting flow fields. The weighting parameter λ of the coupling term is changed logarithmically (in this work from 10^{-2} to 1), as proposed in Sun et al. (2010a). In the end, the resulting $\hat{u}$ and $\hat{v}$ are the horizontal and vertical components of the sought optical flow field.

4.5 Experimental results

In order to evaluate the performance of the proposed variational optical flow method, experiments on the widely used Middlebury optical flow data sets have been performed. The parameters of the proposed method have been experimentally set to: the standard deviation $\sigma_1 = 1.50$ for filtering the homogeneous regions based on stick tensor voting, the standard deviation $\sigma_2 = 0.75$ for textured regions, $\alpha = 15$ and $\gamma = 80$. Moreover, the SNR threshold τ has been set to 25 Rashwan

et al. (2011, 2012). Regarding the coarse-to-fine scheme, the rescaling factor has been set to 0.90.

According to the Middlebury benchmark[2], the proposed technique, referred to as IROF++, is in the $1th$ position out of 64 methods with respect to the Average End-Point Error (AEE), in the $3th$ place regarding the Average Angular Error (AAE), in the $6th$ position with respect to the Average Interpolation Error and in the $4th$ position with respect to the Average Normalized Interpolation Error, Figure 4.5.

In order to separately assess the different contributions of this chapter, the resulting flow fields for the following algorithms have been computed: (a) the baseline method proposed in Rashwan et al. (2012), which uses stick, plate and ball tensor voting plus the discontinuity -preserving stage (TV+DS), (b) the proposed method with stick tensor voting alone (ST), (c) the proposed method with stick tensor voting and the discontinuity-preserving stage proposed in Rashwan et al. (2011, 2012) (ST+DS), (d) the proposed method with stick tensor voting, the discontinuity-preserving stage, and the weighted non-local term proposed in Sun et al. (2010a) (ST+DS+SW), (e) the proposed method with stick tensor voting plus the proposed variation of the aforementioned weighted non-local term by using *surfaceness* saliency as defined in (4.17) (ST+NW), and (f) the proposed method with stick tensor voting, the discontinuity-preserving stage, and the proposed variation of the aforementioned weighted non-local term by using *surfaceness* saliency (ST+DS+NW).

The proposed techniques have been tested upon 12 datasets from the Middlebury database, all of them with corresponding ground-truths. The baseline method Rashwan et al. (2012) and the four aforementioned variations of the proposed technique have been tested by calculating the average end-point error AEE (table 4.1) and the average angular error AAE (table 4.2). The proposed method with stick tensor voting and the weighted non-local term based on the *surfaceness* saliency yields the lowest error among the five variations.

Qualitative results of some of these experiments are shown in Figure 4.7. The flow fields obtained by adapting the data and regularization terms with stick tensor voting present artifacts and deformations near discontinuities and occluded boundaries. Figure 4.6(row 1) shows two examples of blurring edges in the *Dimetrodon* sequence from the Middlebury datasets. Thus, using a weighted non-local term is useful to avoid smoothing near discontinuities and to preserve edges and object boundaries, as well as to prevent smoothing near occluded boundaries, as shown in Figure 4.7 (row 5 and 6) for the proposed method with the weighted non-local term proposed in Sun et al. (2010a) and its adaptation according to (4.17), respec-

[2]The present IROF++ ranking is related to the submission date (Feb. 2012). Results can be seen at: http://vision.middlebury.edu/flow/eval/

Average endpoint error	avg.	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
IROF++ [63]	7.3	0.08 1	0.23 1	0.07 1	0.21 4	0.68 7	0.17 4	0.28 1	0.63 1	0.19 1	0.15 1	0.73 1	0.09 1	0.60 1	0.89 1	0.42 1	0.43 1	1.08 1	0.31 1	0.10 1	0.12 1	0.12 1	0.47 1	0.98 1	0.68 1
MDP-Flow2 [40]	7.6	0.09 11	0.23 4	0.07 8	0.16 2	**0.52** 1	0.13 3	0.22 3	0.46 3	0.17 6	0.17 17	0.93 25	0.09 9	0.65 6	0.98 7	0.43 6	**0.29** 1	0.91 3	0.26 3	0.11 8	0.13 12	0.17 10	0.51 12	1.11 14	0.72 11

(a)

Average angle error	avg.	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
nLayers [61]	5.5	**2.80** 1	**7.42** 1	**2.20** 1	2.71 12	7.24 2	2.55 25	2.61 2	6.24 2	2.45 23	2.30 3	12.7 4	1.16 3	**2.30** 1	**3.02** 1	**1.70** 1	2.62 2	**6.95** 1	2.09 2	2.29 14	3.46 6	1.89 14	1.38 2	3.06 3	1.29 2
Layers++ [38]	8.6	3.11 3	8.22 3	2.79 12	2.43 6	**7.02** 1	2.24 13	**2.43** 1	**5.77** 1	2.18 14	**2.13** 1	**9.71** 1	1.15 2	2.35 2	**3.02** 1	1.96 2	3.81 17	11.4 14	3.22 20	2.74 27	4.01 29	2.35 23	1.45 3	3.05 2	1.79 8
IROF++ [63]	9.0	3.17 4	8.69 4	2.61 2	2.79 9	9.61 11	2.33 11	3.43 4	8.86 11	2.38 16	2.87 4	14.8 11	1.52 7	2.74 4	3.57 4	2.19 3	3.20 3	9.70 3	2.71 1	1.96 1	3.45 4	1.22 1	1.80 4	4.06 7	2.50 14
MDP-Flow2 [40]	9.3	3.32 13	8.76 8	2.85 14	**2.18** 1	7.47 4	1.85 5	2.77 4	6.95 4	2.06 11	3.25 20	17.3 24	1.59 18	2.87 8	3.73 7	2.32 7	3.15 4	11.1 13	2.65 4	2.04 8	3.64 13	1.60 8	1.88 10	4.49 11	1.49 3

(b)

Average interpolation error	avg.	Mequon (Hidden texture) im0 GT im1			Schefflera (Hidden texture) im0 GT im1			Urban (Synthetic) im0 GT im1			Teddy (Stereo) im0 GT im1			Backyard (High-speed camera) im0 GT im1			Basketball (High-speed camera) im0 GT im1			Dumptruck (High-speed camera) im0 GT im1			Evergreen (High-speed camera) im0 GT im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [40]	8.0	2.86 2	5.31 3	1.20 2	3.46 3	5.07 3	1.31 4	**3.49** 1	**5.34** 1	1.47 3	5.40 12	7.95 18	3.41 38	10.2 3	12.7 3	3.61 15	6.12 8	11.8 7	2.38 21	7.48 3	17.1 3	1.51 2	**7.32** 1	**11.4** 1	1.75 34
CBF [12]	11.4	**2.83** 1	5.20 2	1.23 22	3.97 23	5.79 21	1.56 26	3.62 3	5.47 2	1.60 9	5.21 2	**7.12** 1	3.29 21	10.1 2	12.6 2	3.62 15	5.97 4	11.5 4	2.31 9	7.76 15	17.8 16	1.61 15	7.60 6	11.9 6	1.76 41
Aniso. Huber-L1 [22]	13.7	2.95 8	5.44 7	1.24 28	4.42 40	6.27 40	1.67 38	3.79 11	5.70 7	1.50 4	5.31 4	7.42 8	3.24 16	11.1 21	14.0 28	3.61 15	5.91 3	11.4 3	**2.24** 1	7.60 8	17.3 5	1.51 2	7.62 8	11.9 6	1.73 21
CLG-TV [51]	14.0	2.94 7	5.45 9	1.25 33	4.26 38	6.17 32	1.60 31	3.68 7	5.73 9	1.73 13	5.36 7	7.41 5	3.32 31	11.1 21	14.0 28	3.57 6	5.88 2	11.3 2	2.26 2	7.58 6	17.0 2	1.57 12	7.75 12	12.1 13	1.72 16
IROF-TV [56]	14.5	3.07 18	5.91 23	1.23 22	3.71 13	5.47 1	1.40 13	3.70 8	6.27 23	1.58 8	5.25 3	7.60 6	3.17 7	11.0 17	13.9 19	4.47 53	6.37 21	12.4 23	2.30 7	7.79 18	17.9 18	**1.50** 1	7.63 9	11.9 6	**1.66** 1
LCM-flow [65]	14.8	2.86 2	**5.13** 1	1.25 33	3.94 22	5.87 23	1.64 34	3.87 15	6.60 31	1.79 15	5.37 9	7.29 4	3.30 25	**9.99** 1	**12.5** 1	3.56 3	6.12 8	11.8 7	2.26 2	7.76 16	17.7 14	1.68 30	7.58 5	11.8 4	1.80 52
IROF++ [63]	15.2	3.03 16	5.77 8	1.20 1	3.59 7	5.31 2	1.33 1	4.32 28	6.61 32	2.25 28	5.06 1	7.14 1	3.16 1	11.0 17	13.9 19	4.44 51	6.34 19	12.3 19	2.27 4	7.54 1	17.3 1	1.64 28	8.09 28	12.7 8	1.69 1
Second-order prior [8]	16.9	2.91 6	5.39 8	1.24 28	4.26 38	6.21 35	1.56 26	3.82 13	6.34 28	1.62 10	5.39 11	7.68 11	3.04 2	11.1 21	13.9 19	3.59 8	6.14 10	11.9 10	2.31 9	7.61 9	17.4 10	1.63 23	7.90 17	12.4 15	1.78 43

(c)

Average normalized interpolation error	avg.	Mequon (Hidden texture) im0 GT im1			Schefflera (Hidden texture) im0 GT im1			Urban (Synthetic) im0 GT im1			Teddy (Stereo) im0 GT im1			Backyard (High-speed camera) im0 GT im1			Basketball (High-speed camera) im0 GT im1			Dumptruck (High-speed camera) im0 GT im1			Evergreen (High-speed camera) im0 GT im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [40]	8.7	**0.58** 1	**0.71** 1	**0.64** 1	0.63 4	0.87 4	**0.59** 1	0.92 5	1.37 8	0.85 7	0.98 15	1.14 32	1.24 13	**0.98** 1	**0.95** 1	1.15 4	1.13 20	1.60 24	1.08 17	0.68 6	1.23 6	0.68 9	**0.75** 1	**1.06** 1	0.64 20
CLG-TV [51]	15.8	0.63 26	0.86 27	0.66 28	0.81 37	1.12 36	0.66 29	0.96 8	1.43 10	0.96 17	0.97 11	1.03 6	1.25 20	1.06 2	1.08 22	1.15 4	**1.02** 1	**1.25** 1	1.04 2	**0.63** 1	**1.09** 1	0.66 3	0.97 33	1.45 33	0.63 2
LCM-flow [65]	16.2	0.62 18	0.80 19	0.66 28	0.77 27	1.07 27	0.71 42	1.03 14	1.70 28	0.91 14	1.01 25	1.07 12	1.27 28	0.99 2	**0.95** 1	1.16 17	1.07 7	1.43 8	1.04 3	0.67 5	1.20 5	0.71 19	0.84 9	1.21 9	0.65 36
Aniso. Huber-L1 [22]	16.9	0.62 18	0.80 19	0.66 28	0.84 40	1.13 38	0.66 29	1.03 14	1.44 11	0.93 16	0.97 11	1.03 6	1.26 25	1.06 21	1.09 23	1.15 4	1.08 9	1.46 11	**1.03** 1	0.64 2	1.12 2	0.66 3	0.99 37	1.48 40	0.63 2
IROF++ [63]	17.1	0.59 2	0.74 2	**0.64** 1	0.65 8	0.89 7	**0.59** 1	1.15 25	1.71 27	1.17 25	**0.92** 1	**0.96** 1	1.21 4	1.17 42	1.26 42	1.69 53	1.11 13	1.54 13	1.04 2	0.68 6	1.23 6	0.70 13	1.07 57	1.62 58	0.63 2
IROF-TV [56]	17.6	0.62 16	0.84 25	0.65 16	0.67 9	0.92 9	0.60 3	0.92 5	1.49 17	0.79 4	0.94 7	1.02 8	1.22 1	1.18 51	1.28 48	1.70 61	1.12 11	1.58 14	1.05 7	0.79 11	1.67 14	0.70 13	0.85 11	1.24 11	0.64 20
p-harmonic [29]	18.2	0.61 8	0.83 21	**0.64** 1	0.82 38	1.14 41	0.68 35	0.91 3	1.49 17	0.77 3	1.04 32	1.11 21	1.28 33	1.05 17	1.07 16	1.15 4	1.06 5	1.39 5	1.07 14	0.70 11	1.31 12	0.76 33	0.96 36	1.44 32	0.63 2

(d)

Figure 4.5: Results of Middlebury benchmark of Feb. 2012. The proposed method (IROF++) is highlighted. (a) Topmost methods according to the Average End-Point Error (AEE). (b) Topmost methods according to the Average Normalized Interpolation Error (AAE). (c) Topmost methods according to the Average Normalized Interpolation Error (AIE). (d) Topmost methods according to the Average Normalized Interpolation Error (ANIE).

tively. Moreover, the resulting flow fields with stick tensor voting have strongly been affected by shadow regions, as shown in the example of the *RubberWhale* sequence, Figure 4.6(row 2).

In another experiment, the results of the proposed method have been visually compared to those of: (i) the baseline method TV proposed in Rashwan et al. (2011, 2012), (ii) the proposed method with ST+DS, (iii) the proposed method with ST+DS+SW, and (iv) the proposed method with ST+DS+NW. For Rash-

Methods	Dim	Grov2	Grov3	Hyd	Rub	Urb2	Urb3	Venus
(a)	0.124	0.151	0.458	0.152	0.115	1.503	0.143	0.237
(b)	0.326	0.691	0.634	0.754	0.373	1.738	0.572	1.076
(c)	0.238	0.580	0.549	0.415	0.262	1.325	0.462	0.874
(d)	0.109	0.103	0.289	0.112	0.081	1.917	0.127	0.231
(e)	0.121	0.116	0.302	0.126	0.092	1.118	0.139	0.223
(f)	**0.094**	**0.098**	**0.224**	**0.092**	**0.073**	**1.027**	**0.087**	**0.209**

Table 4.1: AEE for the eight tested sequences from the Middlebury dataset.

Methods	Dim	Grov2	Grov3	Hyd	Rub	Urb2	Urb3	Venus
(a)	3.005	2.152	4.722	1.832	4.207	5.048	2.874	3.249
(b)	5.382	3.907	6.124	3.547	5.004	5.348	4.149	5.897
(c)	5.012	3.326	5.415	3.124	4.939	4.471	3.854	5.071
(d)	2.861	1.441	3.412	**1.287**	2.581	7.045	2.623	**3.119**
(f)	3.097	1.603	3.346	1.421	2.641	7.235	2.698	3.382
(e)	**2.525**	**1.404**	**3.340**	1.302	**2.363**	**3.859**	**2.507**	3.294

Table 4.2: AAE for the eight tested sequences from the Middlebury dataset.

wan et al. (2011, 2012), the parameters suggested in those references were used. Figure 4.8(column 1-2) shows a visual comparison for two regions of the *Army* sequence. Both the method proposed in Rashwan et al. (2011, 2012) and the proposed technique with the weighted non-local term suggested in Sun et al. (2010a) estimate good flow fields with adequate preservation of discontinuities. However, the flow fields estimated with the proposed technique (iv) contain more motion details than the baseline method (i) and the two different variations (ii) and (iii). In addition, it preserves flow discontinuities and the contours of small objects significantly better than the three aforementioned variations. For instance, the fist crop of Army sequence flow field with the proposed technique (iv) shown in Figure 4.8(row 1) is able to show the smallest details of the soldiers and the rifle contours. Moreover, Figure 4.8(column 3 and 4) visually compares the flow fields estimated for two regions within the *Grove*2 sequence with the aforementioned optical flow methods. The proposed technique (iv) is able to clearly identify object boundaries (*e.g.*, the tree branches) better than the other approaches (i), (ii) and (iii).

Additionally, in order to asses the weight of each component in the final proposed algorithm, the effect of the different variations of the proposed technique with respect to the AEE with respect to the baseline method (TV+DS) has been measured. As shown in table 4.3, the resulting flow fields significantly deteriorate

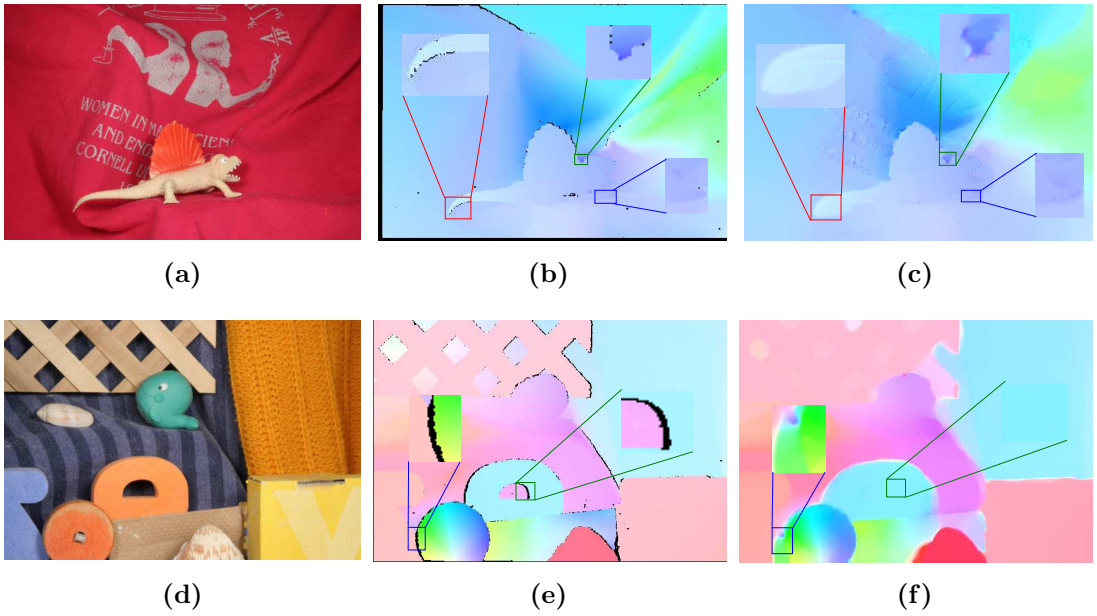


Figure 4.6: (Column 1) Original frames of Middlebury sequences Dimetrodon and Grove2. (Column 2) Corresponding ground-truths. (Column 3) Corresponding flow fields with the proposed method ST+DS.

when only stick tensor voting is applied. In addition, adding the discontinuity preserving stage yields a slight improvement of the AEE for the resulting flow fields. In turn, adding the non-local term proposed in Sun et al. (2010a) leads to a significant improvement of the AEE. Furthermore, the weighted non-local term based on *surfaceness* saliency yields a major improvement of the final flow fields.

Methods	Dimetrodon %	Grove2 %	Hydrangea %	RubberWhale %
ST (b)	-160	-358	-396	-224
ST+DS (c)	-92	-285	-173	-127
ST+DS+SW (d)	+13	+31	+26	+29
ST+NW (e)	+03	+23	+17	+20
ST+DS+NW (f)	+25	+35	+40	+37

Table 4.3: Effect of the five tested variations of the proposed technique on the Average End-Point Error with respect to the baseline method.

The present work aims at reducing the computational time associated with full tensor voting as proposed in Rashwan et al. (2012), while keeping the accuracy of

the resulting flow fields. Thus, the computation times for the Yosemite (252x316 gray images) and Urban (640x480 color images) sequences have been obtained and shown in table 4.4 for the four different variations proposed in this chapter and the baseline method proposed in Rashwan et al. (2012). All methods have been run on an Intel Dual Core at 3.2 GHz executing Matlab code. As shown in table 4.4, there is a significant reduction in the execution time of the baseline method proposed in Rashwan et al. (2012).

Methods	**Yosemite** (seconds)	**Urban** (seconds)
TV+DS (a)	123	270
ST (b)	51	143
ST+DS (c)	53	145
ST+DS+SW (d)	78	187
ST+DS+NW (e)	67	165

Table 4.4: Computation times for the Yosemite and Urban sequences corresponding to the five tested variations of the proposed technique.

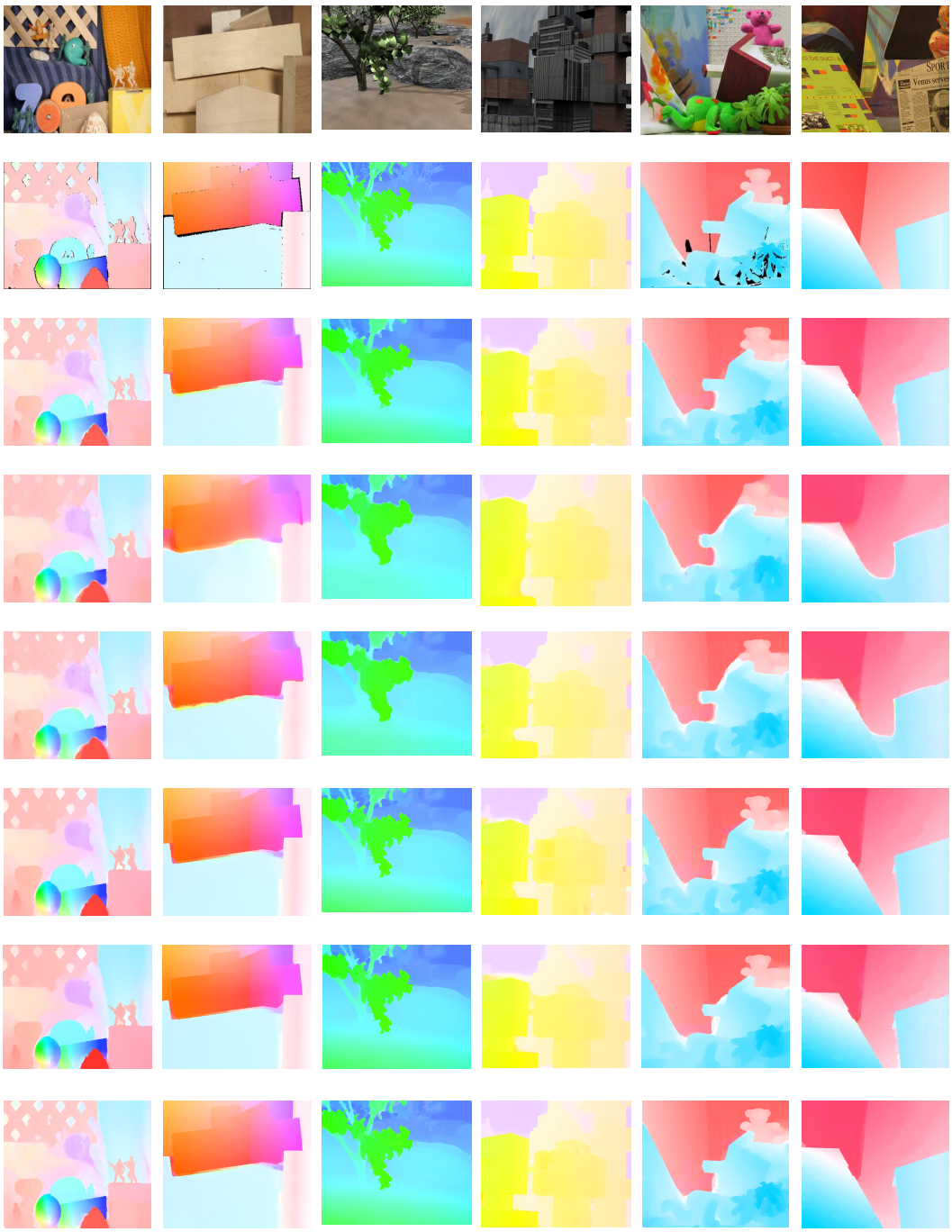


Figure 4.7: (row 1) Original frames of Middlebury sequences "Army", "Wooden" and "Grove2". (row 2) Corresponding ground-truths. (row 3) Corresponding flow fields with the baseline method TV+DS. (row 4) Corresponding flow fields with ST. (row 5) Corresponding flow fields with ST+DS. (row 6) Corresponding flow fields with ST+DS+SW. (row 7) Corresponding flow fields with ST+NW. (row 8) Corresponding flow fields with ST+DS+NW.

