IVORA (Image and Computer Vision for Augmented Reality) : Color invariance and correspondences for the definition of a camera/video-projector system

Aleksandr Setkov





NNT: 2015SACLS168

UNIVERSITÉ PARIS-SACLAY

ÉCOLE DOCTORALE : Sciences et Technologies de l'Information et de la Communication

Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur

DISCIPLINE : Informatique

THÈSE DE DOCTORAT

présenté et soutenu publiquement par

Aleksandr SETKOV

le 27 novembre 2015

IVORA (IMAGE AND COMPUTER VISION FOR AUGMENTED REALITY): COLOR INVARIANCE AND CORRESPONDENCES FOR THE DEFINITION OF A CAMERA/VIDEO-PROJECTOR SYSTEM

Composition du jury :

Rapporteurs :	Alain TREMEAU	Professeur (Université Jean Monnet Saint-Etienne)
	Anthony STEED	Professor (University College London)
Président :	Samia BOUCHAFA	Professeur (Université d'Évry-Val-d'Essonne)
Examinateur :	Catherine ACHARD	Maître de conférences (HDR) (Université Pierre et Marie CURIE)
Directeur de thèse :	Christian JACQUEMIN	Professeur (LIMSI)
Encadrant :	Michèle GOUIFFES	Maître de conférences (LIMSI)

Acknowledgements

First of all, I would like to express all my gratitude to my supervisors, Christian Jacquemin and Michèle Gouiffès, who thoroughly and unceasingly guided me through all of the three years of my PhD work. Although the work towards a PhD was not simple, I always received constructive and timely feedback, and sometimes a portion of fair and reasonable criticism. All those helped me to stay on my way to completing a PhD. As a result of our joint work, I can remember only good moments, interesting projects, collaborations, and publications.

I would also like to thank Maria Vanrell and Ramon Baldrich who supervised me during my stay at CVC UAB in Barcelona. I believe that our work was interesting and fruitful for both sides. Acknowledgments should also be given to EIT Digital Doctoral School, which funded this international mobility.

Second, I am thankful to the reviewers of my manuscript, Anthony Steed and Alain Trémeau, for their constructive feedback that broached many interesting topics related to my thesis. I also thank the examiners of my PhD committee, Samia Bouchafa and Catherine Achard for interesting questions in my PhD defense.

Next, I want to thank my colleagues at LIMSI, including Alexandros and Fabio, whom I have had the pleasure to work with. Moreover, a lot of thanks to my other LIMSI colleagues with whom I spent three happy years working together, sharing meals at CESFO and playing soccer.

I am so grateful to my French friends, Guillaume, Florine, Lucie, Véronique, and Laurent, whose help was invaluable. I also thank my Mexican friends, David and Rigel, for the lot of fun we had in our free time. Of course, my heartfelt gratitude goes to my Russian childhood friends, Serega, Ilya B., and Valya, and to my university friends Ilya K., and Artur, for persistently supporting me while I was far from home.

Finally, I want to thank my dear parents and all my relatives who, despite being far from me, continuously sent me their support and inspiration. Also, I endlessly thank my girlfriend, Aina, for making every single day better.

Abstract

Spatial Augmented Reality (SAR) aims at spatially superposing virtual information on real-world objects. Over the last decades, it has gained a lot of success and been used in manifold applications in various domains, such as medicine, prototyping, entertainment etc. However, to obtain projections of a good quality one has to deal with multiple problems, among them the most important are the limited projector output gamut, ambient illumination, color background, and arbitrary geometric surface configurations of the projection scene. These factors result in image distortions which require additional compensation steps.

Smart-projections are at the core of PAR applications. Equipped with a projection and acquisitions devices, they control the projection appearance and introduce corrections on the fly to compensate distortions. Although active structured-light techniques have been so far the de-facto method to address such problems, this PhD thesis addresses a relatively new unintrusive content-based approach for geometric compensation of multiple planar surfaces and for object recognition in SAR.