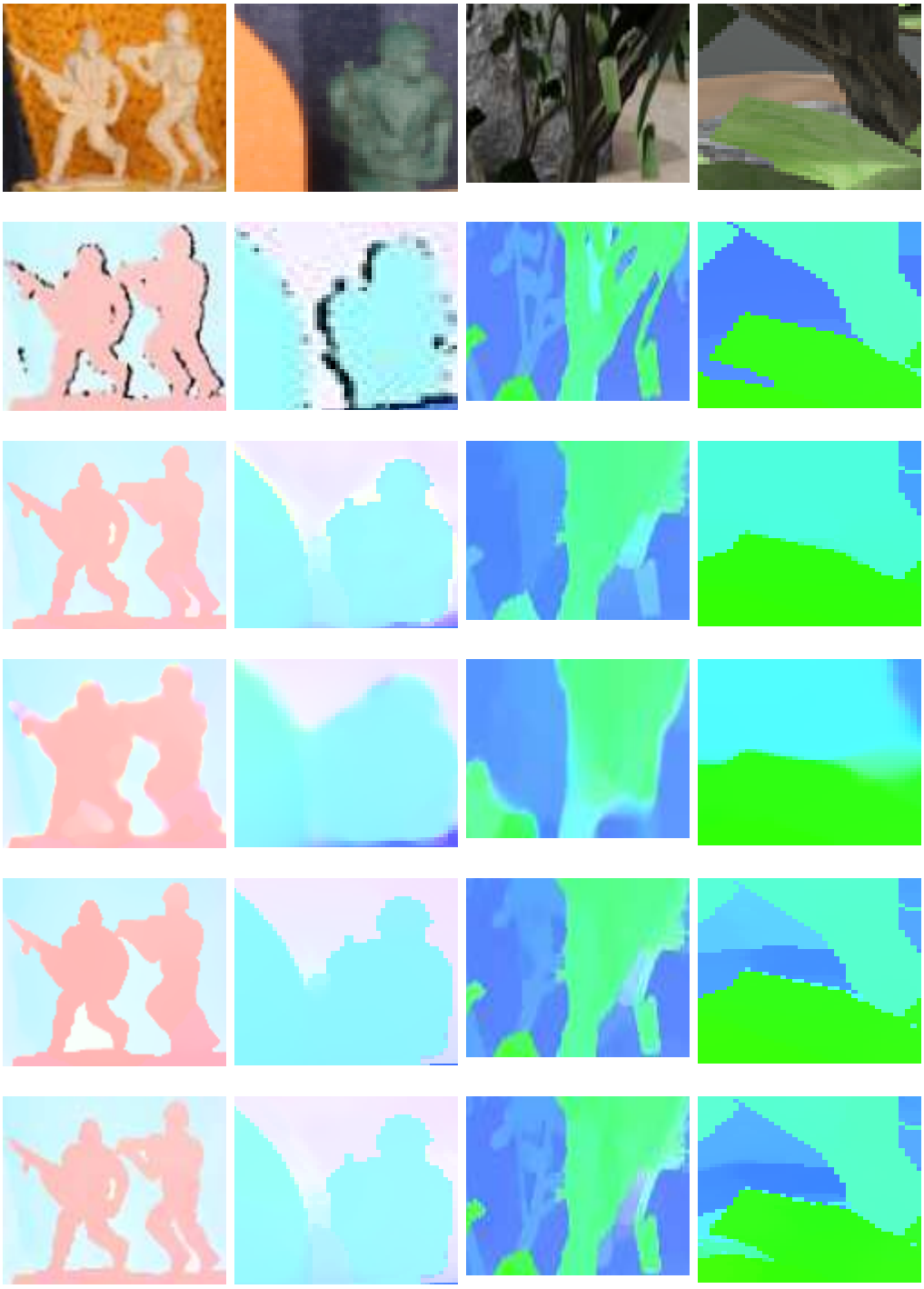


Figure 4.8: (row 1) Original crop of Middlebury sequences "Army" and "Grove2". (row 2) Corresponding ground-truth. (row 3) Resulting flow fields with the baseline method TV. (row 4) Resulting flow fields with ST+DS. (row 5) Resulting flow fields with ST+DS+SW. (row 6) Resulting flow fields with ST+DS+NW.

Chapter 5

Illumination-Robust Optical Flow Model Based on Histogram of Oriented Gradients

Outdoor video surveillance systems require to cope with several surrounding environment factors such as objects shadow and illumination changes. The brightness constancy assumption has widely been used in variational optical flow approaches as their basic foundation as illustrated in Chapter 3 and 4. Unfortunately, this assumption does not hold when the illumination changes or for objects that move into a part of the scene with different illumination. This chapter proposes the replacement of the classical data term depend on either the brightness constancy assumption or high-order constancy assumptions, such as the gradient constancy, by a texture constancy assumption based on a robust feature descriptor. The proposed method is a variation of the L1-norm dual total variational optical flow model with a new robust data term defined from the histogram of oriented gradients computed for two consecutive frames. In addition, a weighted non-local term is utilized for denoising the resulting flow field. Experiments with complex textured images belonging to different scenarios show results comparable to state-of-the-art optical flow models, although being significantly more robust to illumination changes.

The rest of the chapter is organized as follows. Section 5.1 introduces for the relative work. The HOG descriptor and different descriptors used for extract features of an image and their benefits and shortcomings are discussed in Section 5.2. In addition, Section 5.3 summarizes the proposed variational optical flow model, which consists of a data term, a regularization term and a weighted non-local term. Finally, experimental results are shown and discussed in Section 5.4, including a comparison with state-of-the-art optical flow methods.

5.1 Introduction

Optical flow allows the estimation of the apparent motion of the scene. Motion estimation is a key task of video surveillance systems (VSS). The VSS require robust optical flow methods that are able to cope with different dramatically changing scenarios. The robustness of optical flow is badly affected by several surrounding environment factors such as fog, sunshine, clouds, shadow, shading, and lighting changes that yield brightness changes between two consecutive images.

A wide variety of optical flow approaches have been proposed during the last years achieving outstanding levels of accuracy such as Middlebury datasets. Among them, the variational approaches are considered to provide the best results due to their ability to fill gaps where motion information is not available as mentioned in Chapter 3 and 4. However, most of these techniques are based on two main assumptions: brightness and high-order constancy assumptions, such as gradient constancy. Both constancy assumptions respectively depend on the brightness and the derivative of the brightness of the pixels contained in a given pair of images. However, the brightness of a point on an object can dramatically change if the object moves to another part of the scene with different illumination or after global or local illumination changes Kim et al. (2005), see Figure 5.1.

Figure 5.1: four images for the same scene with different illumination changes.

In order to reduce this dependency on brightness, classical approaches apply a structure-texture decompensation of the input images, such as ROF algorithm proposed in Rudin et al. (1992), as a preprocessing stage to reduce the effect of noise and illumination changes. In addition, Mattavelli and Nicoulin (1994) has suggested a more realistic model by assuming that the brightness at time $t + dt$ is related to the brightness at time t through a set of parameters that can be estimated from the image sequence. However, Mattavelli and Nicoulin (1994) fails at estimating accurate motion discontinuities. In turn, Kim et al. (2005) has solved this problem through an approach that simultaneously deals with motion discontinuities and large illumination variations in an integrated framework by taking into account multiplicative and additive illumination factors. Notwithstanding, the accuracy of the estimated optical flow field can be affected by the coupling

between the two factors and the corresponding components of the flow field and, in addition, the optimization problem becomes much more complex.

Furthermore, Mileva et al. (2007) proposed a photometric invariants of the dichromatic reflection model. However, this model is only applicable to color images with brightness variations. In turn, Molnár et al. (2010) has proposed both a non-linear scheme and a linearized scheme for a variational optical flow model based on the normalized cross-correlation in order to implement a illumination-robust data term. In addition, Werlberger et al. (2010) has incorporated a low-level image segmentation process by considering an illumination-robust data term based on the normalized cross correlation, as well as a non-local term in order to tackle the problems of poorly textured regions, occlusions and small scale image structure in order to preserve motion discontinuity.

In turn, Zimmer et al. (2011) has presented an advanced data term that is robust to outliers and varying illumination conditions by using constraint normalization, as well as an HSV color representation with high-order constancy (gradient constancy) assumptions to cope with illumination changes. In addition, Zimmer et al. (2011) have proposed the complementary regularization term in order to get accurate motion discontinuities (see Chapter 3). However, the data term based on gradient constancy is affected by large illumination changes and it is very sensitive to noise.

Related work

Recently, Müller et al. (2011) has proposed the census transform descriptor in order to implement a texture constancy assumption by replacing the classical data term by the Hamming distance between two census transform signatures. Unfortunately, the census transform is not accurate enough and has various shortcomings, such as the inability to discriminate between dark and bright regions in a neighborhood, as well as being very sensitive to noise due to its dependency on the brightness values.

In addition, Liu et al. (2011) has proposed a method based on the SIFT descriptor to compute a dense correspondence field between two images through a discrete optimization based on a belief propagation approach. While, the SIFT flow algorithm proposed in Liu et al. (2011) is based on matching or visual features and yields pixel accuracy, the optical flow model proposed in this chapter is based on the classical motion estimation and yields sub-pixel accuracy. In turn, Brox and Malik (2011) integrates a discrete pixel matching term based on a HOG/SIFT-like descriptor into the continuous variational energy function in order to cope with large displacements while preserving the classical data term based on the brightness and high-order constancy assumptions.

Therefore, this chapter introduces a new optical flow model that can be used

for outdoor VSS. This model proposes the replacement of the classical brightness constancy assumption by a local texture descriptor that is highly invariant to illumination changes. In particular, the Histogram of Oriented Gradients (HOG) is proposed as a texture descriptor in order to extract texture features from two consecutive images (see Dalal and Triggs, 2005). These features are then utilized in order to implement a texture constancy assumption for the data term of the total variation with L1 norm (TV-L1) optical flow model Zach et al. (2007). In addition, the loss of accuracy of the estimated flow field due to the use of an isotropic regularization term is compensated with an additional weighted non-local term similar to the one proposed in Sun et al. (2010a).

5.2 Texture features descriptors

Many approaches have been used for extracting the features from an image to use it in a robust data term. The classical variational optical flow approaches used gradient constancy (GC) introduced in Chapter 3.6.1, and structure texture decomposition via total variation (ROF) (Rudin et al., 1992, Chambolle and Lions, 1997) for illumination robust data term. In turn, Census transform (CT) Zabih et al. (1994) and its variations Modified (mean and Median) census transform Froba and Ernst (2004), ternary census transform Stein (2004) have been used for improve the data term to be more robust against illumination changes. In addition, the well-known histogram of oriented gradients (HOG) proposed in Dalal and Triggs (2005), which is used for the people detection in a scene have utilized in this chapter as a texture descriptor in order to get a robust data term for the TV-L1 variational optical flow model.

5.2.1 Structure texture decomposition via total variation (ROF)

Decomposing an image into meaningful components is an important topic in image processing, Rudin et al. (1992). Range of images are denoising by assuming that images have been contaminated by noise, and the decomposing purpose is to remove the noise. This task can be regarded as a decomposition of the image into signal parts and noise parts. Certain assumptions are taken with respect to the signal and noise, such as the piecewise smooth nature of the image, which enables good approximations of the clean original image. Recently, the main successful approaches for denoising images are based on solving nonlinear partial differential equations (PDE's) associated with the minimization of an energy function composed of some norms of the gradient. One of the best decomposition method proposed in Rudin et al. (1992) and Chambolle and Lions (1997) is a popular

denoising algorithm, which preserves well the edges of the original image, while removing most of the noise.

This algorithm decomposes an image I into two components I_u and I_v. In this approach, the following functional is being minimized

$$\min_{(I_u,I_v)/I=I_u+I_v}(\int \mid DI_u \mid +\lambda\|I_v\|^2), \tag{5.1}$$

5.2.2 Gradient constancy

Recently, many optical flow methods depend on the gradient constancy to cope the illumination changes. Therefore, it is possible to circumvent the problem by considering that the gradient of an object does not change with the motion of the object. This yields the so-called gradient constancy assumption or gradient constraint between two images $I1(x, y, t)$ and $I2(x + u, y + v, t + dt)$,, which is formulated in Chapter 3.6.1 as:

$$\nabla_3 I1(x, y, t) - \nabla_3 I2(x + u, y + v, t + dt) = 0. \tag{5.2}$$

5.2.3 Census transform (CT)

Census transform is a form of non-parametric local transform (*i.e.* relies on the relative ordering of local intensity values, and not on the intensity values themselves) used in image processing to map the intensity values of the pixels within a square window to a bit string, thereby capturing the image structure. The intensity value of the center pixel is replaced by the bit string composed of set of boolean comparisons such that in a square window, moving left to right. For each comparison the bit is shifted to the left, forming an 8 bit string for a census window of size 3×3 and a 24 bit string for a census window of size 5×5, depending on:

$$\xi(P, P') = \begin{cases} 1 & I(x, y) >= I(x + i, y + j) \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

Census transform can reduce effects of variations caused by the camera gain and bias. In addition, It can increase the robustness to outliers near depth-discontinuities. Furthermore, it can also encodes local spatial structure. If a minority of pixels in a local neighborhood has a very different intensity distribution than the majority, only comparisons involving a member of the minority are affected.

Modified Census transform

Modified census transform is a non-parametric local transform that modified census transform introduced in Zabih et al. (1994). It is an rendered set of comparisons of pixel intensities in a local neighborhood that represents which pixels have an intensity value greater than the mean or the median pixel intensity value within a certain window Froba and Ernst (2004). For each comparison the bit is shifted to the left, forming a 9 bit string for a census window of size 3×3 and a 25 bit string for a census window of size 5×5. The modified census is very useful to distinguish between the darkness and brightness regions that census transform fails to detect it.

Ternary census transform

Ternary census transform maps a local neighborhood surrounding a pixel p to a ternary string representing the set of neighbors pixels. Each ternary census vector $\xi(p, p')$ is defined in Stein (2004) as:

$$\xi(P, P') = \begin{cases} 0 & p - p' \quad > \varepsilon \\ 1 & |p - p'| \quad \leq \varepsilon \\ 2 & p' - p \quad > \varepsilon \end{cases} \tag{5.4}$$

where ε is a threshold. In this transform, the intensity of a pixel is compared with the median pixel value. For each comparison the bit is shifted to the left, forming a 9 bit string for a census window of size 3×3 and a 25 bit string for a census window of size 5×5.

5.2.4 Histogram of oriented gradients

Histograms of oriented gradients (Dalal and Triggs, 2005) are a robust visual descriptor that allows the discrimination of the objects present in a scene, since the local appearance and shape of objects can be characterized to a large extent by the local distribution of intensity gradients, which, in addition, is largely invariant to shadows and illumination changes.

The HOG descriptor proposed in Dalal and Triggs (2005) is based on dominant edge orientations. The gradient operator has been applied by computing local image gradients, d_x and d_y, within a local window (3×3 or 5×5) using a centered derivative mask. The magnitudes and orientations of the resulting derivatives for every window are computed. In addition, the orientations are divided into n localized bins. In practice, the angles between 0 and 2π are divided into a number of bins (experimentally set to 8 in this work). The value of each bin is obtained by summing the magnitudes of the gradients whose orientations are mapped to

that bin. The obtained histogram is normalized using $L2-norm$, $L1-norm$ or $L2-sqrt$ Dalal and Triggs (2005).

The pair of images $I_1(x,y)$ and $I_2(x+u,y+v)$ used to estimate the optical flow field yield multi-channel images $S_1(x,y)$ and $S_2(x+u,y+v)$, respectively. $S_1(x,y)$ consists of n channels, such that each channel contains the values corresponding to an orientation bin of the resulting normalized histogram for every pixel.

In order to illustrate the advantages of the HOG descriptor with respect to the census transform, Figure 5.2 shows the features extracted with the census transform, as well as the HOG signatures for two windows that contain both a bright and a dark region. In particular, Figure 5.2 shows the gray values within a 3×3 window with a central pixel equal to 110, for the first and the second window. The census transform has been computed for the two windows yielding the 8-bit string 11111111, since all neighbors are larger than the value of the central pixel. On the other hand, the features with the HOG descriptor have been computed after calculating the 8 bin histogram of the resulting orientations of the input windows. For simplicity, a centered mask has been used for computing the gradients and each bin has been obtained by counting the number of orientations associated with that bin. The HOG descriptors obtained for the input windows are 01131111 and 03210210, respectively. Clearly, the census transform produces the same code for different textures and is not able to cope with image blocks with a saturated center pixel, whereas HOG can detect changes in the intensity regions by yielding a different descriptor.

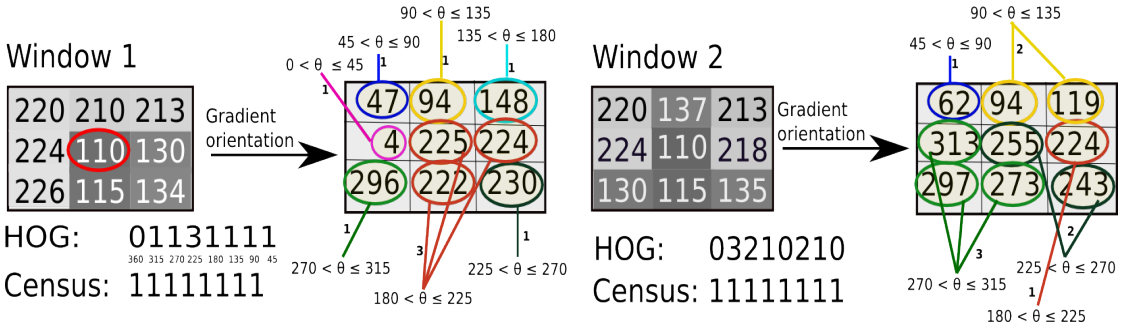


Figure 5.2: Comparison of a 3×3 HOG descriptor vs. a 3×3 census descriptor.

5.3 Optical flow model

Let flow field $w = (u,v)$ is defined as the apparent motion of pixels between a frame $I_1(x,y)$ at time t and a frame $I_2(x+u,y+v)$ at time $t+1$. The duality of the TV-L1 optical flow model Zach et al. (2007) is used to compute the vector

flow field w associated with every pixel $p = (x, y)$ belonging to the image domain Ω based on the optical flow energy functional (3.11) that can be reformulated as:

$$\arg min_{(w)} \int_{\Omega} (\lambda E_D(w) + E_s(w)) \ d\Omega, \tag{5.5}$$

where E_D is a data term, E_s a regularization term (a total variational term Zach et al. (2007)) and λ is the weight of the data term. The energy functional is divided into two parts that are solved iteratively:

$$\arg min_{(w)} \int_{\Omega} (\lambda E_D(w) + E_c(w, \hat{w})) \ d\Omega, \tag{5.6}$$

$$\arg min_{(\hat{w})} \int_{\Omega} (E_c(w, \hat{w}) + E_s(\hat{w})) \ d\Omega, \tag{5.7}$$

where E_c is a coupling term and $\hat{w}$ an auxiliary vector flow field.

5.3.1 Data and regularization terms

In this chapter, the data term includes the residual of two texture features extracted from the input images in order to ensure texture constancy:

$$\arg min_{(w)} \int_{\Omega} \left(\lambda \psi(S(x, y, w)) + \frac{1}{\theta}(w - \hat{w}) \right) \ d\Omega, \tag{5.8}$$

where $\frac{1}{\theta}$ is the weight of the coupling term. $S(x, y, w)$ can be formulated as:

$$S(x, y, w) = S_2(x + u, y + v) - S_1(x, y) = 0, \tag{5.9}$$

such that $S_1(x, y)$ and $S_2(x + u, y + v)$ are the texture features extracted from two consecutive images $I_1(x, y)$ and $I_2(x + u, y + v)$, respectively. In turn, $\psi(x)$ is a convex penalization function. Thus, (5.9) implements a texture constancy assumption that assumes that texture features do not change when objects move.

The residual S can be linearized around the starting value w using first order Taylor expansion as:

$$\begin{aligned} S(x, y, w) \approx \tilde{S}(x, y, w) &= (S_2(x, y) - S_1(x, y)) + \nabla^T S(x, y, \hat{w})(w - \hat{w}), \\ &= S_t + \nabla^T S(x, y, \hat{w})(w - \hat{w}), \end{aligned} \tag{5.10}$$

where $\nabla^T S(x, y, \hat{w}) = [\frac{\partial S}{\partial x} = S_x, \frac{\partial S}{\partial y} = S_y]$. Now, (5.8) can be solved for $w = (u, v)$ by doing:

$$\frac{\partial}{\partial u}(\lambda\psi(\tilde{S}(x,y,w)) + \frac{1}{\theta}(u-\hat{u})) = 0,$$
$$\frac{\partial}{\partial v}(\lambda\psi(\tilde{S}(x,y,w)) + \frac{1}{\theta}(v-\hat{v})) = 0. \quad (5.11)$$

Both equations can be expressed in vector form as proposed in Müller et al. (2011):

$$\lambda\frac{\psi'(\tilde{S}(x,y,\hat{w}))}{\tilde{S}(x,y,\hat{w})}\tilde{S}(x,y,w)\nabla S(x,y,\hat{w}) + \frac{1}{\theta}(w-\hat{w}) = 0. \quad (5.12)$$

Since (5.12) is linear in (u,v), it can be solved as a linear system, $Aw = b$. In addition, the final data term can be extended in order to be applicable to a multi-channel descriptor:

$$\arg min_{(w)} \int_\Omega \left(\lambda\sum_{i=1}^{n}\psi(\tilde{S}_i(x,y,w)) + \frac{1}{\theta}(w-\hat{w})\right) d\Omega, \quad (5.13)$$

where n is the number of channels of the texture descriptor used in the data term. Hence, A and b can be written as:

$$A = \begin{pmatrix} \frac{1}{\theta} + \lambda\sum\psi'(\tilde{S}(x,y,\hat{w}))\sum S_{i_x}^2 & \lambda\sum\psi'(\tilde{S}(x,y,\hat{w}))\sum S_{i_x}S_{i_y} \\ \lambda\sum\psi'(\tilde{S}(x,y,\hat{w}))\sum S_{i_x}S_{i_y} & \frac{1}{\theta} + \lambda\sum\psi'(\tilde{S}(x,y,\hat{w}))\sum S_{i_y}^2 \end{pmatrix} \quad (5.14)$$

and:

$$b = \frac{1}{\theta}\begin{pmatrix}\hat{u}\\ \hat{v}\end{pmatrix} - \lambda\sum\psi'(\tilde{S}(x,y,\hat{w}))\begin{pmatrix}\sum S_{i_x}\\ \sum S_{i_y}\end{pmatrix}\left(\sum S_{i_t} - \left(\sum S_{i_x}\hat{u} + \sum S_{i_y}\hat{v}\right)\right). \quad (5.15)$$

Similarly, the smoothness term represents the isotropic total variation Chambolle (2004). As a result, (5.7) can be decomposed into two equations and rewritten as:

$$E_u = \int_\Omega \left(\frac{1}{\theta}(u-\hat{u}) + \parallel \nabla\hat{u} \parallel\right) d\Omega, \quad (5.16)$$

$$E_v = \int_\Omega \left(\frac{1}{\theta}(v-\hat{v}) + \parallel \nabla\hat{v} \parallel\right) d\Omega. \quad (5.17)$$

E_u and E_v have two unknowns, $\hat{u}$ and $\hat{v}$, while u, v are constants obtained after solving the data term.

For E_u, thus the Euler-Lagrange equation is:

$$-div[\frac{\nabla u}{\parallel \nabla u \parallel}] + \frac{1}{\theta}(u-\hat{u}) = 0 \quad (5.18)$$

Let $P_u = \nabla u / \parallel \nabla u \parallel$. Thus:

$$u = \lambda div(P_u) + \hat{u}, \tag{5.19}$$

By using (5.18) and (5.19), P_u can be rewritten as:

$$P_u^{h+1} = \frac{P_u^h + \tau \nabla (div(P_u^h) + \frac{\hat{u}}{\theta})}{1 + \tau \parallel \nabla (div(P_u^h) + \frac{\hat{u}}{\theta}) \parallel}, \tag{5.20}$$

where h is the iteration number, and $\tau \leq 1/8$ is the time step. The same can be applied to get P_v. That equations can be solved through a fixed-point iteration scheme as described in Zach et al. (2007).

Furthermore, a multi-scale, coarse-to-fine scheme is used for solving the energy functional (5.5) in order to allow for both small and large displacements and to improve the accuracy of the estimated flow fields. In each pyramid level, the scaled images are warped representations of the input images based on the flow estimated at every preceding scale Brox et al. (2004) as illustrated in Chapter 3.

5.3.2 Anisotropic filtering based on a weighted non-local term

The smoothing term utilized in the energy functional (5.5) described above is isotropic and propagates the flow field in all directions. Thus, flow vectors near motion discontinuities are usually inaccurate due to occlusions and over-smoothing. In order to tackle this problem, the resulting flow fields at every pyramid level require a denoising stage in order to preserve edge and object boundaries and details. Therefore, the estimated flow fields are improved by detecting motion boundaries through the Sobel operator, and then by dilating the detected regions through a 5×5 mask in order to obtain flow boundary regions. For each pixel $p = (x, y)$ in these regions, a robust weighted median filter proposed in Sun et al. (2010a) is applied as described in Chapter 4 in (4.23):

$$E_w = \sum \sum_{(\acute{x},\acute{y}) \in N_{x,y}} \varpi_{p,\acute{p}} (|\hat{u}_{x,y} - \hat{u}_{\acute{x},\acute{y}}| + |\hat{v}_{x,y} - \hat{v}_{\acute{x},\acute{y}}|), \tag{5.21}$$

where $\varpi_{p,\acute{p}}$ is a weighting function that takes into account the occlusion state of pixels, $O(p)$, as proposed in Sand and Teller (2008), as well as the intensity difference and the spatial distance. Thus, $\varpi_{p,\acute{p}}$ is formulated as:

$$\varpi_{p,\acute{p}} \propto exp\left(-\frac{(p - \acute{p})^2}{2\sigma_s^2} - \frac{(I(p) - I(\acute{p}))^2}{2\sigma_r^2} \right) \frac{O(\acute{p})}{O(p)}, \tag{5.22}$$

where $I(p)$ and $I(\acute{p})$ are the intensity values of pixels p and $\acute{p}$, respectively, and σ_s and σ_r are standard deviations experimentally set to 7.0 and 7.0, respectively.

5.4 Experiments

A qualitative comparison have been done using real images (2144×1424) for the same scene with different global illuminations. Figure 5.3 shows a real example that compares the performance of HOG, the census transform, the gradient constancy (GC) and the structure-texture decompensation ROF Rudin et al. (1992). The comparison is performed by computing the histogram of normalized errors between the same two features extracted from the pair of images. For the census transform (CT), the error is computed based on the Hamming distance between the two descriptors (binary descriptors). In turn, the error generated for HOG and ROF is the difference between the resulting features. In addition, the similarity between the pair of input images is obtained for the gradient constancy. As shown in the Figure, the gradient constancy yields the smallest average error($AE = 0.0146$) among the different tested descriptors. However, HOG detects the largest number of pixels with zero error among them, as well as it yields a good average error ($AE = 0.0184$). Thus, HOG is likely to be advantageous for motion estimation under illumination changes.

In another experiment, the variational optical flow model described in Section 5.3 has been tested with different features descriptors by using sequence GROVE2 from the Middlebury datasets with ground-truth by changing the illumination of the second frame as:

$$I_o = uint8\left(255\left(\frac{mI_i + a}{255}\right)^{\gamma}\right), \tag{5.23}$$

where I_i and I_o are the input and output frames, respectively. $m > 0$ is a multiplicative factor, a is an additive change factor and $\gamma > 0$ is the gamma correction. The function $uint8$ is used for quantizing the values to an 8-bit unsigned integer format. Figure 5.4 shows a qualitative comparison of the average end-point error (AEE) and the average angular error (AAE) between the flow fields obtained with HOG and CT, both determined in a 3×3 neighborhood, as well as GC. The effects of different values of m, a and γ have individually been assessed by varying γ while keeping $m = 1$ and $a = 0$, by changing m with $\gamma = 1$ and $a = 0$, as well as by changing a while keeping $m = 1$ and $\gamma = 1$.

As shown in Figure 5.4, the gradient constancy is robust against small changes of both γ and m. In turn, HOG shows a higher robustness against both small and large changes of γ, a and m. In addition, the census transform yields adequate values for both AEE and AAE.

Additionally, the effect of the weighted non-local term on the final proposed algorithm was evaluated. The AEE and the percentage of the bad pixels (BP)

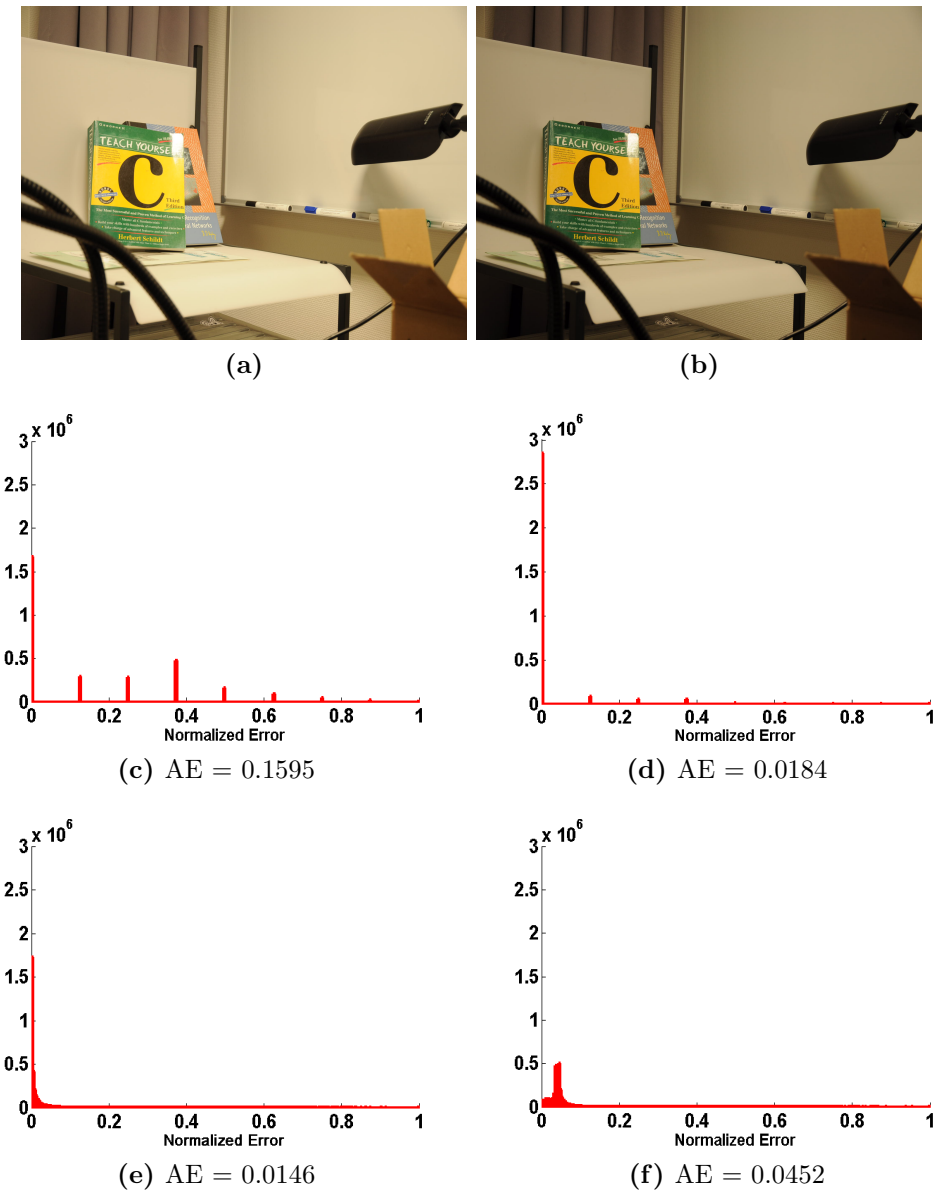


Figure 5.3: (a-b) Two original images. Error histograms for (c) CT, (d) HOG, (e) GC, and (f) ROF.

of the obtained flow fields with 8 KITTI training sequences [1] are calculated for the proposed optical flow technique TV-L1 based on HOG with and without the weighted non-local term and are shown in table 5.1. As shown, the values of both

[1] http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php

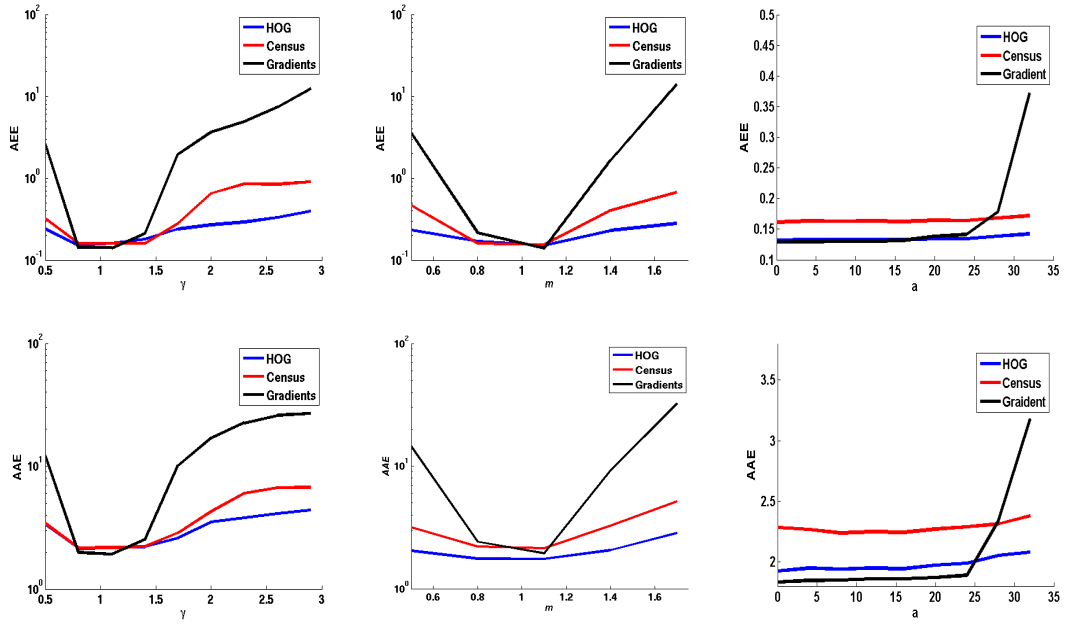


Figure 5.4: AEE and AAE for HOG, census transform and gradient constancy. Column 1: change of γ; Column 2: change of m; Column 3: change of a

AEE and BP for the proposed algorithm are reduced due to the detected accurate borders after using the weighted non-local term. In addition, the use of a weighted non-local term yields more accurate flow fields. In Figure 5.5, the color flow field, the error image and the histogram of error with and without the non-local term was visualized.

At the time of submission (April 2013), the results of the proposed model with HOG (TVL1-HOG) have been evaluated with the KITTI Vision Benchmark, which contains 195 testing image sequences with ground truths, and it has been ranked in the seven position against current state-of-the-art optical flow algorithms [2]. The KITTI benchmark considers the bad flow vectors at all pixels that are above a spatial distance of 3 pixels from the ground truth. (TVL1-HOG) has average of 8.31% bad pixels as shown in table 5.2, in turn the baseline methods Zach et al. (2007) and Sun et al. (2010a) have 30.75% and 24.64%, respectively.

Furthermore, the proposed variational optical flow method based on the HOG descriptor is evaluated with eight real image sequences that include illumination changes and large displacements, as well as low-textured areas, reflections and specularities. Tables 5.3 and 5.4 show the AEE and bad pixels corresponding to

[2]http://www.cvlibs.net/datasets/kitti

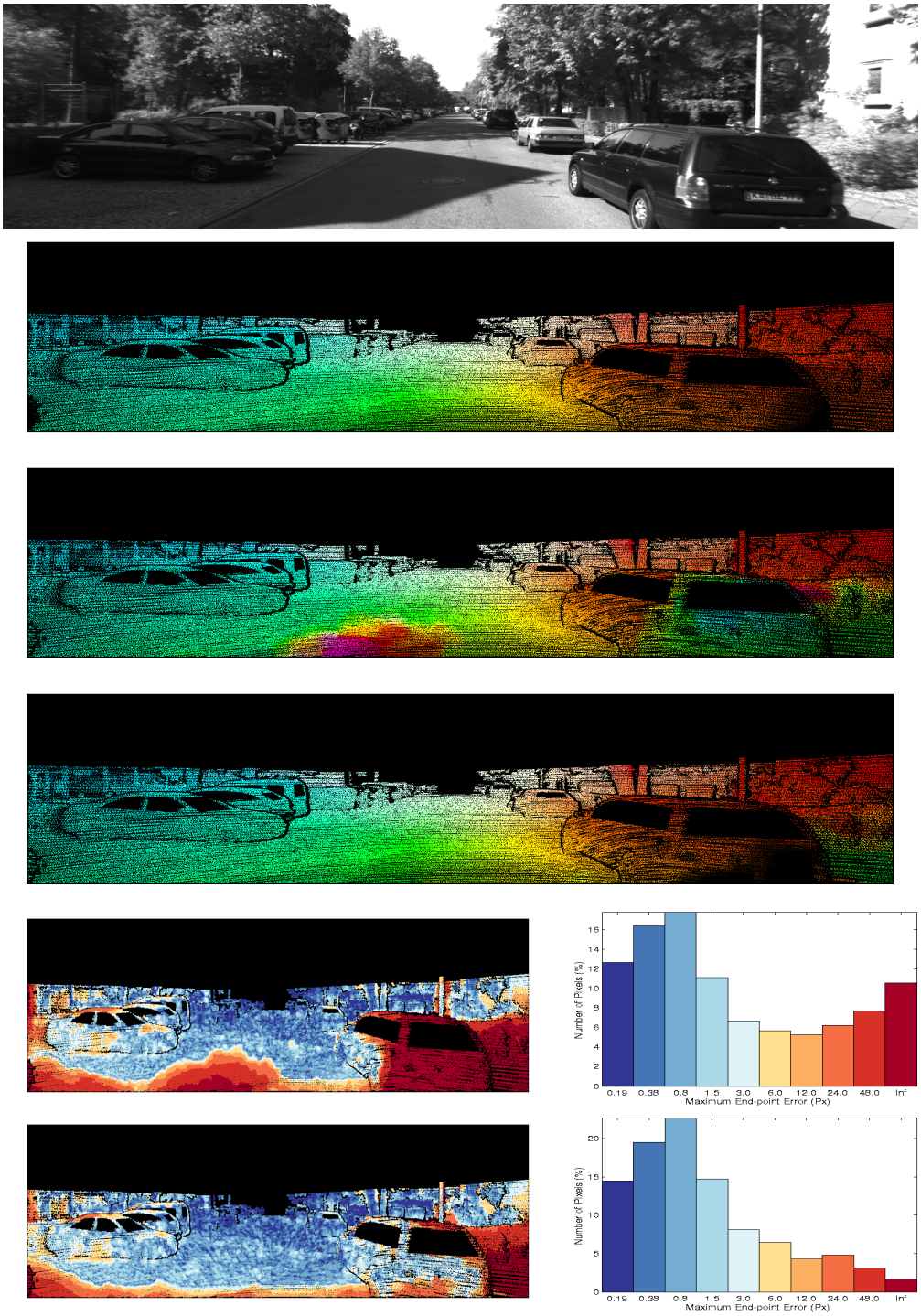


Figure 5.5: Optical flow model, Row 1: Original image for sequence 44 of the KITTI datasets, Row 2: Ground truth. Row 3: Resulting flow field without non-local term, Row 4: Resulting flow field with non-local term, Row 5: Error image and error histogram without non-local term, Row 6: Error image and error histogram with non-local term.

Sequence	TV-L1	TV-L1 with non-local term
11	35.49%(15.77)	29.92%(8.90)
15	26.55%(13.21)	22.30%(6.48)
44	35.46%(14.70)	21.45%(4.68)
74	61.41%(24.41)	52.74%(19.79)
117	31.58%(15.22)	18.50%(12.27)
144	47.96%(20.03)	31.64%(12.86)
147	18.39%(11.22)	12.42%(2.87)
181	59.40%(48.78)	44.89%(33.72)

Table 5.1: The percentage of bad pixels and average end-point error of the proposed technique with and without the non-local term.

Rank	Method	Out-Noc	Out-All	Avg-Noc	Avg-All
1	PR-Sf+E	4.08 %	7.79 %	0.9 px	1.7 px
2	PCBP-Flow	4.08 %	8.70 %	0.9 px	2.2 px
3	MotionSLIC	4.36 %	10.91 %	1.0 px	2.7 px
4	PR-Sceneflow	4.48 %	8.98 %	1.3 px	3.3 px
5	TGV2ADCSIFT	6.55 %	15.35 %	1.6 px	4.5 px
6	Data-Flow	8.22 %	15.78 %	2.3 px	5.7 px
7	**TVL1-HOG**	**8.31 %**	**19.21 %**	**2.0 px**	**6.1 px**
8	MLDP-OF	8.91 %	18.95 %	2.5 px	6.7 px
12	fSGM	11.03 %	22.90 %	3.2 px	12.2 px
13	TGV2CENSUS	11.14 %	18.42 %	2.9 px	6.6 px
14	C+NL-fast	12.42 %	22.27 %	3.2 px	7.8 px
25	DB-TV-L1	30.75 %	39.13 %	7.8 px	14.6 px

Table 5.2: The current evaluation of the state-of-the-art method on the KITTI website.

four sequences with illumination changes calculated for the methods proposed in Zimmer et al. (2011) (OFH), Sun et al. (2010a) (SRB), full version of Sun et al. (2010a) (SRBF), Bruhn and Weickert (2005) (BW), Horn and Schunck (1981) (HS), and Werlberger et al. (2010) (WPB), in addition to the proposed method based on HOG5 (5×5), HOG3 (3×3), the census transform (CT5 with (5×5)) and (CT3 with (3×3)), and the gradient constancy (GC) with respect to the occluded ground-truth and non-occluded ground-truth, respectively. In turn, Tables 5.5 and 5.6 show the same information for four sequences with large displacements

Method	44	11	15	74	Average
HOG3	**21.45% (4.68)**	32.54% (9.04)	**22.30% (6.48)**	53.79% (20.03)	**32.52%**
HOG5	23.23% (5.22)	**29.92% (8.90)**	24.90% (7.64)	**52.74% (19.79)**	32.70%
GC	29.25% (9.54)	35.72%(10.91)	26.41% (8.47)	59.20% (23.07)	37.64%
OFH	23.22% (5.11)	37.26% (12.47)	32.20% (9.06)	62.90% (24.00)	38.89%
CT5	35.23% (12.74)	33.93% (9.75)	29.04% (8.70)	57.57% (20.80)	38.94%
CT3	29.55% (10.22)	37.54% (11.14)	33.74% (9.11)	57.43% (20.53)	39.56%
SRB	26.58%(4.67)	40.61% (13.76)	32.85% (9.72)	62.94% (24.27)	40.74%
SRBF	31.83% (5.62)	40.34% (13.96)	35.13% (12.17)	64.89% (24.64)	43.05%
BW	32.44% (5.19)	33.95% (8.50)	47.70% (12.40)	71.44% (25.15)	46.38%
HS	42.96% (6.77)	38.84% (10.72)	58.08% (12.89)	82.14% (28.75)	55.50%
WPB	49.09% (9.20)	49.99% (28.35)	67.28% (28.36)	88.67% (30.68)	63.76%

Table 5.3: Percentage of bad pixels and AEE for the state-of-the-art methods and the proposed method with four sequences from KITTI datasets: sequences 11, 15, 44 and 74, which include illumination changes with the occluded points ground truth.

Method	44	11	15	74	Average
HOG5	11.35% (2.26)	**15.54% (3.12)**	10.40%(2.41)	**45.76% (13.97)**	**20.76%**
HOG3	**9.98% (2.17)**	18.53 % (3.78)	**8.40% (2.21)**	46.99%(14.20)	20.98%
GC	16.78% (4.95)	19.43%(4.01)	11.97% (3.52)	53.13% (16.38)	25.33%
CT5	24.30% (7.96)	19.83 % (5.06)	15.03% (3.41)	51.10%(15.14)	27.57%
OFH	11.17% (2.44)	24.32% (6.48)	18.34% (3.63)	57.40% (17.25)	27.81%
SRB	14.66% (2.44)	27.83% (6.43)	18.93% (4.05)	57.36% (17.36)	29.69 %
CT3	18.26% (6.30)	24.05 % (7.28)	20.30% (3.95)	57.43%(17.53)	30.01%
SRBF	20.98% (3.29)	27.78% (6.73)	21.66% (4.53)	59.56% (17.52)	32.49%
BW	22.38% (3.16)	20.54% (3.62)	36.85% (6.67)	67.22% (18.49)	36.75%
HS	34.18% (4.61)	25.98% (6.79)	49.57% (7.95)	79.57% (21.55)	47.32%
WPB	40.85% (5.88)	39.25% (18.75)	60.50% (17.63)	87.02% (24.09)	56.90%

Table 5.4: Percentage of bad pixels and AEE for the state-of-the-art methods and the proposed method with four sequences from KITTI datasets: sequences 11, 15, 44 and 74, which include illumination changes with the non-occluded points ground truth.