Firstly, this thesis investigates the use of color-invariance for feature matching quality enhancement in projection-acquisition scenarios. The performance of most state-of-the art methods are studied along with the proposed local histogram equalization-based descriptor. Secondly, to better address the typical conditions encountered when using a projector-camera system, two datasets of real-world projections were specially prepared for experimental purposes. Through a series of evaluation frameworks, the performance of all considered algorithms is thoroughly analyzed, providing several inferences on that which algorithms are more appropriate in each condition. Thirdly, this PhD work addresses the problem of multiple-surface fitting used to compensate different homography distortions in acquired images. A combination of feature matching and Optical Flow tracking is proposed in order to achieve a more low-weight geometric compensation. Fourthly, an example of new application to object recognition from acquired projections is showed. Finally, a real-time implementation of considered methods on GPU shows prospects for the unintrusive feature matching-based geometric compensation in SAR applications.

Abstract

Les techniques de Réalité Augmentée Spatiale (SAR) visent à superposer spatialement l'information virtuelle sur des objets physiques. Au cours des dernières décennies elles ont connu une grande expansion et sont utilisées dans divers domaines, tels que la médecine, le prototypage, le divertissement etc. Cependant, pour obtenir des projections de bonne qualité, plusieurs problèmes doivent être résolus, les plus importants étant la gamme de couleurs réduite du projecteur, la lumière ambiante, la couleur du fond, et la configuration arbitraire de la surface de projection dans la scène. Ces facteurs entraînent des distorsions dans les images qui requièrent des processus de compensation complémentaires. Les projections intelligentes (smart projections) sont au cœur des applications de SAR. Réalisées à partir d'un dispositif de projection et d'un dispositif d'acquisition, elles contrôlent l'aspect de la projection et effectuent des corrections à la volée pour compenser les distorsions. Bien que les méthodes actives de Lumière Structurée aient été utilisées classiquement pour résoudre ces problèmes de compensation géométrique, cette thèse propose une nouvelle approche non intrusive pour la compensation géométrique de plusieurs surfaces planes et pour la reconnaissance des objets en SAR s'appuyant uniquement sur la capture du contenu projeté. Premièrement, cette thèse étudie l'usage de l'invariance couleur pour améliorer la qualité de la mise en correspondance entre primitives dans une configuration d'acquisition des images vidéoprojetées. Nous coomparons la performance de la plupart des méthodes de l'état de l'art avec celle du descripteur proposé basé sur l'égalisation d'histogramme. Deuxièmement, pour mieux traiter les conditions standard des systèmes projecteur-caméra, deux ensembles de données de captures de projections réelles, ont été spécialement préparés à des fins expérimentales. La performance de tous les algorithmes considérés est analysée de façon approfondie et des propositions de recommandations sont faites sur le choix des algorithmes les mieux adaptés en fonction des conditions expériementales (paramètres image, disposition spatiale, couleur du fond...). Troisièmement, nous considérons le problème d'ajustement multi-surface pour compenser des distorsions d'homographie dans les images acquises. Une combinaison de mise en correspondance entre les primitives et de Flux Optique est proposée afin d'obtenir une compensation géométrique plus rapide. Quatrième-

ment, une nouvelle application en reconnaissance d'objet à partir de captures d'images vidéo-projetées est mise en œuvre. Finalement, une implémentation GPU temps réel des algorithmes considérés ouvre des pistes pour la compensation géométrique non intrusive en SAR basée sur la mise en correspondances entre primitives.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context

Projection-based (Spatial) Augmented Reality (PAR or SAR) is a form of Virtual Reality that superposes virtual information on real-world objects by means of projection devices. Over the last decades it has been used for many applications in cultural heritage, medicine, industry, computer games production and many others. The central role in such applications is played by the Projector-Camera system that generally combines one or several projector devices and a camera or proprioceptive sensors. With the development of modern technologies, such as pico-projectors, the size of devices has been reducing significantly along with their cost which makes them appropriate for the use in various everyday applications.

The main benefit of the Projector-Camera (ProCam) system is the possibility to control and dynamically adjust the projected content to achieve the desirable appearance. It is secured by placing the camera so that it captures the part of the scene where the content is projected. Then the system analyzes projections acquired with the camera and compares them with the reference images. Finally, according to the computed difference, it introduces a correction to the reference image so that its projection is perceived by the viewer as similar to the original image.

Unfortunately, the compensation appears to be a very difficult task due to numerous conditions to be accounted for, such as the ambient illumination, colorful or textured background and the geometry of the projection scene. All these components distort the final projection photometrically and geometrically, in such a manner that it can hardly be precisely estimated by a theoretical model. To deal with this problem, approximated models are widely used. Geometric and photometric distortions are often compensated separately, which means that color correction methods assume that the projection and the reference

image are perfectly registered. Similarly, strong photometric distortions may complicate geometric transform estimation.

The distortion compensation process in Camera-Projector Systems usually consists of two main stages. The first one is offline-calibration which aims at estimating the color and geometric device parameters of both camera and projector that do not change while the system is working. The set of parameters may contain, for example, the camera and projector responses, geometrical per-pixel mapping from projector to camera, and the illumination in the scene if it remains constant over time. The second part of the compensation process is meant for the estimation of missing parameters that change dynamically such as surface geometry and background color. Once all the parameters are estimated, the compensated image is computed and projected back in the scene.

The focus of my PhD thesis is geometric compensation of the projection with any color correction precedently applied. For instance, uncalibrated projection and acquisition devices as well as different illuminations are the factors that need to be addressed in this case.

There have been a lot of works on geometric projection compensation in Projector-Camera systems. Most of them belong to the group of active methods which is based on structured light [41, 96, 97, 128, 138]. Such algorithms embed special patterns in the projected images to facilitate information extraction which is relevant to color and geometry estimation. Although they can address both the static and the dynamic scenarios without any severe constraints on the projection scene, they still have some drawbacks. Firstly, the initial projected content has to be modified to embed the patterns. This limitation can be incompatible with some applications that require high fidelity of rendering, for example medical PAR applications. Secondly, depending on the coding strategy, a projected pattern either interleaves the image data sequence or is embedded in the original images which can interfere with the human perception. The former requires a precise camera-projector synchronization [25], whereas the latter makes it extremely difficult to imperceptibly embed a pattern into the stream using standard commercial cameras [97]. Moreover it puts some constraints on the projected colors and affects the projected frame-rate which can be a restrictive limitation for such applications as teleconferences or video content projections.

Using imperceptible infrared structured-light scanners [4, 8, 122] can solve the aforementioned problems. However, most of them have strong limitations on the maximum work distance and on the amount of acceptable daylight in the scene. Moreover, the affordable devices, kinect for example, have low resolution.

Apart from the structured light approach there have been a few attempts to perform geometric compensation unintrusively by relying only on initial content and captured pro-

jections [31, 127, 129]. Motivated by these works, this PhD thesis explores this elegant approach that does not require any additional projections. The key role here is played by local features that are extracted and used to match images. The major drawback of such approach, therefore, is that the compensation quality highly depends on the projected image content. When the images consist of homogenous areas, it becomes very difficult to estimate the transformation due to low presence of distinctive morphological features. Another problem lies in the fact that geometric compensation usually precedes color correction. When projected and acquired, the image can undergo complex photometric distortions due to simultaneously camera and projector responses, color background and ambient illumination. Moreover, acquistions can be spoiled by the camera image noise or by defocus blur. For these reasons, the geometric transformation often has to be estimated on two photometrically different images.