Method	147	117	144	181	Average
HOG5	14.04% (2.90)	18.5% (12.27)	**31.64 % (12.86)**	**44.89 % (33.72)**	**27.27%**
HOG3	**12.42% (2.87)**	24.49% (14.99)	36.64% (14.40)	55.58% (42.97)	32.28%
OFH	15.04% (4.96)	**16.26% (4.33)**	42.04% (15.01)	63.86% (50.52)	34.30%
GC	12.28% (3.93)	17.70% (10.81)	44.51% (18.67)	67.63% (58.40)	35.53%
SRB	14.59% (4.85)	24.71% (9.74)	50.67% (19.03)	67.11% (47.70)	39.27%
SRBF	14.79% (5.17)	24.41% (9.92)	50.66% (19.34)	68.41% (48.81)	39.57%
BW	16.98% (5.17)	28.80% (7.86)	46.98% (16.85)	69.04% (45.27)	40.45%
CT5	13.98% (3.41)	27.33% (15.23)	47.68% (16.75)	73.85% (58.59)	40.71%
CT3	14.76% (3.54)	28.80% (15.20)	48.97% (16.83)	73.63% (58.58)	41.54%
HS	24.84% (6.61)	43.24% (15.32)	51.89% (14.81)	74.11% (49.28)	48.52%
WPB	32.72% (8.10)	46.80% (13.67)	52.25% (17.94)	76.00% (50.18)	51.94%

Table 5.5: Percentage of bad pixels and AEE for the state-of-the-art methods and the proposed method with four sequences from KITTI datasets: sequences 117, 144, 147 and 181, which include large displacement with the occluded points ground truth.

Method	147	117	144	181	Average
HOG5	6.41% (1.01)	9.09% (5.42)	**16.82% (4.23)**	**27.48% (11.97)**	**14.95%**
HOG3	**5.80% (0.92)**	17.04% (8.04)	22.56% (6.85)	41.44% (18.68)	21.71%
OFH	8.03% (1.98)	**9.09% (2.17)**	29.62% (6.77)	52.32% (23.46)	24.76%
GC	7.13% (1.25)	9.70% (4.42)	32.25% (8.26)	57.21% (29.92)	26.57%
SRB	7.55% (1.74)	18.11% (5.28)	39.55% (9.33)	56.51% (22.88)	30.43%
SRBF	7.69% (1.97)	17.95% (5.29)	39.64% (9.59)	58.25% (23.78)	30.88%
BW	10.07% (2.20)	22.25% (4.23)	35.01% (8.17)	59.05% (22.58)	31.60%
CT5	6.78% (0.95)	20.52% (9.82)	36.29% (7.71)	65.55% (31.26)	32.29%
CT3	6.63% (1.00)	21.85% (9.59)	37.49% (8.43)	65.29% (31.92)	32.82%
HS	18.52% (3.38)	37.82% (9.77)	41.30% (7.32)	65.77% (23.40)	40.85%
WPB	25.92% (4.43)	41.23% (9.18)	41.53% (8.94)	68.27% (25.96)	44.24%

Table 5.6: Percentage of bad pixels and AEE for the state-of-the-art methods and the proposed method with four sequences from KITTI datasets: sequences 117, 144, 147 and 181, which include large displacement with the non-occluded points ground truth.

In another experiment, the estimated flow fields with HOG (3×3) and HOG (5×5) have visually been compared with the proposed optical flow method by using the data term based on the brightness constancy (BC) assumption, as well as the one based on the census transform. Figure 5.6 shows the estimated flow field for sequence 15, which includes illumination changes, as well as the error images and the error histograms. In addition, Figure 5.7 shows the same information for sequence 181, which includes large displacements.

In another experiment, the estimated flow fields with HOG, CT, GC and Brightness constancy (BC) were compared visually. Figure 5.8 shows the esti-

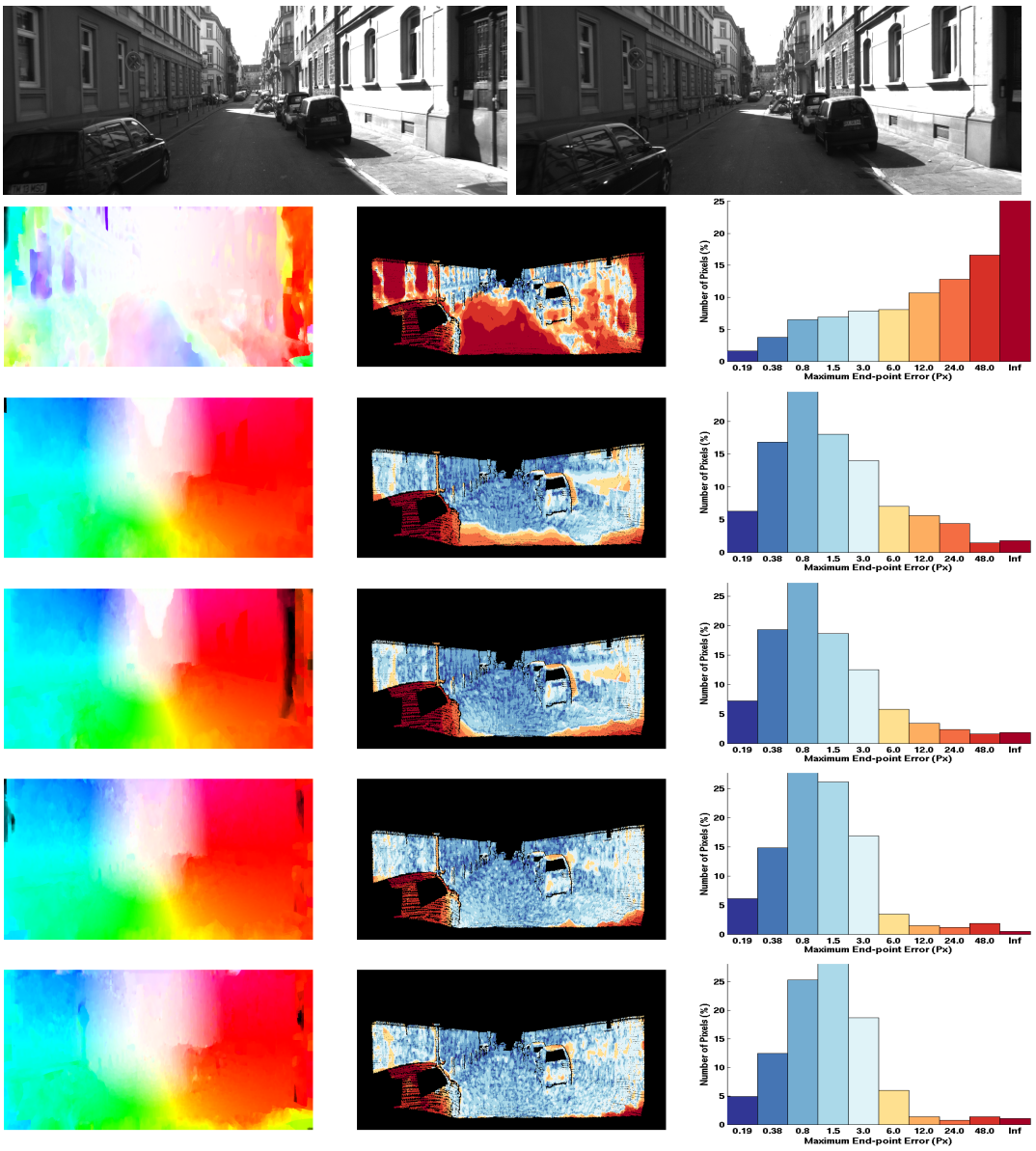


Figure 5.6: (row 1) Two original images for a sequence 15 of KITTI datasets. Resulting flow field, error image and error histogram for the proposed optical flow model with: (row 2) BC, (row 3) 3×3 CT, (row 4) 5×5 CT, (row 5) 3×3 HOG, and (row 6) 5×5 HOG.

mated flow fields for the proposed model with BC, GC, CT and HOG on Cross-Cars, CurTruck and BlinkArrow sequences provided for the currently HCI Bosch Robust Vision Challenge[3].

[3]HCI datasets, http://hci.iwr.uni-heidelberg.de/Static/challenge2012/

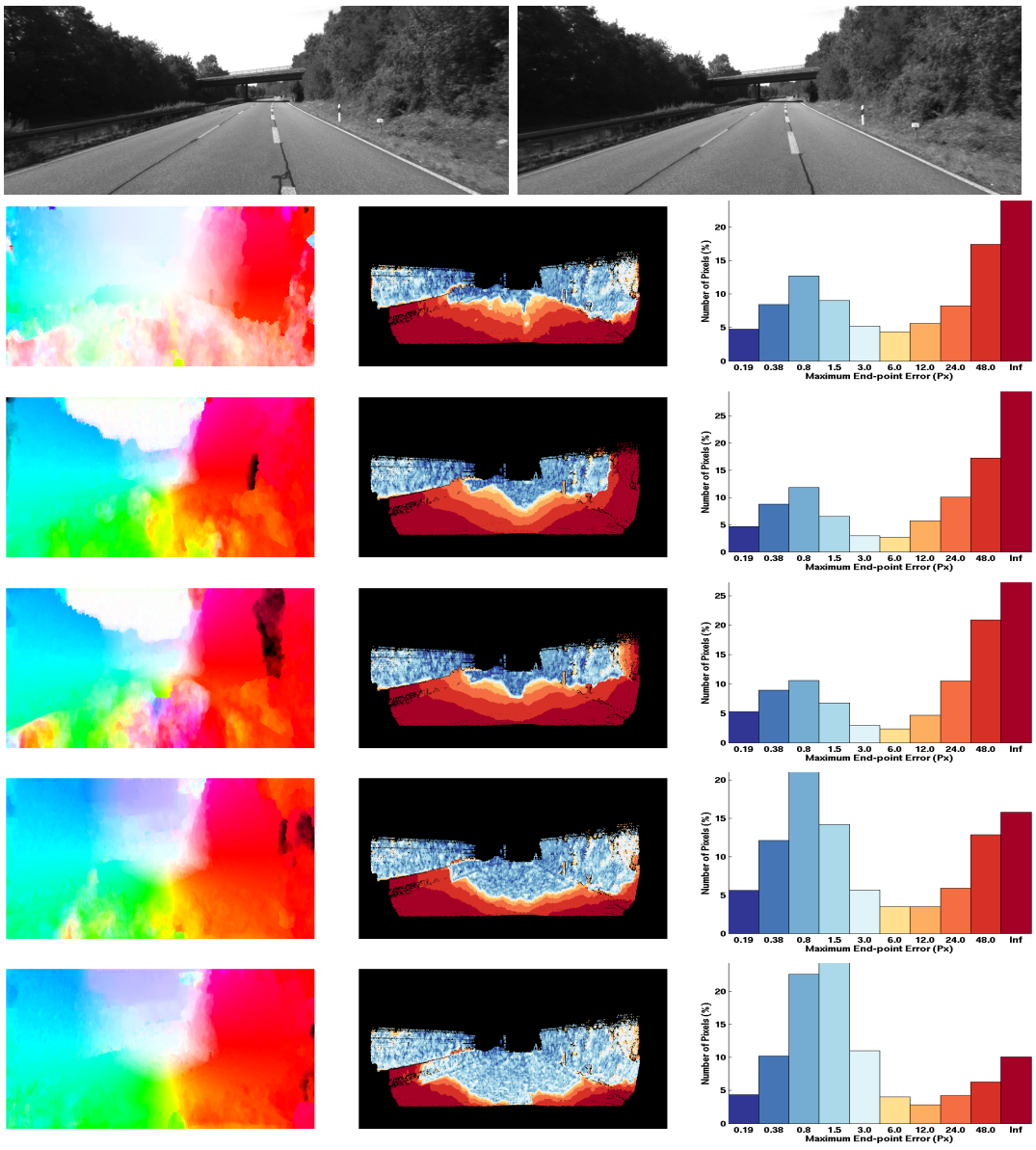


Figure 5.7: Row 1: Two original images for a sequence 181 of KITTI datasets. Resulting flow field, error image and error histogram for the proposed optical flow model with: (row 2) BC, (row 3) 3×3 CT, (row 4) 5×5 CT, (row 5) 3×3 HOG, and (row 6) 5×5 HOG.

Among the evaluated approaches, the optical flow model based on HOG with different window size yields the most accurate flow fields with respect to the state-of-the-art methods for real images from HCI datasets, as well as KITTI datasets that include both illumination changes and large displacements.

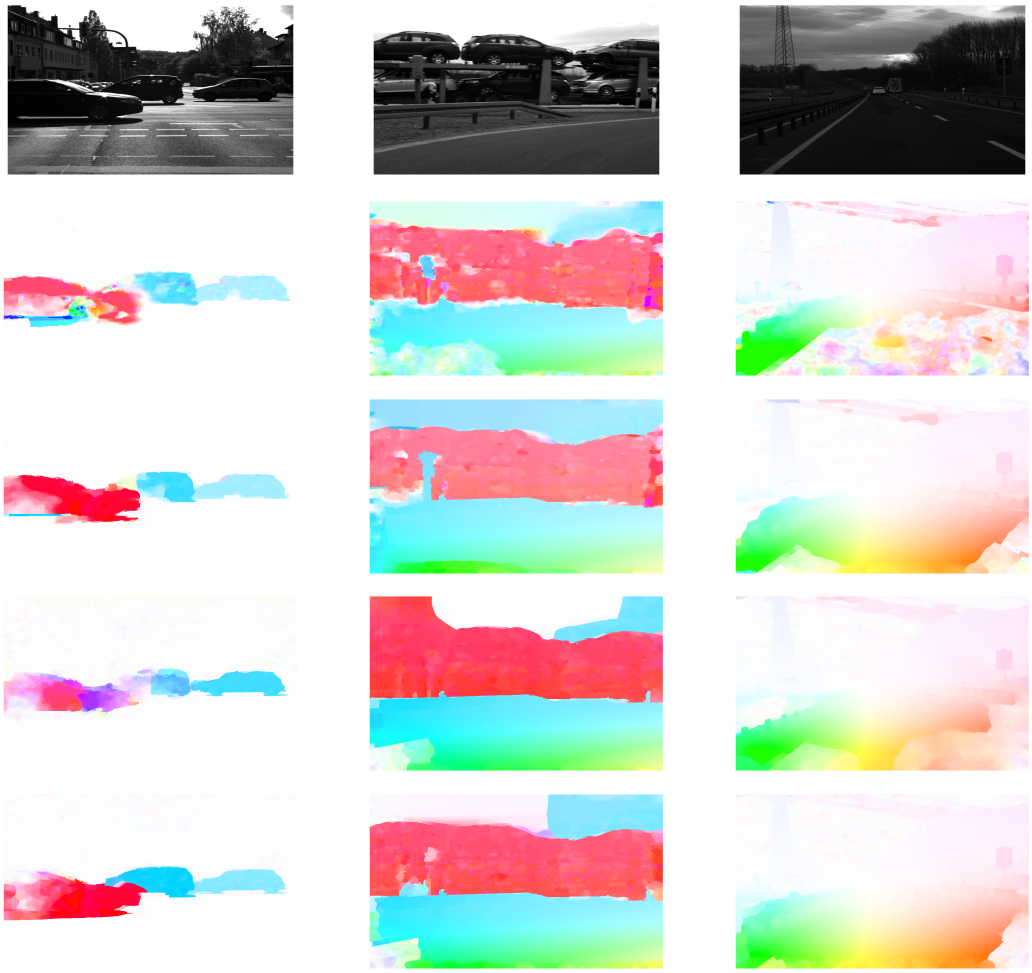


Figure 5.8: (row 1) Original frames of HCI sequences CrossCars, CurTruck and BlinkArrow. (row 2) Corresponding flow fields for the proposed method with BC. (row 3) Corresponding flow fields for the proposed method with GC. (row 4) Corresponding flow fields for the proposed method with CT. (row 5) Corresponding flow fields for the proposed method with HOG.

Chapter 6

A Platform for Trustworthy Storage of Privacy-aware Surveillance Videos

This chapter defines a new methodology for trustworthy video surveillance databases, which fulfills four crucial properties: high accuracy, reversibility, real-time performance and information security. We detect the regions of interest (*e.g.* faces of people), which are protected by means of the alteration of the coefficients of the compressed video stream. The proposed protection guarantees the property of reversibility, since the original coefficients can be restored.

Specifically, we elaborate on the definition of the Protection Stream, which is required for both protecting and unprotecting a video sequence. We propose a procedure for securely generating a protection stream for each group of pictures, thus avoiding the large unsecure streams required if the protection stream was generated for each video file. The proposed model has been implemented and tested, and the results confirm the real time performance of the system developed while keeping the aforementioned crucial properties.

The platform presented in this chapter takes into account the above requirements, and tackles privacy in video surveillance from a holistic point of view: we focus on all the relevant steps involved in the video surveillance system. The rest of the chapter is organized as follows: Section 6.1 introduces for the trustworthy privacy-aware video surveillance systems. In addition, Section 6.2 describes the platform. The detection and protection techniques is described in Section 6.3. Finally, Section 6.4 discusses the effectiveness of the implementation of the platform.

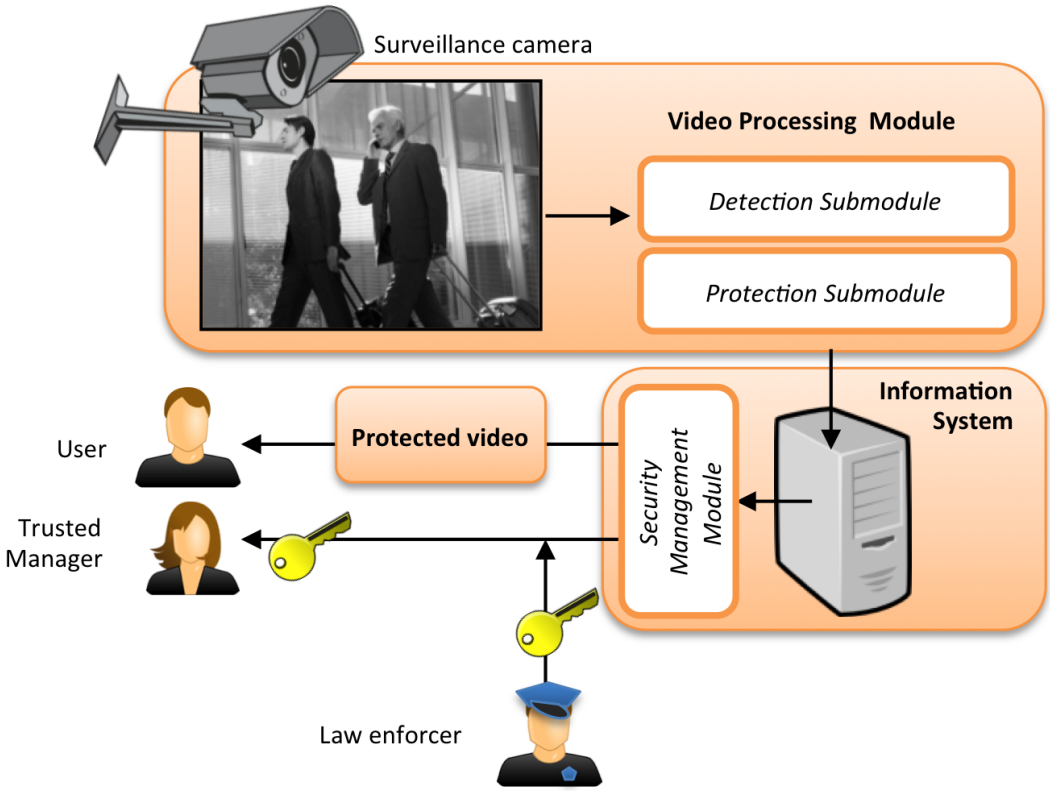


Figure 6.1: Model for a privacy-aware VSS.

6.1 Introduction

As mentioned in Chapter 1, in a trustworthy privacy-aware video surveillance system (TP-VSS) (cf. Figure 6.1), the video from the camera is handled by a *Video Processing Module*, a *Detection Submodule* and a *Protection Submodule*. We also assume in Chapter 1 that any user can retrieve a video file from the database but, since the ROIs are protected, no identity information can be disclosed from it. Only the Trusted Manager of the system, who has access to an *unprotection key* can unprotect the video. Last but not least, the Trusted Manager may need the permission of a Law Enforcer to effectively unprotect the video in case of investigations. Hence, the problem of a trusted manager arbitrarily unprotecting videos is avoided.

Despite of the large amount of literature dealing with ROI detection and protection, there is no proposal for a privacy-preserving video surveillance database that takes into account the concepts of trust and security in a holistic fashion.

6.2 A platform design

In this section we address the basic aspects of the platform. On the one hand, we discuss the technique that uses the Detection Submodule. On the other hand, we describe the method that uses the Protection Submodule. Finally, we present the data stored in the Information System of the platform.

6.2.1 The Detection Submodule

At a first stage, the Detection Submodule makes use of computer vision techniques to detect the pixels to be protected. As these ROIs are detected, an XML file related to the current video is also written. This file contains ancillary data such as the number and location of the ROIs at each frame. The Detection Submodule must detect the ROIs in real time and accurately. Specifically, the following computer vision techniques offer high accuracy and can work in real time as explained in Chapter 2:

- When the ROIs to be detected are faces, the *Haar-Features* technique Viola and Jones (2001) is accurate and works in real-time, if it is run on fast hardware. Although the protection of a face is widely accepted for privacy protectionDufaux (2006), identification can be performed based on other factors. Hence, it must be stated that if a VSS considers the faces as ROIs, some identity disclosure could be done by merely analyzing clothes or via gait recognition processes. Hence, we encourage VSS developers to consider the full body as the ROI, in the case of people.

- For generic ROI detection (*i.e.* moving objects in the scene are considered ROIs) based on background subtraction, the *Codebook Construction* Kim et al. (2004) technique fulfills the properties. However, in this technique using a fixed camera is mandatory. In order to overcome this shortcoming, detection techniques based on optical flow Horn and Schunck (1981), such as *Farnebäck* Farneback (2000) might be used.

In our platform, we can use either a robust optical flow technique based on tensor voting Rashwan et al. (2013) or a background subtraction technique Kim et al. (2004) to detect the accurate moving objects as ROIs. Note that using motion detection based on optical flow techniques is essential in order to avoid the problems of using background subtraction techniques under certain conditions: for instance, moving leaves or even rain would be detected as ROIs in case of using background subtraction with outdoor camera. However, the implementation of motion detection based on optical flow is more time consuming than other techniques based on background subtraction (for more details see Chapter 2).

The process to detect the ROIs can be summarized as follows:

1. The original compressed video stream is obtained from the camera controller.
2. A set of frames is uncompressed and the ROIs are detected.
3. An ancillary data structure containing information of the ROIs is stored.