Since the projection may undergo very complex geometrical transformations, in my work I choose local image feature matching as an adequate solution to cope with such distortions. Because local features are spread throughout the image, they are less sensitive to partial occlusions and local deformations. However, the local features in their classical definition [6, 71] are not robust to the complex color changes [118]. The classical descriptors, computed on the grayscale image channel, cannot provide sufficient invariance to deal with color changes in the images to be matched.

To address various illumination conditions in the scene, I make use of color feature descriptors that have been proposed to combine both color invariance and geometric robustness of feature matching [108, 109, 118]. They differ from the intensity-based ones in that the descriptor is computed on several image channels, instead of using only grayscale images. The reference image is preliminarily transformed from RGB to a color-invariant space.

1.2 Contributions

This PhD thesis addresses the problem of geometric correction in Projector-Camera systems in the presence of complex illumination.

The questions formulated in my work are the following ones:

- is it possible to geometrically compensate video projections on the basis of the only projected content;
- what are the more suited color invariants in the context of ProCams;
- can it be performed in real-time.

Since the chosen approach does not rely on any source of informations except the projected images themselves, it is important to ensure robust matching of the reference images with the acquired projections. This represents a very challenging task due to *(i)* arbitrary nature of projected images and *(ii)* various distortions introduced to the model both by the devices through the projection and acquisition processes, and by the environment through ambient illumination and unhomogenous color background. The problem of geometric compensation in the context of complex illumination is addressed by distinctive local features extracted from the matched images. In order to cope with complex and varying photometric conditions, my research work focuses on color invariance property of feature descriptors that were chosen as a baseline for this work.

In order to compensate for geometric distortions due to projections on multiple surfaces, my work explores a combination of homography estimation and heuristic methods to handle the case of multiple planar transformations. The methods benefit from color-invariant feature descriptors to improve the performance of the Projector-Camera system.

To stress again the difference in the approaches, this PhD work addresses unintrusive geometric compensation that needs only one image frame to estimate and compensate for distortions and does not require projections of any artificial pattern. In contrast, most previous works are based on more intrusive structured light techniques that interleave patterns with projected image frames.

For the considered algorithms this PhD thesis presents qualitative and quantitative evaluations on static data that reflect the quality of geometric compensation achieved using these methods. For this purpose I introduce a new database of projected images, called ProCam database, that I have built for the purpose of evaluating algorithms for geometric compensation in static projector-camera systems. Furthermore, the presented database is extended by added video projections to cover dynamic scenarios with mobile ProCams.

For dynamic compensations, in this PhD thesis I address the problem of spatial-temporal feature matching for projection compensation. In this way, I introduce a new compensation framework that is based on feature matching and Optical Flow point tracking. This temporal adaptation significantly decreases the algorithms time complexity, while keeping the compensation quality on a similar level.

The last part of the thesis is devoted to algorithms implementations. A real-time compensation system based on GPU programming has been evaluated in terms of runtime performance and compensation quality.

To sum up, the contributions of this work are the following:

1. A comprehensive investigation in the domain of Color Invariant Feature Matching is

presented in the thesis;

2. A Local Histogram Equalization (LHE) method for descriptor computation is introduced to improve the robustness of Feature Matching against complex photometric distortions;
3. Compensation of single- and double-homography transforms in the scene is ensured;
4. A new database of projection images (ProCam database) is presented and used for qualitative algorithm evaluation. A new application of this database for object recognition is presented;
5. Video projections are included in the ProCam database in order to cover the scenario when projector-camera systems are dynamic;
6. A new spatial-temporal compensation framework is introduced in order to efficiently handle projected video sequences;
7. Projection compensation through feature matching is optimized with the use of GPU and evaluated in a real-time compensation framework in terms of run time and compensation quality.

1.3 Thesis outline

The PhD thesis is organized as follows. Chapter 2 first provides an overview of existing SAR applications. Then, it goes into detail describing three approaches for geometric compensation: the active one, based on Structured-Light, the passive one which relies on projected image content, and the hybrid one that makes use of both passive and active compensations. Chapter 3 focuses on color invariant feature matching and its application for the use in a projector-camera system, that is for projective transforms compensation. Starting from a review of state-of-the-art methods in local feature matching, the chapter addresses color invariance enhancement of intensity-based descriptors. Then it discusses the geometric compensation methods addressed in this PhD thesis. Only projective transformations due to projections on planar surfaces are in the scope of this work. Therefore, homography compensation methods, as well as an extension for multiple-homographies are studied in this chapter. Moreover, a method for spatial-temporal compensation is presented at the end of the chapter. Chapter 4 presents three test datasets deliberately prepared in this work to evaluate different algorithms in the context of projector-camera systems. The first dataset is composed of synthetic images with color and geometric distortions, while the second one

consists of real-world static projections obtained using a projector-camera system. The last dataset comprises video projections acquired in a dynamic scenario. Chapter 5 details the applications of the presented datasets which includes color invariant descriptor evaluation, homography compensation quality assessment both in the static and dynamic scenarios, and object recognition from acquired projections. Finally, chapter 6, devoted to algorithms implementations, describes a real-time compensation framework built with the help of GPU parallel optimization. This framework is evaluated on real-world projections in terms of compensation quality and execution time.

Chapter 2

Related works: Projector-Camera Systems

This chapter reviews the background and research works related to projector-camera systems and Spatial Augmented Reality (SAR). In the first part the latest SAR applications are presented to demonstrate the relevance of this field and particularly the importance of projector-camera systems for such applications. Next, the photometric model of the projector-camera system is provided along with some common issues that arise when using such systems. In the second part of the chapter an overview of state-of-the-art works for geometric compensation is presented. Active, passive, and hybrid approaches are considered.

2.1 Spatial Augmented Reality applications

Projector-Camera systems are mainly used in different applications for Spatial Augmented Reality where they play an important role in performing image registration and compensation, one of the key problems. Over the last ten years a lot of SAR installations have been shown in many different fields from social interaction, games and historical and cultural heritage to industrial prototyping and medical applications. Most prevalent works are briefly described below in this section.

Cultural Heritage : Museums, Historical buildings. SAR can be used in museography to ensure interactive visits when projections are done directly on paintings or another exhibits. It makes the viewing process more striking and interactive. O. Bimber *et. al.* [11] presented an artistic process in which pictorial atrworks were used as projection screens for further augmentation (Figure 2.1.(a)). B. Ridel *et. al.* [100] developed a system that allows augmenting archaeological artifacts by projecting 3D visualization to highlight their fea-

tures (Figure 2.1.(b)). There are other applications that perform projections inside [35, 84] and outside buildings and historical sites [54, 84]. Some of them are illustrated in Figures 2.1.(c)-(d).