Note that since the video is not modified, there is no need to recompress the video in the detection process.

6.2.2 The protection submodule

In turn, the protection consists of modifying the video data so as to hide the identifying features that could permit the disclosure of identities. In Chapter 2, the state-of-art proposals for video protection were divided into two groups, depending on the video domain in which ROIs are protected: *Pixel domain.* and *Compressed domain.*.

- *Pixels domain.* If ROIs are protected in pixel domain, this will clearly affect the quality of the compressed image. For instance, a scrambling of pixels entails a set of high-frequency image blocks which will suffer a heavy information loss after compression and unprotection (*i.e.* unscrambling). Moreover, some proposals are based on hiding the ROIs using solids, such as squares or circles. In these cases, also the original version of the video must be stored because the utility requirement is not achieved.

- *Compression domain.* If ROIs are protected in compression domain (*i.e.* some parts of the compressed video stream data structure are encrypted) any unauthorized user (*i.e.* without decryption key) will obtain noise in the ROI pixel area. On the contrary, authorized users will be able to decrypt the structure and reconstruct the original ROI. The methods in compressed domain must aim at not increasing substantially the size of the compressed video once protected. It is also interesting that a protection method is suitable for a variety of compression video codecs (*e.g.*, MPEG-2, H.264, MJPEG, etc.). However, some of the state-of-art protection methods are restricted to H.264 video Peng et al. (2013).

The protection methods based on coefficient alteration Dufaux and Ebrahimi (2008) of the compressed unprotected video are suitable to implement a TP-VSS. A compressed video is a set of compressed frames, grouped in GOPs (*Group of Pictures*). Each GOP starts with an I-frame (*intra-coded*) and contains several

P-frames (*predicted*) and B-frames (*bi-predictive*). I-frames are stored and compressed entirely: the frame is divided into 8×8-pixel blocks which are applied a frequency transform (*e.g.* Discrete Cosine Transform). The obtained 8×8-coefficient blocks describe the pixel block in terms of texture and details. For each block, there is one DC and 63 AC coefficients. A quantization is applied to each block, *i.e.* each coefficient is divided by a number, aiming at reducing the number of discrete symbols but resulting in a lossy compression and, also, a set of zero coefficients. Finally, entropy encoding (for the non-zero coefficients) and run-length encoding (for the zero coefficients) are applied for a lossless compression of the block. The information needed to reconstruct the frame is stored in a specific and standardized data structure. In addition, P and B-frames are not stored entirely: they just consist of the changing blocks between frames in the GOP.

The unprotection of the video is simply done by applying the inverse method on the protected video and the alteration of the video does not affect substantially the compression properties of the video file. We use this technique in the Protection Submodule of our platform. For example, the following procedure might be performed Dufaux and Ebrahimi (2008):

1. Generate a seed for a pseudorandom number generator (PRNG). Encrypt the seed using a secret key.

2. Generate the *protection stream* PS, a pseudorandom bit sequence with length $B \cdot 63$, where B is the number of coefficient blocks belonging to ROIs in the compressed frames.

3. Flip the sign of the i-th AC coefficient if the $(b \cdot 63 + i)$-th bit of PS equals '1', where b is the number of coefficient block being protected.

If any user attempts to retrieve and play a protected video, it would obtain noise in the pixels belonging to ROIs. When unprotecting the video, if the same PS is generated, the AC coefficients whose sign was flipped in the protection operation would have their sign properly restored and, as a result, the pixel blocks of the ROIs would be correctly displayed.

Assuming large video lengths, if the PS was generated for each video, large sequences of bits would be needed. Moreover, even if the trusted manager only had to access (and thus unprotect) a small amount of frames, a whole large PS should be generated. Pseudorandom bit sequences may suffer from security problems due to extremely large lengths (as a reference, in conditional access systems for digital TV, the so-called *control word* key is typically renewed every two minutes). Hence, we propose generating a PS for each GOP in the video file. If the trusted manager has to access a certain set of frames, only the PSs belonging to their corresponding GOP must be generated.

Another aspect related to the length of PS is whether the videos to be protected are grayscale (*i.e.* only the luminance component is considered) or in colour (*i.e.* luminance and chrominances are considered). The specific length of PS will depend also on the chroma subsampling model used (*e.g.* 4:2:2, 4:2:0, etc.). In this chapter, we assume that videos are in color since this is common in state-of-art VSS.

6.2.3 The information system

In our platform, the Information System stores the protected videos and the information needed to unprotect the videos if necessary. Specifically, for each video file, the next information is stored:

- `VideoTime` The timestamp of the video. We assume that the length of each video file is equal (*e.g.* 10 minutes each).
- `VideoLocation` In our case, the Information System uses paths that aim at classifying the videos according to their date of archiving (*e.g.* /videos/2012/11/15/video023.mp4)
- `ROIdescription` It consists of an XML file containing information on the ROIs of the video file, for instance the list of ROIs in each frame and a ROI is defined by its bounding box. The Detection Submodule is in charge of writing this information.
- `VideoMAC` This is a set of message authentication codes (MAC) for the different fragments the video file is divided into, aiming at checking the video integrity. If MAC checking fails when a video is open, it can be replaced from a backup database.
- `VideoKey` This contains the information to protect and unprotect the video.

`VideoKey` is uniquely related to each video v stored in the Information System. It is chosen and computed by the Protection Submodule prior to start protecting a new video file. When a new video has to be protected, the Protection Submodule contacts the server of the Law Enforcer. We describe now how this value is computed:

1. Let r be a random number chosen by the Protection Submodule.
2. Let TM_{sk} and LE_{sk} be the secret keys of the Trusted Manager the Law Enforcer respectively.

3. The Protection Submodule contacts the Law Enforcer server using a secure channel (*e.g.* protected by Secure Socket Layer, that guarantees confidentiality and integrity) and sends r.

4. The Law Enforcer server sends $k_v = \mathcal{E}(r, LE_{sk})$, where $\mathcal{E}$ is a symmetric encryption function, to the Protection Submodule.

5. Now, `VideoKey` is stored as

$$\mathcal{E}(k_v||\mathcal{H}(k_v), TM_{sk}),$$

where $\mathcal{H}$ is a one-way hash function used for integrity checking of VideoKey.

Last but not least, the different `VideoMAC` values are computed as

$$\mathcal{E}(\mathcal{H}(video_f), TM_{sk}),$$

where $video_f$ corresponds to a fragment f of the protected video stream.

Note that the software in the Protection Submodule does not store the value r and, consequently, the participation of the Law Enforcer counterpart will be mandatory in case of video unprotection. Certainly, the original value r could be dishonestly used by the Trusted Manager but we assume this scenario is not possible: the Trusted Manager is indeed trusted.

6.3 Protection and unprotection of videos

In this section, we specifically address the procedures of protection and unprotection of a video. Moreover, we detail the construction of the protection stream and propose different variations of the coefficient alteration technique.

6.3.1 Construction of the protection stream

The PS is used for both protecting and unprotecting the GOPs in a video file. Hence, prior to altering the coefficients of the ROIs of a GOP, this bit stream must be generated. Similarly, prior to unprotecting the GOP, the same PS must be generated.

Pseudorandom sequences are generated using an initial value or *seed*. This seed depends on a random value, r, that is stored in the Information System in an encrypted manner. We now describe the whole process to generate a PS for a GOP g:

1. Extract from the file `ROIdescription` the list of ROIs to protect.

2. Let B be the number of coefficient blocks corresponding to ROIs in the GOP.

3. Let $K_g = \mathcal{H}(g||r)$ be the key of the block cipher function, for GOP g.

4. Let $I_g = \mathcal{H}(g||K_g)$ be the initial counter, for GOP g.

5. To obtain the PS, run the Advanced Encryption Standard (AES) block cipher in *counter mode* $\lceil B \cdot (N)/n \rceil$ times, where n is the output size of the block cipher and N is the number of bits needed to "protect" a block.

6.3.2 Protection procedure

The coefficient alteration method introduced in Section 6.2.2 suits the property of utility (*i.e.* the sign of AC coefficients will be correctly "restored" if the same pseudorandom sequence is generated for unprotection). In order to test the robustness of the protection method, we propose three different flavours of coefficient alteration:

- **AC sign flipping.** The sign of non-zero AC coefficients is flipped according to the PS.

- **DC encryption.** Only the DC coefficient is altered. To that end, we use the bits of the PS to encrypt, using the XOR function, the value of the DC coefficient. Note that the bitwise XOR operation is reversible.

- **DC encryption + AC sign flipping.** It consists of flipping the sign of the non-zero AC coefficients and encrypting the DC coefficient.

The value of N defined in Section 6.3.1 depends on the coefficient alteration scheme used: For AC sign flipping $N = 63$, for DC encryption $N = b_{DC}$ (thus, assuming that the DC coefficient is encoded using b_{DC} bits) and, for DC encryption + AC sign flipping $N = b_{DC} + 63$.

Note that other possible schemes have not been used: on the one hand, an AC encryption scheme would result in a large increase of the size of the compressed video; on the other, a DC sign flipping only results in unnoticeable image changes (cf. Figure 6.2). Moreover, protection must be done on all the image components, *i.e.* luminance and chrominance. Figure 6.4 shows the effect of applying the DC encryption + AC sign flipping only on the chrominance components of the frame. The resulting frame still allows the identification: certainly, the luminance components are the most relevant in the context of image perception.

Figure 6.2: Result of DC pseudorandom sign flipping.

6.3.3 Unprotection Procedure

Unprotection of a video can only take place upon specific events, such as criminal investigations. Hence, the trusted manager cannot unprotect a video on his/her own. To do so, the unprotection request to the platform entails the contact with the Law Enforcer server. The purpose is twofold: on the one hand, the value needed to decrypt the unprotection keys was encrypted by the Law Enforcer; on the other, the Law Enforcer might log all the requests received from different TP-VSS. In order to unprotect a GOP g of a video:

1. The Information System checks that the video fragment f to be unprotected has not been modified. The value $\mathcal{E}(\mathcal{H}(video_f), TM_{sk})$ is computed and compared with the `VideoMAC` value of the corresponding fragment. If this integrity checking fails, the fragment of the protected video file could be restored from a backup server.

2. The Information System reads the `VideoKey` value from the database and checks its integrity (using the $\mathcal{H}(k_v||TM_{sk})$ value stored in `VideoKey`). If integrity check fails, the value could be restored from a backup server.

3. Contact the Law Enforcer server using a secure channel and send the value k_v.

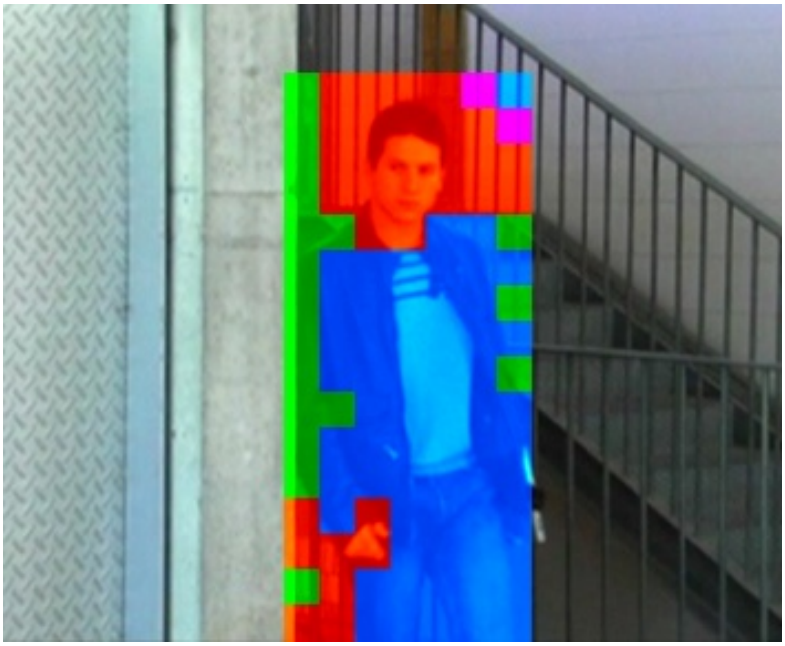


Figure 6.3: Result of DC encryption + AC sign flipping on the chrominance components.

4. The Law Enforcer server will decrypt k_v using the secret key LE_{sk}, *i.e.* $r = \mathcal{D}(k_v, LE_{sk})$, where $\mathcal{D}$ is a decryption function.

5. Construct the PS as described in Section 6.3.1, for the GOPs to be unprotected.

6. Using the information in the `ROIdescription` file, and the bits of PS, unprotect the video.

6.4 Implementation and discussion

The aim of this section is to discuss on the features of the platform and the techniques used in it. To illustrate its behavior, we have developed a testbed prototype. Our TP-VSS consists of a webserver that manages a front-end for the complete system. Both Detection and Protection submodules are programmed in C language and can be executed upon web-activaded requests as shown in Figure 6.4. The Information System resides in a plain MySQL server. All the software is installed in an Intel NUC computer.

We have divided this section into two parts: in the first one we address the degree of protection offered by the image transformations for the three variations

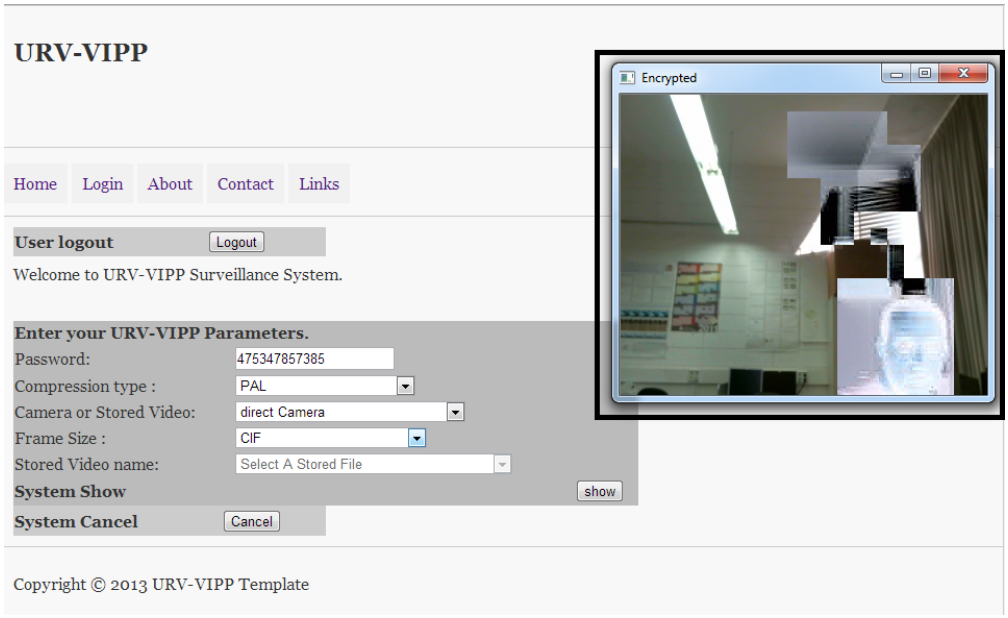


Figure 6.4: A snapshot of our implemented prototype.

of coefficient alteration, elaborating on both grayscale and color videos. In the second part, we address the security of the information involved in the platform.

6.4.1 Effectiveness of the protection techniques

The primary goal of the protection technique is to prevent that identities could be disclosed by merely observing protected frames. However, some other aspects should also be considered, such as robustness against attacks.

Identity concealment

Firstly, we evaluate the effectiveness of the three coefficient alteration techniques with respect to their capability of concealing the identity of individuals. To that end, we proceed to alter the coefficients of four different frames: BoyLift, GirlDoor (both images were obtained by the authors of the article), 1-Corridor, 2-Corridor (both images were taken from the CAVIAR database [1]. These frames are shown in Figure 6.5.

We have applied the three protection methods to the example frames. The results can be observed in different figures: Figure 6.6 shows the protection with the AC sign flipping method, Figure 6.7 shows the protection with the DC encryption

[1]INRIA Labs, *CAVIAR Test Case Scenarios.* http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

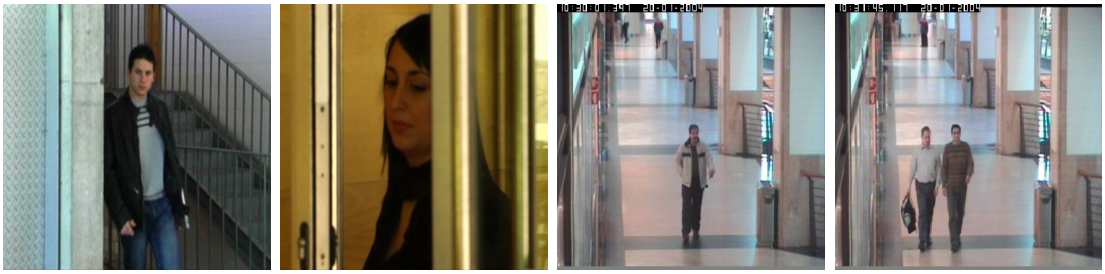


Figure 6.5: The four frames used to evaluate the protection method: BoyLift (top, left), GirlDoor (top, right), 1-Corridor (bottom, left) and 2-Corridor (bottom, right).

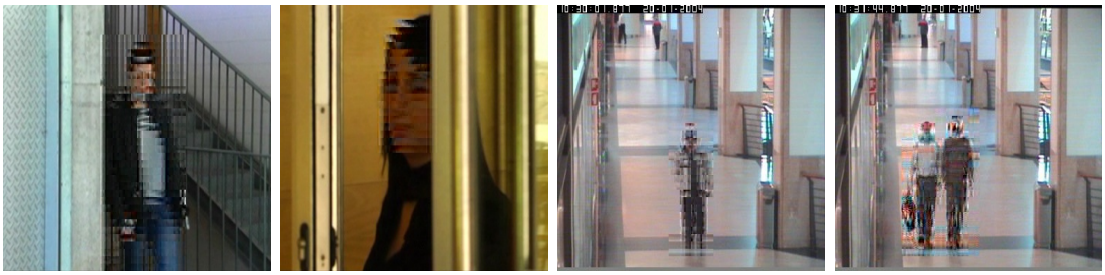


Figure 6.6: The four frames protected with the AC sign flipping technique.

method and finally Figure 6.8 shows the protection with the DC encryption + AC sign flipping method.

We can observe that AC sign flipping might not be enough to protect the privacy of individuals. On the contrary, methods including the encryption of the DC coefficients offer higher privacy protection. However, keeping the AC signs unaltered results in some of the details of the original image being still noticeable. Note that combining both AC sign flipping and DC encryption makes unfesaible disclosing the identity of an individual by visual inspection.

In order to quantitatively compare the three coefficient alteration methods with regards to privacy protection, we use a variation of the Mean Square Error (MSE) between the original frame and the protected frame. A high MSE value indicates a high information loss in the protected frame (and hence high privay protection). Specifically, we compute the MSE only for the pixels that belong to a ROI. This measure indicates the error per pixel due to the protection of the ROIs and allows comparing the effect of a protection technique between different images. Table 6.1 shows the values for the coefficient alteration schemes presented.

Random alteration attacks

In Martínez-Ballesté and Rashwan (2013), we introduced the concept of Random Alteration Attack. A natural way of dishonestly unprotecting a GOP is to generate

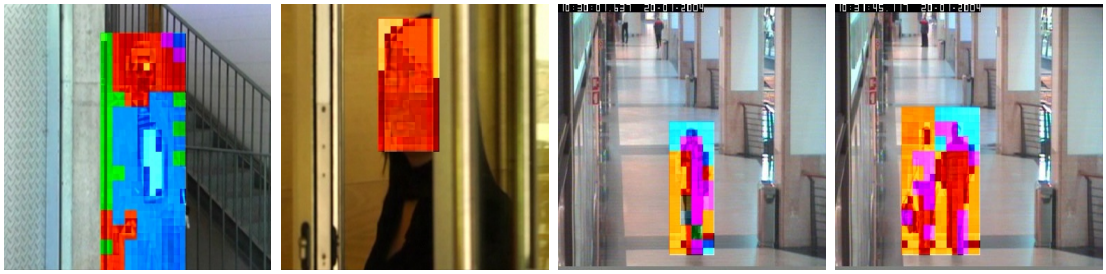


Figure 6.7: The four frames protected with the DC encryption technique.

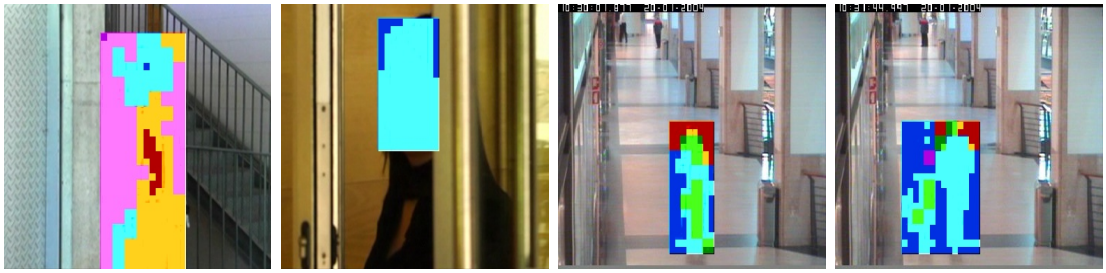


Figure 6.8: The four frames protected with the DC encryption + AC sign flipping technique.

Scheme	BoyLift	GirlDoor	1-Corridor	2-Corridor
AC sf	415	337	125	296
DC enc	3479	697	1230	1943
AC sf + DC enc	9790	3659	2430	3798

Table 6.1: Mean Square Error taking into account the pixels belonging to ROI (sf: sign flipping, enc: encryption).

a valid PS. However, the Random Alteration Attack does not consist of attacking the cryptographic information, but of altering the visual elements of the protected object (*i.e.* the pixels) so as to obtain an image that, far from being the exact unprotected frame, allows the identification of the individuals. Hence, if attackers know the method utilized to protect the frames they can randomly change the values of the bits of the DC coefficients and/or change the sign of the AC coefficients until the person in the ROI becomes identifiable.

In order to assess which of the variations is more effective, we have tested the robustness of the three coefficient alteration techniques against this attack. To that end, we have generated 1000 random variations of a protected frame, and have selected the one with less MSE (hence, the random unprotected image that is "nearest" to the original image). The results are shown in Figure 6.9, together with the original frame. We can observe that, although the schemes using DC encryption were offering high protection, the robustest method against this attack

Figure 6.9: The best frames unprotected by the Random Alteration Attack, for the different coefficient alteration schemes studied (top, left: original frame; top, right: AC sign flipping; bottom, left: DC encryption, bottom, right: DC encryption + AC sign flipping).

is the DC encryption + AC sign flipping (that is, in the 1000 random frames, the frame nearest to the original one does not allow identification). As a conclusion, we select the DC encryption + AC sign flipping method as the protection technique used in our platform.

Other aspects

There are some other minor aspects that are worth to be addressed here. An advantage of our proposal is related to the effect of protection on video compression: the coefficient variation techniques proposed here do not affect significantly the compression ratio of the protected video. Our tests show that protection increases up to 1% the size of the compressed video stream. Note that, for the sake of brevity and, due to the lack of relevance, we do not include the results of these tests.