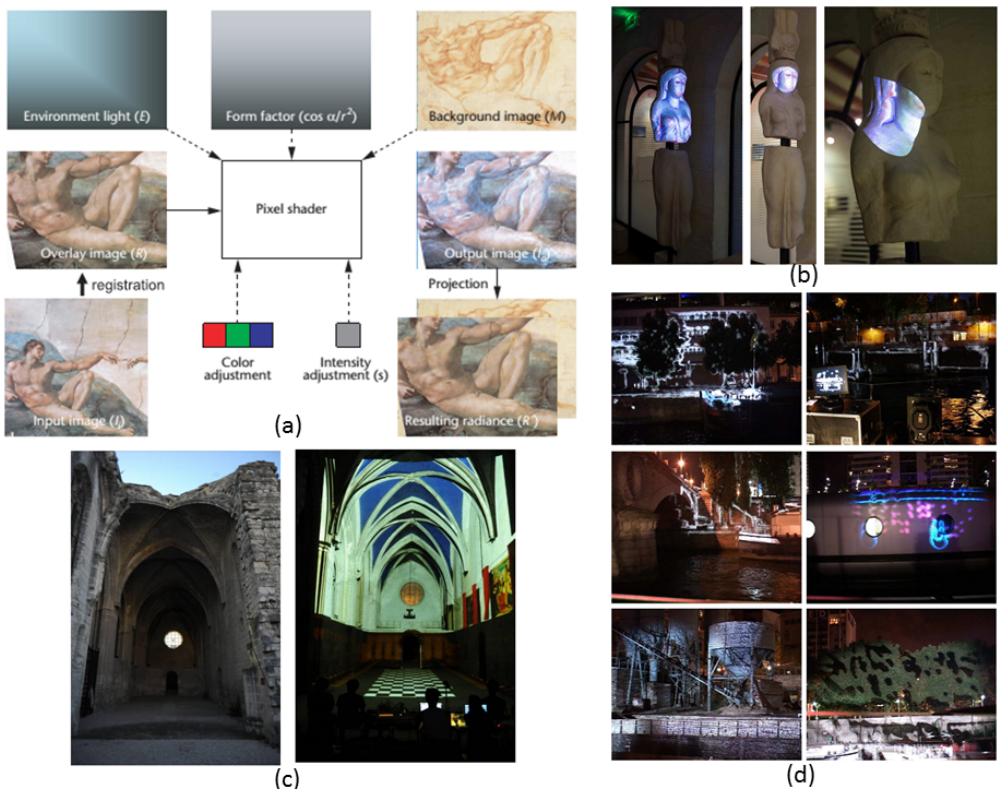


Fig. 2.1 Examples of SAR in cultural heritage. (a) - The process of superimposing a real drawing with a projected image [11]; (b) Virtual projection on the Isis bust [100]; (c) - church nave in its current state and virtually augmented with a historical projection [35]; (d) - several visual effects projected from a boat on the river bank and its surroundings [54].

Industry (prototyping). There have been many works that apply SAR for interactive rapid prototyping. The design process is simplified by projecting various product configurations on a physical form to simulate its appearance without the need of implementing the full model. Particularly, SAR systems have gained a lot of success in automotive industry to simulate location of gears on the board [78, 99] (see Figure 2.2).

Medicine. SAR, used to assist in image-guided surgery, projects images from MRI [1], computed tomography or other different sources, directly onto the patient body [43, 48, 92, 123] (Figure 2.3). It avoids the surgeon to look at another screen while performing the operation. Since the use of SAR techniques for medical operations require a very high fidelity, their applications were limited to visual training simulators.

[1] Magnetic Resonance Imaging

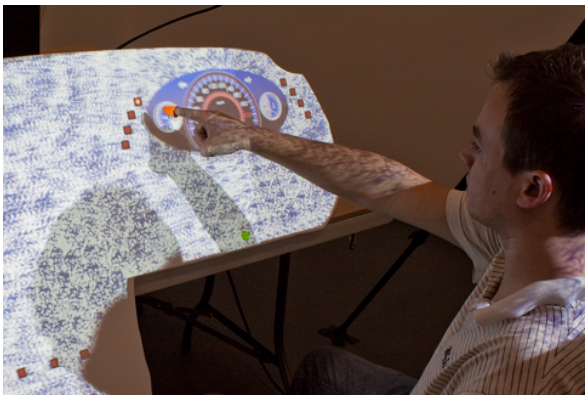
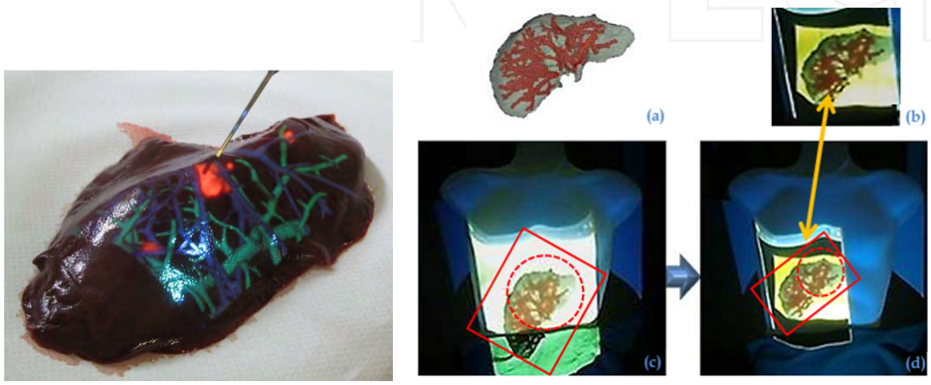


Fig. 2.2 A virtual control panel made by means of SAR system that projects onto a white panel [99].



(a) (b)

Fig. 2.3 Examples of SAR use for medical applications. (a) - projection of liver vessels, tumors, and resection planes on pig liver tissue [43]; (b) - Projected image of liver model and its correction [123].

Games and social interaction. Interactive spatial projections have long been used to create gaming experience in the real world. The system can detect and track physical objects used in the game [7] (Figure 2.4.(a)). D. Michelsen and S. Björk presented an immersive horror game *The Rooms* by means of SAR technologies [79]. D. Willis *et. al.* introduced the *MotionBeam* metaphor that uses handheld projector movements to interact with virtual and real objects [125] (Figure 2.4.(b)). In [22], the authors presented and detailed three installations BlogWall, MediaMe and Shared Design Space, intended to create social events, to improve the exchange of artistic capabilities in poetry, relationship between people and media, creativity and digital entertainment (Figure 2.4.(c)). Recently, Microsoft Research presented IllumiRoom system [55] that augments the area surrounding a television screen with projected visualizations to enhance the living room entertainment experience (Figure 2.4.(d)).

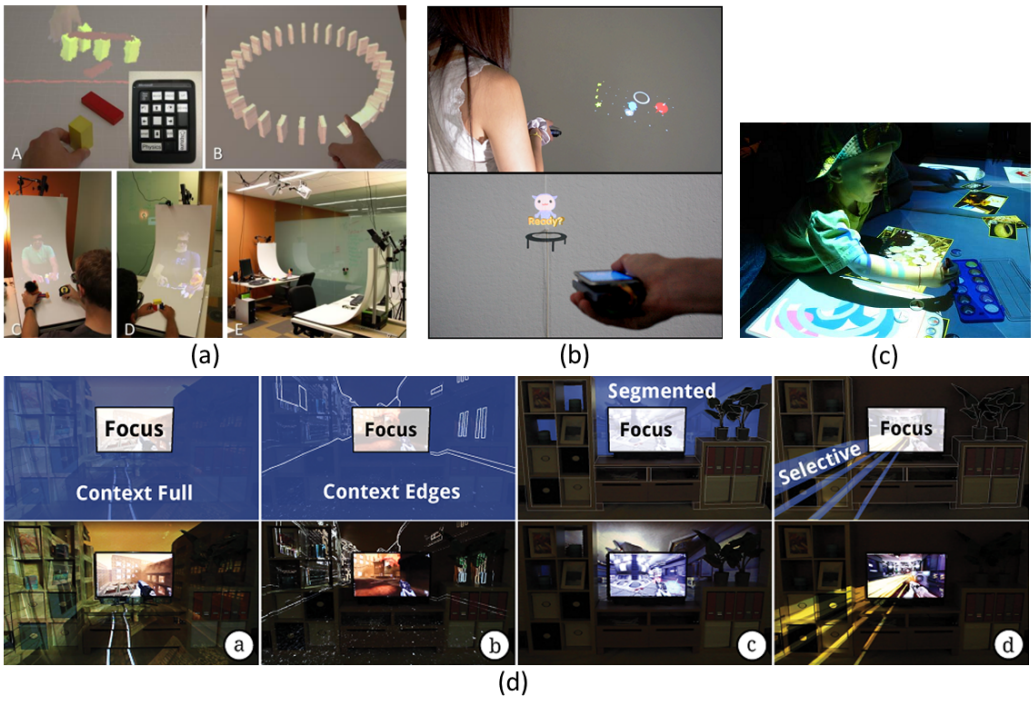


Fig. 2.4 SAR games and social interaction setups: (a) - *MirageTable*, interactive tabletop SAR setup [7]; (b) - handheld projector based interaction between real and virtual objects [125]; (c) - a collaborative SAR tabletop environment [22]; (d) - *IllumiRoom*, a system that augments the area surrounding a television with projected visualizations [55].

Collaborative work. SAR environment can serve as an efficient tool for collaborative work. F. Bérard in his work presented The *Magic Table* [10], an augmented whiteboard surface for creative meetings (Figure 2.5.(a)). Several users, with the help of tokens, can make virtual notes on the whiteboard, that the system digitizes and converts into a convenient format. *Mano-a-Mano* from Microsoft Research [8] is a SAR system that supports multi-view projection of virtual objects and, thus, allows users to interact with the objects and each other (Figure 2.5.(b)).