However, the reversibility that provides the coefficient alteration methods is not guaranteed if some transformations are done to the video (*e.g.*, scaling, changing the aspect ratio, etc.). Although this issue seems promising, studying the effect of these transformations on the reversibility property is out of the scope of the chapter.

6.4.2 Time performance

As mentioned in the Introduction, all the processes involved in the Detection and Protection submodules must work in real time aiming at avoiding the temporary storage of original video. Certainly, the internal components of the system make use of temporary buffers as a support of the software processes. Notwithstanding, we assume that at least the VSS does not write temporary data in its filesystem. Moreover, the number of frames per second should also be considered. In this sense, the number of frames per second is in general lower (*e.g.* 15 fps), for the sake of processing in real time.

Three important elements [2] must be considered when addressing the time performance:

- The time spent for ROI detection.
- The time spent for PS creation.
- The time spent for ROI protection.

Table 6.2 shows a summary of the processing times related to the above elements. We have measured the average times in 1200 frames captured by means of our prototype. There are results for CIF and PAL resolutions. We have used MD5 (Message-Digest Algorithm 5) as hash function and 128-bit AES as encryption function.

Note that the most time consuming procedure is the one related to ROI detection. For PAL frames and optical flow detection, our implementation allows the process of 11 frames per second.

Regarding unprotection, the whole procedure is naturally faster since, in this case, the ROI detection procedure is not executed. Moreover the ROI unprotection and PS creation will consume the same seconds than in the protection procedure.

Step	**CIF** (352 × 288 px)	**PAL** (720 × 576 px)
Detection (BS)	10.35	18.20
Detection (OF)	58.23	88.72
PS creation	0.37/12	0.56/12
Protection	0.25	0.38
Time per frame (BS)	10.63 (94.3 fps)	18.61 (53.7 fps)
Time per frame (OF)	58.51 (17.2 fps)	89.13 (11.2 fps)

Table 6.2: Time (in milliseconds) for the processes in our platform prototype. A GOP consists of 12 frames. We present the results for background subtraction (BS, Kim et al. (2004)) and for optical flow (OF, Farneback (2000)).

6.4.3 Security overview

Besides providing a real-time and reversible privacy preservation, the system must accomplish some security requirements. On the one hand, disclosing the identity of the persons of the video should not be straightforward for the attackers. On the other hand, the cryptography functions utilized in our platform must be evaluated

[2] Note that video compression takes place in the camera hardware and, hence, is not considered here.

in an appropriate manner. To introduce this security analysis, we present some assumptions:

1. The goal of attackers is to unprotect the ROIs of a protected video.
2. Both the original video and PSs are never stored in the system.
3. Any public user (including potential attackers) can only access to the protected video files and the XML description files.

The security aspects related to information security are tackled through the analysis and discussion of some specific claims.

Claim 1. *Attackers cannot feasibly generate the PS unless they gain access to* r. The PS is generated using a block cipher in counter mode. Both the encryption key and the initial counter are generated from the value r. These elements are generated using secure cryptographic functions. To dishonestly obtain r, attackers might gain access to the Information System and use the $VideoKey$ value in a brute search attack. The time consumed for succeeding with the attack is upper bounded by the bit length of r. Last but not least, attackers could use attacks such as the man-in-the-middle to eavesdrop r from the secure communication channel between the Trusted Manager and the Law Enforcer.

Claim 2. *Attackers can dishonestly modify a protected video and/or its* $VideoKey$ *value, but these attacks can be detected.* Attackers could attack the system aiming at modifying the videos in the Information System. These modified videos can produce incorrect renderings after unprotection. Since a MAC is stored for each video (in fact, for video fragments), the Information System can check if the video has suffered any modification. Also, the use of integrity checking aims at detecting dishonest modifications of the $VideoKey$ value. To mitigate the effects of such attacks, some backup policies are implemented. However, if an attacker could infer the secret key of the TM, both new $VideoMAC$ and $VideoKey$ values could be generated.

Claim 3. *The probability of an attacker unprotecting a ROI in a GOP is negligible.* If attackers had access to the original (unprotected) video, they could compare the signs of the non-zero coefficients[3] in the frames to attack. Upon differences in the coefficients between videos, attackers could build the PS for that specific GOP. However, given the assumptions, this way of obtaining the PS is not feasible. Hence, attackers can only perform the aforementioned Random Bit Alteration Attack and visual inspection of the resulting frames. This is certainly unfeasible for the DC encryption + AC sign flipping protection. Finally, note that

[3]The information to compare depends on which of the four coefficient alteration schemes has been applied to protect the video.

building the exact PS by brute search is computationally infeasible. For instance, for the robustest scheme, finding the right combination of alterations for a unique block is upper bounded by 2^{d+63-z} (where d is the bit-length of the DC coefficient, and z is the number of zero AC coefficients for the block).

To conclude, the selection of secure block ciphers and hash functions is necessary to fulfill the security requirements. However, as in many scenarios, the linchpin of the security of the system is the correct management of the secret keys involved in the system.

Chapter 7

Robustness of the Coefficient Alteration Protection Method

In this chapter, we propose a face reconstruction algorithm for protected faces. These faces were protected by altering AC and DC coefficients of the blocks corresponding to a face region in a video compression. The proposed unprotection algorithm exclusively depends on video processing techniques instead of disclosing the unprotection key used in the protection process. This approach consists of the following stages. Firstly, random unprotected faces are generated based on a random alteration of AC coefficients with a fixed value of DC coefficients. Secondly, the best unprotected faces are selected by an Eigenfaces model trained with facial images from a repository of potentially protected persons. In addition, a single facial image is generated by merging the best resulting images through median stacking. Finally, the Eigenfaces model is utilized to recognize the face from the repository, which is the closest to the resulting image in order to improve the aspect of the unprotected face. Experimental results based on both a proprietary database and a public CALTEC faces database show that face reconstruction using the proposed approach is very effective in order to break the protection applied to faces.

The structure of the chapter is as follows. A summary of the method used for protecting videos and the scenario of the attacking are introduced in Section 7.1. In Section 7.2, a method to unprotect a protected facial image is described. In Section 7.3, several tests are conducted in order to evaluate the efficiency of the attack.

7.1 Introduction

Video surveillance systems (VSS) are recording people while doing their daily activities, since this aspect entails privacy issues, legislations regulate the management of video surveillance data. However, they are focused on the behavior of VSS managers and operators. In Chapter 6, we stated that legislations should go one step forward and concentrate on the adoption of Privacy Enhancing Technologies (PET) applied to video surveillance in order to prevent the reidentification of individuals.

In Chapter 6, we described the properties that a privacy-aware VSS must have in order to move trust from the VSS operators to the VSS themselves. In a nutshell, in order to avoid the need for human supervision (and hence, to avoid a privacy issue) the detection module must work accurately (*i.e.*, all the ROIs must be detected); In addition, in order to avoid the storage of the original (and unprotected) video, the protection process must be completely reversible (*i.e.*, in case of being accessed by a law enforcer, the video must be easily reverted to the original one without any loss of information). Finally, all the processes involved must work in real time so as to avoid the temporary storage of video sequences. Note that storing the original video (both temporarily or permanently) paves the way for the VSS becoming a focus of attacks aimed at leaking the original videos.

In addition, in Martínez-Ballesté and Rashwan (2013), we described the design and implementation of a database system for a privacy-aware VSS that makes use of these techniques, and relies the security on a secret key owned by the trusted operator of the VSS. The ROIs protection system is constituted by the following stages applied to a given sequence of the raw video:

1. Detect the ROIs in the sequence and write an ancillary data file with the information of the ROIs in the sequence.

2. Compress the raw video into a set of MPEG group of pictures, or GOP[1].

3. For each GOP, generate a seed for a pseudo-random number generator (PRNG) using the GOP number in the sequence and some other random values. Protect the seed using the secret key of the trusted operator.

4. Protect each GOP as follows:

[1]In the compressed video stream, Each GOP starts with an I-frame (*intra-coded*) and contains several P-frames (*predicted*) and B-frames (*bi-predictive*). I-frames are stored and compressed entirely: the frame is divided into 8×8-pixel blocks which are applied a frequency transform (*e.g.* Discrete Cosine Transform). The obtained 8×8-coefficient blocks describe the pixel block in terms of texture and details. For each block, there is one (direct) DC (a coefficient with zero frequency) and 63 (alternate) AC coefficients (coefficients with non-zero frequencies)

- Generate the *protection stream* PS, a pseudo-random bit sequence of length $l = B \times (b_{DC} + 63)$, where B is the number of coefficient blocks belonging to ROIs in the compressed GOP, and b_{DC} is the number of bits for encoding the DC component of a block.
- Protect each coefficient block b by XORing, *i.e.* encrypting, the i-th bit of the DC coefficient with the $(64 \cdot b + i)$-th bit of PS and flipping the sign of the j-th AC coefficient if the $(64 \cdot b + j)$-th bit of PS equals one, where b is the number of the coefficient block being protected.

7.1.1 Attacks on protected frames

Certainly, we showed in Martínez-Ballesté and Rashwan (2013) that the described system is trustworthy and, moreover, we proved that all the data involved is stored in a secure manner. We assumed that attackers cannot unprotect the videos if they cannot access the secret key of the trusted operators. However, we sketched that the goal of an attacker might not be to retrieve or disclose the seed of the PRNG but to unprotect a protected frame by simply randomly altering the values of the pixels. If attackers had previous knowledge on the method utilized to protect the ROIs, they could concentrate on randomly XORing and flipping the sign of coefficients of the blocks belonging to ROIs.

In Martínez-Ballesté and Rashwan (2013), we proposed four variations of the coefficient alteration method Dufaux and Ebrahimi (2008): DC encryption, AC sign flipping, DC + AC sign flipping, and DC encryption + AC sign flipping. We demonstrated that the latter combination (the one specifically described in the previous section) is the most robust against these kind of attacks.

Cryptographic attacks aim at sabotaging the security of cryptographic algorithms, and they attempt to decrypt encrypted data or a part of them without a prior-knowledge of the secret key, which is an important part of cryptanalysis. The attempt of hacking on an encrypted image, video or data depends on knowledge of the encryption methods used, and can be perpetrated in two different ways:

- If attackers only have the encrypted data without any knowledge about the original data and the secret key, they should try to generate a number of possibilities to estimate the secret key used for decrypting the data. With long keys, this process is very time consuming and, in general, computationally unfeasible. In Martínez-Ballesté and Rashwan (2013), we proved that this method is unfeasible for the coefficient alteration methods, especially for those involving AC sign flipping.
- If attackers have access to some information about the original/unprotected data, they can match the available data to the decrypted data to efficiently reconstruct the original data.

In this chapter, we describe an attack against the protection system described in Martínez-Ballesté and Rashwan (2013). Instead of trying to disclose or obtain the unprotected data that is used to generate the key of the PRNG, we exclusively apply video processing technology.

Specifically, we assume the following scenario: A company has a VSS to observe people who enter the company buildings. The VSS used preserves the privacy of individuals by protecting their faces by means of using the technique proposed in Martínez-Ballesté and Rashwan (2013). These videos can only be decrypted by the trusted operator and under legal authorization.

We assume that there is an attacker who can break and penetrate the security of the database of the VSS of this company, thus obtaining the protected video surveillance video. Furthermore, the attacker has access to a public database of facial images of the employees (such databases are quite common in public web pages of companies and institutions). This attacker aims at reconstructing the protected facial images within a frame. The attacker does not aim at reconstructing a perfectly unprotected frame, but a facial image that allows the recognition of the employees (*optical decryption* Li et al. (2008)).

7.2 Attack to unprotect a facial image

In this section, we address the method to unprotect the ROI in a protected video frame. We assume that the attacker has previous knowledge about the faces that might be protected (*i.e.* a public database of the pictures of the employees has been accessed). In addition, we assume that the attacker is aware that faces have been protected using the coefficient alteration technique proposed in Martínez-Ballesté and Rashwan (2013).

As shown in Figure 7.1, the attack consists of four steps: (i) generating random unprotected faces; (ii) selecting the best unprotected faces; (iii) merging the best images and (iv) improving the aspect of the unprotected image.

7.2.1 Generating random unprotected faces

The coefficient alteration methods perform the protection in the compressed domain of the video, not in the pixel domain. In particular, they change the DC and AC coefficients of the compressed frame blocks. A DC coefficient encodes the average intensity value of the pixel block. It often consumes more bits than the AC coefficients, which represent the detailed views of the encoded video sequences. For the sake of compression, coefficients are quantized. As a result, some of the AC coefficients become zero.

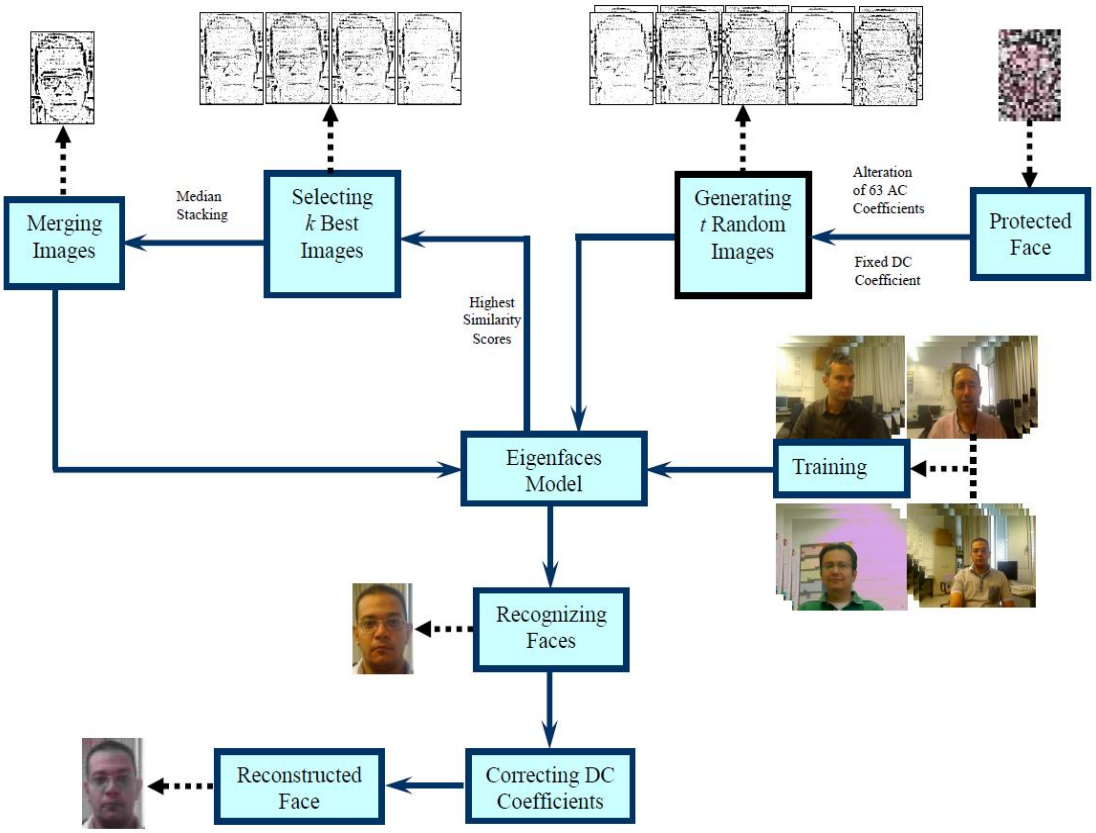


Figure 7.1: Overview of the proposed algorithm for attacking a protected face.

In this first step and, in order to reduce the search space, we set the luminance DC coefficient of each block to a fixed value between 0 and 255. Then, we obtain t random candidate unprotected images by randomly flipping the sign of the non-zero AC coefficients.

7.2.2 Selecting the best decrypted faces

In the second step, the attacker uses the well-known *eigenfaces algorithm* Turk and Pentland (1991) in order to select the k best images corresponding to the t unprotected candidate images.

Faces can be represented by a vector that consists of the image rows concatenated with each other. Eigenfaces algorithm yields a new space that can describe faces in more discriminant way than in the original image space. The base vectors of this space are the eigenvectors.

The aforementioned eigenvectors constitute as a set of features that character-

ize the variations between different face images. Each face image can be represented by one or more eigenvectors corresponding to the largest eigenvalues. Those eigenvectors that are related to as eigenfaces are used for the classification stage.

A very important aspect is the difference of sizes between the protected faces and the faces in the database. In order to gain scale invariance, every face is centered and scaled such that its height and width are set to a predefined size $(H \times W)$.

The Eigenfaces procedure is summarized in the following steps:

1. Construct an initial set of images that are the original facial images obtained from the public picture database as a training set, $(H \times W \times c)$, where $H = height$, $W = width$, and c is the number of the original face images. However, The original images are modified by setting the face luminance DC coefficients to a fixed value between 0 and 255.

2. Compute the eigenvectors from the training set, only keeping the vectors that correspond to the M largest eigenvalues.

3. Finally, calculate the corresponding distribution in the M-dimensional weight space for each known individual, by projecting their face images onto the *face space* calculated in the previous step.

The following steps are used to select the k best images from the candidate unprotected images:

1. Compute the set of weights, using an input image and the M eigenvectors.

2. Determine if the face is sufficiently close to the face space or not by calculating a similarity score.

3. If it is a face, classify the weight pattern as either a known or an unknown person.

4. Repeat the three previous steps for the t candidate unprotected images.

5. Finally, select the k images that correspond to the largest similarity scores.

7.2.3 Merging the best faces through median stacking

Each image of k selected images is an unclear facial image (ghost face) for a same person. In addition, these images are noisy images and show unclear details of face elements (eyes, nose, etc.). Therefore, in the third step, the attacker merges the k extracted images into a single image in order to obtain an image that contains the

most of face elements. The simplest method to merge the k images is by averaging the images. This is a widely used method for noise reduction although it causes image blurring.

Therefore, in order to preserve the image details and reduce noise effect, the *median stack* procedure Gabbouj et al. (1992) is used so as to increase the signal-to-noise ratio in images. Median stacking considers all the values of a pixel at a single location across each image in the stack, and then chooses the final value for that pixel based on the median result.

Now, the attacker has a single image I that contains details of a protected face. In order to obtain a noise-robust image, a 3×3 median filter is applied to the resulting single image I. In addition, the Eigenfaces algorithm is used for recognizing the closest faces in the database. This is done by computing the similarity score between the *unknown unprotected face* and each face in the original database of facial images. We select the k_s faces corresponding to the k_s highest similarity scores, if they are related to the face of the same person (the original database may contain different faces of the same person with different poses and scales).

7.2.4 Improving the quality of the reconstructed face

As indicated above, all DC coefficients have been set to a fixed value, which leads to an uncorrect intensity value for the image pixels. In this last step, the correct values for DC coefficients are estimated, in order to obtain a correct color value for each pixel of the reconstructed face.

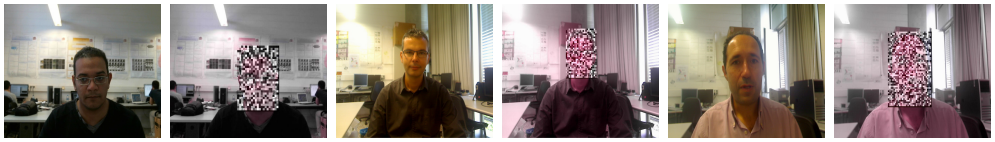
Firstly, the attacker uses the database of the original faces to estimate correct values of DC coefficients. Each recognized face is divided into five general regions: forehead, eye, nose, mouth and jaw. Each region is then divided into 8×8 blocks with an overlap, namely, by sliding dividing-partition one pixel by one pixel. Then, the pixels in each image block of each region of the recognized face(s) are transformed using the DCT transformation. In this step, the attacker constructs a dataset containing DC coefficient values for each block in a region of each recognized face.

Therefore, in this algorithm, that attacker has five matrices containing DC coefficients, each is related to one of the five regions. The number of columns per region equals the number blocks. In turn, the number of rows equals the number of the recognized faces (k_s). Additionally, if k_s is more than one, the median value of rows of each matrix is used as the estimated DC coefficient value for each block in a region.

7.3 Experimental results

We have developed a small database of facial images. The images were collected with different scales and face rotations with an uncontrolled indoor environment using a webcam (Logitech QuickCam Orbit/Sphere AF). The database contains 50 static images for eight unique people. The attack algorithm has been implemented in Matlab.

Assume that the attacker has a protected frame (see Figure 7.2 for two examples of original and protected faces). In the first step, $t = 1,000$ random images. Figure 7.3 shows some of these t random candidate images. In addition, DC luminance coefficients of ROIs blocks are set to 255. As shown in Figure 7.3, the rough details of faces are not correct, and the images are ghostly faces. However they contain some face elements, such as the eyes, nose, edges, boundaries, etc.



(a) (b) (c) (d) (e) (f)

Figure 7.2: Original images and their corresponding protected versions.

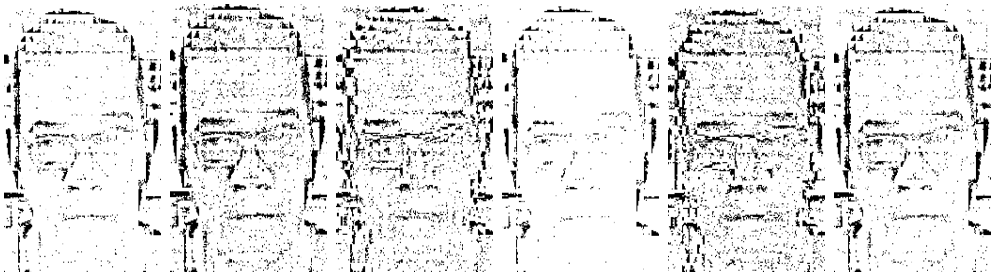


Figure 7.3: Some of the t random candidate images (only the luminance component is shown).

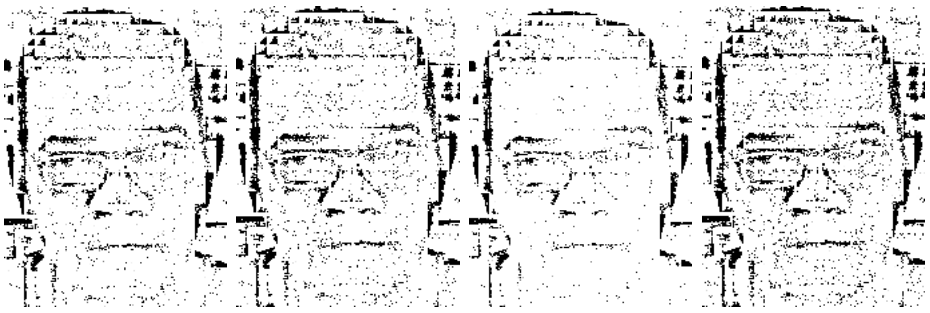
The training images (the faces obtained from our database) are modified with a luminance fixed DC value of 255 as shown in Figure 7.4. Note that the face belonging to the attacked protected frame is not in the training database. Now, a number of the original images with fixed DC values are used to build an Eigenfaces model for a face detection.

After applying Eigenfaces on the t random images, each image is projected on the Eigenfaces subspace in order to determine if a face or not a face by checking the

(a) (b) (c) (d) (e) (f)

Figure 7.4: (a and c) Original facial images and (b and d) Original facial images with a fixed DC value (DC = 255) used for training an Eigenfaces model.

similarity score (S) between the input image and the resulting model. The $k = 20$ images with the highest similarity scores yielded by the Eigenfaces algorithm are selected from t images. Examples of the k best selected images for a protected person are shown in Figure 7.5.



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

Figure 7.5: The four best images corresponding to the highest similarity scores (only luminance component is shown).

By applying median stacking, a single image I is generated, which has more information about details and face edges, but still fails in intensity values. Figure 7.6 shows three different examples for the single median image for three different facial images from our private database.

The Eigenfaces algorithm is again used for recognizing the closest face to the resulting single face. We evaluated the accuracy of the Eigenface model by training it with different facial images of the same person. Table 7.1 shows the accuracy of the Eigenface model with the number of facial images of the person used for constructing the model. As shown, the use of different facial images with different views of a same person yields an increase of the correct detection rate of the Eigenfaces model. For instance, the results show that if the Eigenfaces model is trained with one facial image per person, the classification rate is more than 20%.

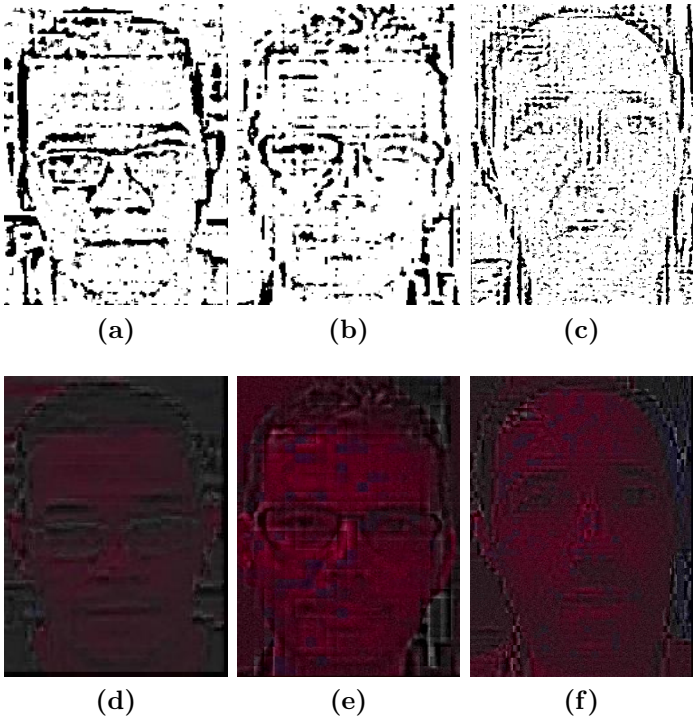


Figure 7.6: (a) Three median luminance component images for three different persons; (b) Three median facial images after adding chrominance components.

In turn, using more facial images per person yields higher classification rates. Notwithstanding, the percentage of training with one image per person is only 20% (low classification rate), however it is considered a good rate comparing with completely random unprotection algorithms. Actually, the completely random unprotection is unfeasible in this case, since 2^{ν} tests are necessary, where ν is the number of blocks times the number of non-zero AC coefficients, and finally the attacker may obtain a face that is a complectly different from the original one. Furthermore, in order to increase the chances of getting a correct face detection in this algorithm, the attacker can select the three maximum similarity scores of the recognized facial images.

The attacker uses the recognized faces to correct DC coefficients for the resulting image I, as shown in Figure 7.7. The fixed DC coefficients are replaced by the DC coefficients of the recognized faces in order to reconstruct the unprotected face, as shown in Figure 7.7. The attacker can use the first three detected faces corresponding to the highest three similarity scores. If, they are for the same person, median stacking is applied to get estimated DC values for each region within the face. However, if they are for different faces, each is individually used for

Number of facial images per person	Total number of facial images	Detection rate percentage
01	08	23%
02	16	38%
03	24	49%
04	32	62%

Table 7.1: Rate of correct face detection vs. number of facial images for a unique person used for training an Eigenfaces model.

correcting the DC coefficients.

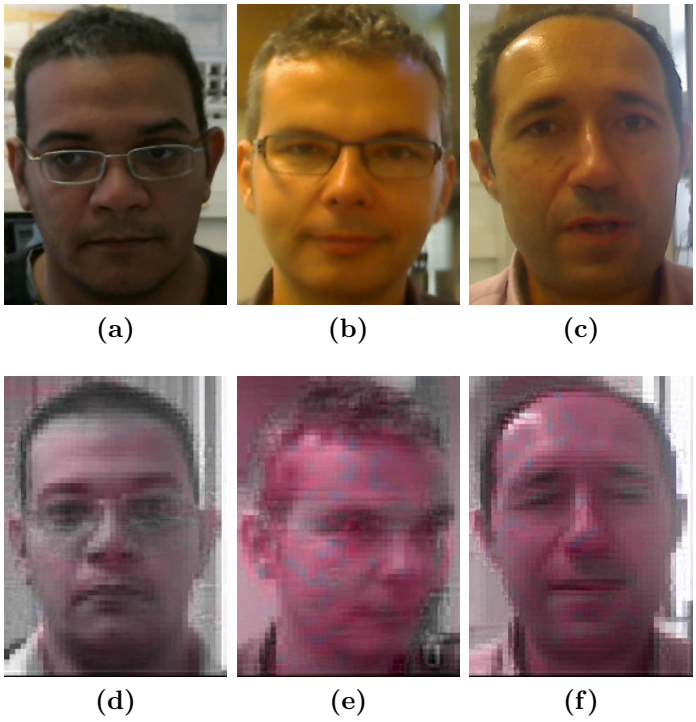
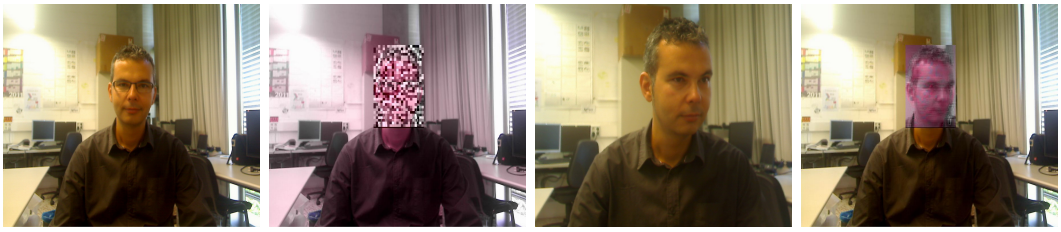


Figure 7.7: (a-b) Original facial images, (e-h) Reconstructed facial images.

The proposed algorithm is not affected by scaling, as a result of using a fixed template for training and testing the Eigenfaces model. Figure 7.8 shows that the size of the recognized face used for correcting DC coefficients is different from the size of the input protected face. However, there is no influence on the algorithm accuracy.

The Matlab execution time of the whole algorithm is around $1,300$ seconds on



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

Figure 7.8: (a) Original image without protection, (b) protected image, (c) original image contains a recognized face used for correcting DC coefficients, and (d) reconstructed image with the resulting unprotected face.

an Intel Dual Core at 3.2 GHZ, including the Eigenfaces model training and by considering 250×170 color facial images.

Additionally, in order to get a more realistic algorithm, the proposed algorithm has been applied to the public CALTECH face database [2], which contains 450 color frontal images (896×592) of 25 unique people. Each person has 20 views in different scales and under different illumination conditions.

We applied the proposed algorithm to different images for the same people. Qualitative results are shown in Figure 7.9. The proposed system yields up to 90% correctly detected faces. The impact on the algorithm accuracy of the number of facial images per person for the Eigenfaces model is kept with the public database. As shown in table 7.2, the larger number of facial images, the more accurate the obtained results. In addition, the results show if the Eigenfaces model is trained with 10 or 15 facial images per person, it leads to more than 90% of detection rates.

Number of facial images per person	Total number of facial images	Detection rate percentage
01	25	29%
02	50	42%
05	150	67%
10	250	92%
15	375	94%

Table 7.2: Rate of correct face detection vs. number of facial images for a unique person used for training an Eigenfaces model using the CALTECH face database.

[2]Caltech Frontal Face Dataset. Online: http://www.vision.caltech.edu/html-files/archive.html.

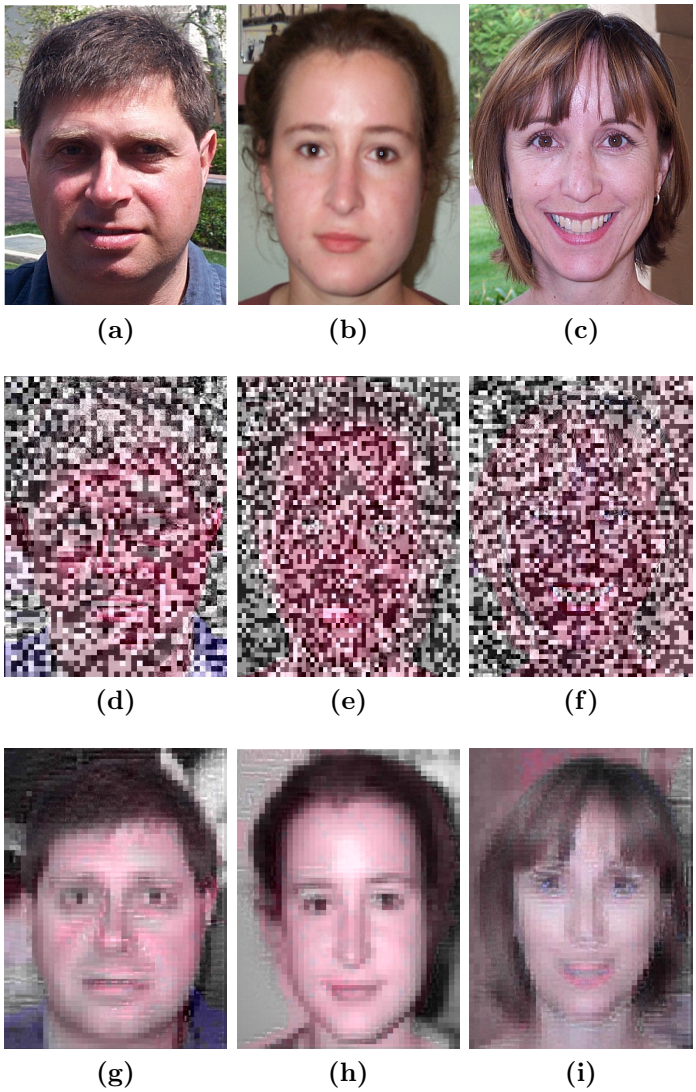


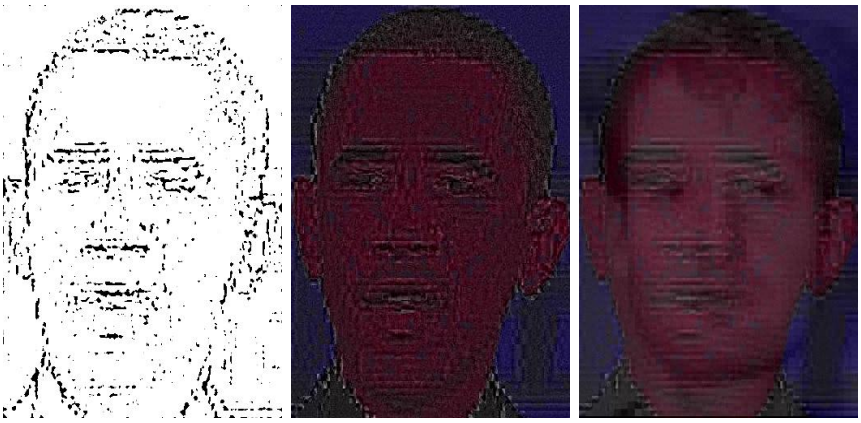
Figure 7.9: (column 1) Original facial images from the public CALTECH face databas, (column 2) Protected facial images with Martínez-Ballesté and Rashwan (2013), (column 3) Reconstructed facial images.

Furthermore, the algorithm has been applied in order to unprotect a face which does not belong to the database used for training the Eigenfaces model. Figure 7.10 shows that the reconstructed image contains the closest face recognized by the Eigenfaces model to the protected face. Actually, the reconstructed face is a fake deformed face that contains of the AC coefficients of the protected face (Figure 7.11(b)), as well as the DC coefficients of the recognized face (Figure 7.11(c)).

In order to diminish the impact of correcting DC coefficients using a wrong face, the similarity score of the recognized face is used to update the DC coefficients values. This means the final DC coefficients depends on the DC coefficients values of the recognized face and the similarity score resulted from a Eigenfaces model for this face, as: $(DC_{reconstructed} = score \times DC_{recognized})$. For instance, if the similarity score is very small, the colors of face regions are dark and this is a signal to a fake face as shown in Figure 7.11. In turn, the high score yields that the DC coefficients values of the reconstructed face are close to the DC values of the recognized face and this is a signal to a true face.

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

Figure 7.10: (a) Original image without protection, (b) protected image, (c) original image contains a recognized face used for correcting DC coefficients, and (d) reconstructed image with the resulting unprotected face.



(a) **(b)** **(c)**

Figure 7.11: (a) Resulting facial image (luminance component) with fixed values for DC coefficients, (b) resulting facial image (chrominance + luminance components) with fixed values for DC coefficients, and (c) reconstructed fake facial image contains a known recognized face used for correcting DC coefficients.

Chapter 8

Summary and Conclusions

"Imagination is more important than knowledge."

- Albert Einstein

In this thesis, the trust in video surveillance systems has been analyzed from the perspective of the trust in technology: ROIs detection and ROIs protection. From the theoretical standpoint, the concepts introduced and discussed are of concern for the computer vision community due to the increasingly widespread use of video surveillance systems. In general, an accurate understanding and control of the trust in video surveillance systems preserving privacy plays a key role in that people can do their daily activities under different cameras with full comfort and without tension. From the practical perspective, the concepts introduced in this thesis have been exploited for increasing the widespread of the presented algorithms of optical flow estimation as motion detection techniques. The techniques introduced in this thesis provide accurate noise-robust and illumination-robust flow fields that can intensively be used in a variety of computer vision applications.

In addition, this dissertation introduced a trustworthy model for a privacy-preserving video surveillance system, which ensures real-time performance, high accuracy, reversibility and information security by using a protection technique based on a coefficient alteration scheme. The coefficient alteration algorithms assure the reversibility and do not significantly increase the length of the video protection streams.

This final chapter presents a summary of the contributions and final remarks of this thesis and suggests future research directions.

8.1 Summary of contributions

Regarding the detection stage, this thesis has introduced a new stage exclusively based on tensor voting to build an optical flow estimation model to be more robust than the current state-of-the-art against the noise and outliers, as well as to preserve the discontinuities of the estimated flow field. In addition, in this thesis, an improved stage based on stick tensor voting is used in order to reduce computational cost of the proposed optical flow model based on a full tensor voting. In turn, most optical flow approaches can not cope with illumination changes over video captured. Thus, to overcome this problem, an illumination-robust optical flow algorithm was developed in this thesis based on the popular histogram of gradients (HOG) technique.

In turn, a robust protection scheme for the detected regions was developed based on a DCT coefficients alteration scheme using a secure protection stream during the decoding process. This method is reversible in order to retrieve the original data in a case of need under legal authorization, in addition does not affect on the compression ratio and the length of the video protection streams. Furthermore, a random attack algorithm was proposed to hack the protection algorithm to serve for improving the cryptographic algorithm used in the protection stage.

8.1.1 ROIs detection and protection approaches

The contribution introduced in Chapter 2 is twofold: first, we have described, analyzed and tested well-known methods and techniques involved in the main steps of a video surveillance system. In this regard, we have analyzed the techniques in the literature focusing on the properties that a trustworthy video surveillance system must fulfil (*i.e.*, real time performance, high accuracy and utility).

8.1.2 Using Tensor voting for estimating accurate flow fields

In Chapter 3, we have proposed an adaptation of the variational optical flow techniques described in Zimmer et al. (2009) and in in Bruhn et al. (2005) by replacing the Gaussian filtering applied in the form of structure tensors by a discontinuity-preserving filtering stage based on tensor voting. The application of tensor voting requires a pre-segmentation of the input images into three regions (homogeneous-moving, textured-moving, stationary) based on their spatio-temporal gradients. Tensor voting is separately applied to the homogeneous-moving and textured-moving regions. The proposed technique has been tested on a wide variety of real image sequences and compared with classical and state-of-the-art differential optical flow methods. Experimental results show that the proposed technique yields

flow fields with lower quantitative errors than previous techniques, and is able to better estimate the optical flow fields, especially over homogeneous regions, and better discriminate the boundaries of moving objects.

8.1.3 Optical flow fields estimation through a stick tensor voting

Chapter 4 presents a new approach for optical flow that can be used for detecting motion in a video surveillance system. This chapter has proposed an adaptation of the variational optical flow technique described in both Rashwan et al. (2011, 2012) and illustrated in details, in Chapter 3, by replacing the discontinuity-preserving filtering stage based on tensor voting by a stage exclusively based on stick tensor voting in order to significantly reduce the computational time, which is currently around half the processing time corresponding to the technique initially proposed in Chapter 3. Furthermore, a weighted non-local term (a practical median filter) is used to improve the details of the estimated flow fields along discontinuities. The proposed weighted non-local term depends on both the occlusion state of pixels, as proposed in Sand and Teller (2008), and the *surfaceness* saliency obtained through stick tensor voting.

8.1.4 Illumination-robust optical flow estimation model

Video surveillance systems are badly affected by illumination changes and most classical optical flow approaches used for surveillance systems can not cope with illumination changes. Therefore, Chapter 5 introduced a robust optical flow approach which is very robust concerning illumination changes based on the histogram of oriented gradients (HOG). The optical flow model estimates dense flow fields using a duality of the TV-L1 optical flow model with a non-local term. The HOG descriptor, which is robust to illumination changes, has been used in order to define an alterative data term. The proposed approach yields the most accurate flow fields for real images with both illumination changes and large displacements.

8.1.5 Privacy-aware approach to store video surveillance

In Chapter 6, we have presented a trustworthy platform for privacy-preserving video surveillance. We have defined the properties that such a system must fulfill in order to be trustworthy. To the best of our knowledge, this is the only proposal of privacy-aware video surveillance system from a holistic perspective, taking into account several aspects instead on only focusing on detection and simple protection of ROIs. We have divided our platform between a Detection Submodule, a Pro-

tection Submodule and an Information System. We also involve a Law Enforcer authority.

We have overviewed the different trends in ROI detection (face detection, background subtraction and optical flow). We have stated that, on the one hand, it is necessary to use full moving objects (such as bodies) as ROIs instead of faces; on the other hand, we have addressed the advantages and disadvantages of the different groups of techniques. We have also recalled the categories of protection methods: transformation in the pixel domain and in the compressed domain (the latter fulfills the property of utility since the methods are fully reversible). We have focused on the coefficient alteration as the best protection method. The protection and unprotection depend on a pseudorandom bit stream (the protection stream). We have described a method to generate this sequence in a secure manner. To prevent the Trusted Manager from arbitrarily unprotecting videos, we propose that before protecting and unprotecting a video, the Protection Submodule must contact its counterpart in the Law Enforcer side.

We have implemented a prototype of our proposed platform in order to do some tests. We have discussed the effectiveness of the protection method, in terms of identity concealment, robustness against image reconstruction by means of the Random Alteration Attack and compression efficiency. We have addressed the time performance of the protection procedure and have ended with some discussion on the security of the information and data involved in the platform. The platform has been prototyped in a Intel NUC computer, with i3 CPU.

8.1.6 Attacks against privacy-aware protected video

Chapter 7 has proposed an algorithm for unprotected ROIs (faces) in a protected frame by assuming of a previous knowledge about (i) the faces protected (having access to a public database of facial images) and (ii) the faces have been protected with the algorithm proposed in Martínez-Ballesté and Rashwan (2013).

The proposed algorithm consists of five steps. First, it generates a number of images by randomly altering the 63 AC coefficients with a fixed DC coefficient in order to reduce the complexity of the search space. Secondly, the number of facial images from the resulting random images are selected according to the highest similarity score calculated by the Eigenfaces model. A single image is thirdly generated through median stacking. A number of faces are matched to the resulting single image based on the trained Eigenfaces model. Finally, a reconstructed face is obtained by correcting the DC coefficients of the single image through the DC coefficients of the recognized face(s).

The proposed algorithm has proved to be valid in the empirical tests that have been conducted, achieving good results with two evaluation databases: a public CALTECH face database and a proprietary database. Therefore, the alteration

of AC coefficients with an encrypted DC coefficient is not enough for trustworthy surveillance systems enabling privacy, since our algorithm can significantly succeed guessing the protected person in a frame from the protected video.

8.2 Future research directions

The concepts and the results presented in this dissertation pave the way for new applications and solutions to different detection and protection problems. Some future research directions are summarized below.

8.2.1 A fast automatic optical flow model

The optical flow estimation is of fundamental interest in computer vision with its great importance in many applications. The errors generated during the flow fields estimation can yield wrong results in different tasks, such as automated video surveillance, tracking or gait recognition, among others. Thus, the developing of fast optical flow techniques is very important for practical purposes. Therefore, the models introduced in Chapter 3 can be improved by focusing on the reduction of the computational time of the proposed algorithm, which is currently around six times larger than the one in Zimmer et al. (2009), by applying the technique recently proposed in Moreno et al. (2011b). Furthermore, further work is also required in order to automatically determine the values of the various parameters that have currently been tuned experimentally. This is necessary in order to have a fully automatic optical flow estimation algorithm.

8.2.2 A real-time optical flow estimation

Based on the concepts developed in this thesis, and more precisely on the theoretical optical flow model derived in Chapter 4, new research efforts are being devoted in order to derive real-time solutions for the motion estimation based on optical flow problems. This solution would not only represent a new approach for the optical flow problem, but could also be integrated into a real video surveillance system in order to allow for a reliable system based on an accurate ROIs detection. Therefore, future work will aim at implementing the optical flow model proposed in Chapter 4 on GPU Systems that accelerate computations using modern accelerators such as GPUs and future devices while simplifying implementations. In addition, future work will also focus on hardware implementations such as, the field-programmable gate arrays (FPGAs) that allow the developers to program product features and functions, adapt to new standards, and reconfigure hardware for optical flow applications.

8.2.3 A gait recognition based on optical flow fields

The optical flow model proposed in Chapter 4 can be used to design a novel gait recognition algorithm that will be based on the histogram of co-occurrence of optical flow fields. The optical flow fields are firstly estimated between each two consecutive images during a complete gait cycle using the proposed model. Then, the extracted descriptors along a gait cycle will train a robust classifier based on support vector machines or neural networks for a gait classification. The accurate flow fields will yield an accurate gait recognition used for a robust video surveillance system.

8.2.4 An Illumination-robust and noise-robust optical flow model

In Chapter 5, the proposed optical flow methodology based on the histogram of gradients allows to estimate flow fields that can cope with illumination changes. These illumination-robust models are very useful for dealing with environmental changes surrounding cameras used in surveillance systems. However, to our knowledge, most of the cameras used in surveillance systems suffer from a lot of noise types. Therefore, future work will aim to integrate the two optical flow models in Chapter 4 and Chapter 5 respectively to build both an illumination-robust and a noise-robust optical flow model based on HOG features and tensor voting process.

8.2.5 A new robust ROIs protection

Regarding the protection sub-module mentioned in Chapter 6 and 7, due to the success of the attack algorithm proposed in Chapter 7 for reconstructing the protected faces (with a prior knowledge, *i.e.* public unprotected faces), an immediate work will focus on using a more robust and accurate classification approach to recognize a correct face. In addition, future work aims at reconstructing a face without any previous knowledge, trying to evaluate a randomly generated unprotected face by a robust face-detection approach. Furthermore, the effect of applying a permutation to the non-zero AC coefficients will be studied, aiming at producing a more robust protected image.

Future work can also include, on the one hand, the study of the effect on the protected videos of transformations such as rescaling and cropping. Moreover, we expect to implement some routines taking into account the performance and resource consuming, aiming at allowing at least 15 fps with the Intel NUC, for motion detection based on accurate optical flow techniques and PAL frames.

As a final remark, although new models used for a description of real phenomena of the surveillance system cameras require extensive validation under different

applications, the models introduced in this thesis accurately describe the observed behavior of the surveillance system technology. Moreover, their application for solving specific problems have produced promising results. Notwithstanding, a thorough theoretical analysis incorporating all concepts from surveillance systems, as well as from the design and manufacture of surveillance cameras, could provide additional insights on the applicability and limitations of the introduced models to a wider set of real surveillance systems.

8.3 Publications

The following publications have been derived from this thesis:

1. Rashwan H. A., Puig D. and Garcia M. A.: On improving the robustness of differential optical flow, Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, 2011, pp. 876-881.

2. Rashwan H. A., Garcia M. A. and Puig D.: Improving the robustness of variational optical flow through tensor voting Computer Vision and Image Understanding, Academic Press, 2012, vol(116), pp. 953-966.

3. Martínez-Ballesté A., Rashwan H. A., Puig D. and Fullana A.P.: Towards a trustworthy privacy in pervasive video surveillance systems. PerCom, 2012, pp. 914-919.

4. Martínez-Ballesté A., Rashwan H. A., Castellà-Roca J. and Puig D.: A trustworthy database for privacy-preserving video surveillance Proceedings of the Joint EDBT/ICDT 2013 Workshops, 2013, pp. 179-183.

5. Rashwan H. A., Garcia M. A. and Puig D.: Variational Optical Flow Estimation Based on Stick Tensor Voting. IEEE Transactions on Image Processing, 2013, vol(22), pp. 2589-2599.

6. Rashwan H. A., Mohamed M. A., Garcia M. A., Mertsching B. and Puig D.: Illumination Robust Optical Flow Model Based on Histogram of Oriented Gradients Pattern Recognition, Springer Berlin Heidelberg, 2013, pp. 354-363.

7. Mohamed M. A., Rashwan H. A., Mertsching B., Garcia M. A. and Puig D.: On Improving the Robustness of Variational Optical Flow against Illumination Changes, 21st ACM International Conference on Multimedia, 2013, pp. 250-258.

8. Martínez-Ballesté A., Solanas A., Segarra M.V. and Rashwan H. A.: Privacy in Pervasive Video Surveillance: Trust through Technology and Users Cooperation, 4^{th} International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS), 2014.

9. Rashwan H. A., Mohamed M. A., Mertsching B., Garcia M. A., and Puig D.: Illumination-Robust Optical Flow Using Local Directional Pattern, IEEE Transactions on Circuits and Systems for Video Technology, 2014.

10. Rashwan H. A., Martínez-Ballesté A., Puig D., "Towards the Implementation of a Trusted Privacy-Awareness Video Surveillance System", SAAEI 2014, Morocco, To appear.

11. Rashwan H. A., Martínez-Ballesté A., Solanas A. and Puig D.: Understanding Trust in Privacy-Aware Video Surveillance Systems, Journal of Information Security and Applications, Submitted manuscript.

12. Rashwan H. A., Martínez-Ballesté A., Solanas A. and Puig D.: A secure and trustworthy platform for privacy-aware video surveillance system, IEEE Transactions on Information Forensics and Security, Submitted manuscript.

13. Rashwan H. A., Martínez-Ballesté A., Garcia M. A. and Puig D.: Face Reconstruction against a Trustworthy Privacy-Preserving Video Surveillance System, Image and Vision Computing, Submitted manuscript.

14. Rashwan H. A., Garcia M. A. and Puig D.: Gait Representation and Recognition From Temporal Co-occurrence of Flow Fields, Pattern Recognition, Submitted manuscript.

Bibliography

Arredondo, M. A., Lebart, K., and Lane, D. (2004). Optical flow using textures. *Pattern Recogn. Lett.*, 25(4):449–457.

Baf, F., Bouwmans, T., and Vachon, B. (2008). Type-2 fuzzy mixture of gaussians model: Application to background modeling. In *Proceedings of the 4th International Symposium on Advances in Visual Computing*, ISVC '08, pages 772–781, Berlin, Heidelberg. Springer-Verlag.

Bainbridge-Smith, a. and Lane, R. (1997). Determining optical flow using a differential method. *Image and Vision Computing*, 15(1):11–22.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2010). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72.

Barron, J. L., Fleet, D. J., Beauchemin, S. S., and Burkitt, T. A. (1992). Performance of optical flow techniques. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 236–242.

Berger, A. M. (2000). Privacy mode for acquisition cameras and camcorders. [Online] avaiable: http://www.google.com/patents/US6067399. Sony cooperation.

Bigun, J., Granlund, G. H., and Wiklund, J. (1991). Multidimensional orientation estimation with applications to texture analysis and optical flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(8):775–790.

Black, M. J. and Anandan, P. (1991). Robust dynamic motion estimation over time. In *Computer Vision and Pattern Recognition, Proceedings CVPR '91., IEEE Computer Society Conference on*, number June, pages 296–302.

Bosc, M., Heitz, F., paul Armspach, J., Namer, I., Gounot, D., and Rumbach, L. (2003). Automatic change detection in multimodal serial mri: application to multiple sclerosis lesion evolution.

Boult, T. (2005). Pico: Privacy through invertible cryptographic obscuration. In *Computer Vision for Interactive and Intelligent Environment*, pages 27–38. Ieee.

Bouwmans, T. (2008). Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 1(3)(3):219–237.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision-ECCV 2004*, volume 4, pages 25–36.

Brox, T. and Malik, J. (2011). Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513.

Bruhn, A. and Weickert, J. (2005). Towards ultimate motion estimation: combining highest accuracy with real-time performance. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 749–755 Vol. 1.

Bruhn, A., Weickert, J., Kohlberger, T., and Schnörr, C. (2006). A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, 70(3):257–277.

Bruhn, A., Weickert, J., and Schnörr, C. (2005). Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.

Bruzzone, L. and Prieto, D. F. (2002). An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. *Trans. Img. Proc.*, 11(4):452–466.

Carrillo, P., Kalva, H., and Magliveras, S. (2009). Compression independent reversible encryption for privacy in video surveillance. *EURASIP J. Inf. Secur.*, 2009:5:1–5:13.

Cavallaro, A. (2004). Adding privacy constraints to video-based applications. In Hobson, P., Izquierdo, E., Kompatsiaris, I., and O'Connor, N. E., editors, *EWIMT*. QMUL.

Cavallaro, A. (2007). Privacy in video surveillance. *IEEE SIGNAL PROCESSING MAGAZINE*, 20(March):166–168.

Chabrier, S., Emile, B., Rosenberger, C., and Laurent, H. (2006). Unsupervised performance evaluation of image segmentation. *EURASIP J. Appl. Signal Process.*, 2006:217–217.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97.

Chambolle, A. and Lions, P.-L. (1997). Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188.

Cheung, S. (2008). Managing privacy data in pervasive camera networks. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1676–1679.

Clarke, R. (2001). Introducing pits and pets: Technologies affecting privacy. [Online] avaiable: http://www.rogerclarke.com/DV/PITsPETs.html.

Collins, R. T., Lipton, A. J., and Kanade, T. (2000). Introduction to the special section on video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):745–746.

Connell, J., Senior, A., Hampapur, A., Tian, Y.-L., Brown, L., and Pankanti, S. (2004). Detection and tracking in the ibm peoplevision system. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 2, pages 1403–1406 Vol.2.

Cristani, M., Bicego, M., and Murino, V. (2003). Multi-level background initialization using hidden markov models. In *First ACM SIGMM international workshop on Video surveillance*, IWVS '03, pages 11–20, New York, NY, USA. ACM.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1.

Dufaux, F. (2006). Privacy enabling technology for video surveillance. In *Proc. SPIE 6250, Mobile Multimedia/Image Processing for Military and Security*.

Dufaux, F. and Ebrahimi, T. (2008). Scrambling for privacy protection in video surveillance systems. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1168–1174.

Elgammal, A. M., Harwood, D., and Davis, L. S. (2000). Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, ECCV '00, pages 751–767, London, UK, UK. Springer-Verlag.

Elizondo, D., Solanas, A., and Martínez-Ballesté, A. (2012). *Computational Intelligence for Privacy and Security*. Springer.

Fang, C.-Y., Chen, S.-W., and Fuh, C.-S. (2003). Automatic change detection of driving environments in a vision-based driver assistance system. *Neural Networks, IEEE Transactions on*, 14(3):646–657.

Farneback, G. (2000). Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Pattern Recognition, 2000. Proceedings. 15th*.

Fermüller, C., Shulman, D., and Aloimonos, Y. (2001). The statistics of optical flow. *Computer Vision and Image Understanding*, 82(1):1–32.

Froba, B. and Ernst, A. (2004). Face detection with the modified census transform. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 91–96.

Fukuoka, N., Ito, Y., and Babaguchi, N. (2012). Delivery method for viewer-specific privacy protected video using discrete wavelet transform. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 2285–2288.

Gabbouj, M., Coyle, E. J., and Gallagher, N. C. (1992). An overview of median and stack filtering. In *Circuits, Systems, and Signal Processing, Special issue on Median and Morphological Filtering*, pages 7–45.

Galvin, B., Mccane, B., Novins, K., Mason, D., and Mills, S. (1998). Recovering motion fields: An evaluation of eight optical flow algorithms. In *British Machine Vision Conference*, pages 195–204.

Gaucher, L. and Medioni, G. (1999). Accurate motion flow estimation with discontinuities. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 695–702 vol.2.

H. Wactlar, S. S. and Ng, T. (2002). Enabling personal privacy protection preferences in collaborative video observation. In *NSF Award*. Springer.

Hadid, A. and Pietik, M. (2004). A discriminative feature space for detecting and recognizing faces 2 face description with local binary. *Computer Vision and Pattern Recognition,*, 2(ii):797–804.

Herrero, S. and Bescós, J. (2009). Background subtraction techniques: Systematic evaluation and comparative analysis.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203.

Huertas, A. and Nevatia, R. (1998). Detecting changes in aerial views of man-made structures. In *in Proceedings of the Sixth International Conference on Computer Vision (ICCV98*, pages 73–79.

Hung, C. H., Xu, L., and Jia, J. (2012). Consistent binocular depth and scene flow with chained temporal profiles. *International Journal of Computer Vision*, 102(1-3):271–292.

Jia, K., Wang, X., and Tang, X. (2011). Optical flow estimation using learned sparse model. In *Computer Vision (ICCV), 2011 IEEE*, number 60903115.

Kervrann, C. and Boulanger, J. (2008). Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *Int. J. Comput. Vision*, 79(1):45–69.

Kim, K., Chalidabhongse, T. H., Hanuood, D., and Davis, L. (2004). Background modeling and subtraction by codebook construction. In *International Conference on Image Processing, ICIP '04.*, pages 3061–3064.

Kim, Y.-H., Martínez, A. M., and Kak, A. C. (2005). Robust motion estimation under varying illumination. *Image Vision Comput.*, 23(4):365–375.

Krähenbühl, P. and Koltun, V. (2012). Efficient nonlocal regularization for optical flow. In *European Conference on Computer Vision (ECCV)*.

Li, S., Li, C., Chen, G., Bourbakis, N. G., and Lo, K.-T. (2008). A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing: Image Communication*, 23(3):212 – 223.

Li, Y. and Osher, S. (2009). A new median formula with applications to pde based denoising. *Commun. Math. Sci*, 7(3)(x):741–753.

Liou, S. P. and Jain, R. C. (1989). Motion detection in spatio-temporal space. *Computer Vision, Graphics, and Image Processing*, 45(2):227–250.

Little, J. J., Bulthoff, H. H., and Poggio, T. (1988). Parallel optical flow using local voting. In *Computer Vision., Second International Conference on*, pages 454–459.

Liu, C., Freeman, W. T., Adelson, E. H., and Weiss, Y. (2008). Human-assisted motion annotation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. Ieee.

Liu, C., Yuen, J., and Torralba, A. (2011). Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence IJCAI81 - Volume 2*, volume 130, pages 121–129.

Martin, K. and Plataniotis, K. N. (2008). Privacy protected surveillance using secure visual object coding. *IEEE Trans. on Circuit and Systems for Video Technology*, 18(8):1152–1162.

Martínez-Ballesté, A. and Rashwan, H. A. (2013). A trustworthy database for privacy-preserving video. In *Proceeding EDBT '13 Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 179–183.

Martínez-Ballesté, A., Rashwan, H. A., Puig, D., and Fullana, A. P. (2012). Towards a trustworthy privacy in pervasive video surveillance systems. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 914–919.

Martínez-Ponte, I., Desurmont, X., Meessen, J., and Delaigle, J. (2005). Robust human face hiding ensuring privacy. In *in Proc. of International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*.

Mattavelli, M. and Nicoulin, A. (1994). Motion estimation relaxing the constancy brightness constraint. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 770–774 vol.2.

Medioni, G., Tang, C., and Lee, M. (2000). Tensor voting: Theory and applications. In *Proceedings of RFIA, Paris, France*.

Mileva, Y., Bruhn, A., and Weickert, J. (2007). Illumination-robust variational optical flow with photometric invariants. In *In DAGM-Symposium, LNCS 4713*, pages 152–162.

Molnár, J., Chetverikov, D., and Fazekas, S. (2010). Illumination-robust variational optical flow using cross-correlation. *Comput. Vis. Image Underst.*, 114(10):1104–1114.

Moreno, R., Garcia, M. A., Puig, D., and Julia, C. (2011a). Edge-preserving color image denoising through tensor voting. *Computer Vision and Image Understanding*, 115(11):1536 – 1551.

Moreno, R., Garcia, M. A., Puig, D., Pizarro, L., Burgeth, B., and Weickert, J. (2011b). On improving the efficiency of tensor voting. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2215–28.

Müller, T., Rabe, C., Rannacher, J., Franke, U., and Mester, R. (2011). Illumination-robust dense optical flow using census signatures. In *Proceedings of the 33rd international conference on Pattern recognition*, DAGM'11, pages 236–245, Berlin, Heidelberg. Springer-Verlag.

Nagel, H.-H. (1983). Constraints for the estimation of displacement vector fields from image sequences. In *Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 2*, IJCAI'83, pages 945–951, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(5):565–593.

Nagy, E., Zhang, T., Franklin, W., Landis, E., Nagy, E., and Keane, D. (2001). Volume and surface area distributions of cracks in concrete. In *in Proc. Visual Form, 2001*, pages 759–768.

Newton, E., Sweeney, L., and Malin, B. (2005). Preserving privacy by de-identifying face images. *Knowledge and Data Engineering, IEEE Transactions on*, 17(2)(March):232–243.

Nicolescu, M. and Medioni, G. (2003). Motion segmentation with accurate boundaries - a tensor voting approach. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, pages I–382–I–389. IEEE Comput. Soc.

Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: an application to face detection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 130–136.

Peng, F., wen Zhu, X., and Long, M. (2013). An roi privacy protection scheme for h.264 video based on fmo and chaos. *Information Forensics and Security, IEEE Transactions on*, 8(10):1688–1699.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1993). *Numerical Recipes in FORTRAN; The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2nd edition.

Rashwan, H., Garcia, M., and Puig, D. (2013). Variational optical flow estimation based on stick tensor voting. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, (JULY 2012):1–11.

Rashwan, H. A., Puig, D., and Garcia, M. A. (2011). On improving the robustness of differential optical flow. In *2011 IEEE International Conference on Computer Vision Workshops ICCV Workshops*, pages 876–881. IEEE.

Rashwan, H. a., Puig, D., and Garcia, M. A. (2012). Improving the robustness of variational optical flow through tensorvoting. *Computer Vision and Image Understanding*, 116(9):953–966.

Ren, Y., Chua, C.-S., and Ho, Y.-K. (2003). Statistical background modeling for non-stationary camera. *Pattern Recogn. Lett.*, 24(1-3):183–196.

Rey, D., Subsol, G., Delingette, H., and Ayache, N. (1999). Automatic detection and segmentation of evolving processes in 3d medical images: Application to multiple sclerosis. In *Proceedings of the 16th International Conference on Information Processing in Medical Imaging*, IPMI '99, pages 154–157, London, UK, UK. Springer-Verlag.

Rowley, H., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, 20:23–36.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268.

Said, A. and Pearlman, W. (1996). A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(3)(3):243–250.

Sand, P. and Teller, S. (2008). Particle video: Long-range motion estimation using point trajectories. *Int. J. Comput. Vision*, 80(1):72–91.

Schnorr, C. (1993). On functionals with greyvalue-controlled smoothness terms for determining optical flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(10):1074–1079.

Schnorr, C. (1994). Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, pages 661–663.

Senior, A. (2009). *Protecting Privacy in Video Surveillance*. Springer Publishing Company, Incorporated, 1st edition.

Senior, A., Pankanti, S., Hampapur, A., Brown, L., Tian, Y.-L., Ekin, A., Connell, J., Shu, C.-F., and Lu, M. (2005). Enabling video privacy through computer vision. *Security Privacy, IEEE*, 3(3):50–57.

Shahid, Z., Chaumont, M., and Puech, W. (2011). Fast protection of h.264/avc by selective encryption of cavlc and cabac for i and p frames. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(5):565–576.

Shavers, C., Li, R., and Lebby, G. (2006). An svm-based approach to face detection. In *System Theory, 2006. SSST '06. Proceeding of the Thirty-Eighth Southeastern Symposium on*, pages 362–366.

Sohn, H., AnzaKu, E. T., De Neve, W., Ro, Y. M., and Plataniotis, K. N. (2009). Privacy protection in video surveillance systems using scalable video coding. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 424–429. Ieee.

Solanas, A. and Martínez-Ballesté, A. (2008). A ttp-free protocol for location privacy in location-based services. *Computer Communications*, 31(6):1181–1191.

Spinder, T., Roth, D., and v. Gool, L. (2006). Privacy in video surveilled areas. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*.

Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, pages 246–252. IEEE Comput. Soc.

Stein, F. (2004). Efficient Computation of Optical Flow Using the Census Transform. In Rasmussen, C., Bülthoff, H., Schölkopf, B., and Giese, M., editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 79–86. Springer Berlin Heidelberg.

Sun, D., Roth, S., and Black, M. J. (2010a). Secrets of optical flow estimation and their principles. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2432–2439. Ieee.

Sun, D., Sudderth, E., and Black, M. (2010b). Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *Advances in Neural Information Processing Systems,*, 23:1–9.

Tansuriyavong, S. and Hanaki, S.-i. (2001). Privacy protection by concealing persons in circumstantial video image. In *Proceedings of the 2001 workshop on Percetive user interfaces - PUI01*, page 1, New York, New York, USA. ACM Press.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846.

Tong, W.-S., Tang, C.-K., and Medioni, G. (2001). First order tensor voting, and application to 3-d scale analysis. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–175–I–182 vol.1.

Turk, M. and Pentland, A. (1991). Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 586–591.

Upmanyu, M., Namboodiri, A., Srinathan, K., and Jawahar, C. (2009). Efficient privacy preserving video surveillance. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1639–1646.

Viejo, A. and Castellà-Roca, J. (2010). Using social networks to distort users profiles generated by web search engines. *Computer Networks*, 54(9):1343–1357.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages 511–518. IEEE Comput. Soc.

Volz, S., Bruhn, A., Valgaerts, L., and Zimmer, H. (2011). Modeling temporal coherence for optical flow. In *2011 International Conference on Computer Vision*, pages 1116–1123. Ieee.

Weickert, J., Bruhn, A., Brox, T., and Papenberg, N. (2006). A survey on variational optic flow methods for small displacements. In Scherzer, O., editor, *Mathematical Models for Registration and Applications to Medical Imaging*, volume 10 of *Mathematics in Industry*. Springer, Berlin.

Weickert, J. and Schnórr, C. (2001). A theoretical framework for convex regularizers in pde-based. *International Journal of Computer Vision*, 45(3):245–264.

Werlberger, M., Pock, T., and Bischof, H. (2010). Motion estimation with non-local total variation regularization. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2464–2471. Ieee.

Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H. (2009). Anisotropic huber-l1 optical flow. In *Procedings of the British Machine Vision Conference 2009*, volume 108, pages 1–11. British Machine Vision Association.

Wickramasuriya, J., Datt, M., Mehrotra, S., and Venkatasubramanian, N. (2004). Privacy protecting data collection in media spaces. In *Proceedings of the 12th annual ACM international conference on Multimedia - MULTIMEDIA 04*, page 48, New York, New York, USA. ACM Press.

Winkler, T. and Rinner, B. (2010). A systematic approach towards user-centric privacy and security for smart camera networks. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras - ICDSC10*, page 133, New York, New York, USA. ACM Press.

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

Xu, L., Jia, J., and Matsushita, Y. (2012). Motion detail preserving optical flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1744–57.

Yabuta, K., Kitazawa, H., and Tanaka, T. (2005). A new concept of security camera monitoring with privacy protection by masking moving objects. *Advances in Multimedia Information Processing - PCM, Lecture Notes in Computer Science*, 3767:831–842.

Yang, M.-H. (2009). Face detection. In *Encyclopedia of Biometrics*, pages 303–308. Springer.

Yue, Y., Gao, Y., and Zhang, X. (2009). An improved tracking algorithm in crowded scenes and dynamic background. In *Proceedings of the First International Conference on Internet Multimedia Computing and Service - ICIMCS09*, page 39, New York, New York, USA. ACM Press.

Zabih, R., , Zabih, R., and Ll, J. W. (1994). Non-parametric local transforms for computing visual correspondence. pages 151–158. Springer-Verlag.

Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the 29th DAGM conference on Pattern recognition*, pages 214–223, Berlin, Heidelberg. Springer-Verlag.

Zhang, C. and Zhang, Z. (2010). A survey of recent advances in face detection. Technical report, Techniqal report in Microsoft research.

Zimmer, H., Bruhn, A., and Weickert, J. (2011). Optic flow in harmony. *International Journal of Computer Vision*, 93(3):368–388.

Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., and Seidel, H.-P. (2009). Complementary optic flow. In *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR '09, pages 207–220, Berlin, Heidelberg. Springer-Verlag.