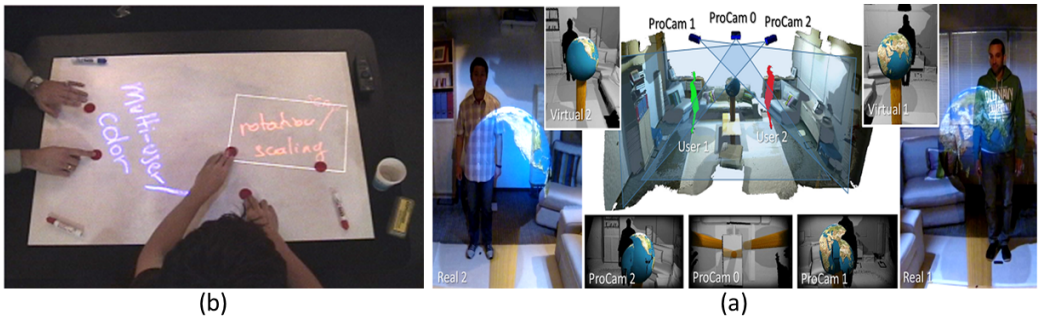


Fig. 2.5 Collaborative SAR environments: (a) - a SAR installation where several users can make virtual notes that are then converted and stored [10]; (b) - *Mano-a-Mano* environment that allows several users to interact with real and 3D virtual objects [8].

2.2 Projector-Camera systems

The term of "projector-camera system" encompasses any setup that employs controllable light-source. The projection part of the system typically includes one or several light projectors of different technologies. Nowadays, tiny projectors, called pico-projectors become more and more popular due to their small sizes. Sensing devices, used to capture projections, range from simple photo-sensors to high-resolution wide-field-of-view cameras.

Projection technologies. There exist several main projector technologies. The first one, *Digital Light Processing* or *DLP*, makes projections by means of tiny mirrors spread out on a semiconductor chip, called Digital Micromirror Device (DMD). Each mirror represents one or several pixels by reflecting light towards the screen (the pixel is "on") or away (the pixel is "off"). The number of mirrors, therefore, corresponds to the resolution of the projected image.

The second technology, *Liquid-crystal display* or *LCD*, makes use of three liquid crystal panels to form a projection. Light emitted by a metal-halide lamp passes through a prism to get three separate channels, which are then sent to the dedicated liquid crystal panels. Finally, all the three light beams are projected together on the screen.

The last common projection technology, called *Liquid Crystal on Silicon* or *LCoS*, represents a hybrid between LCD and DLP. The light is reflected by mirrors like in DLP, but blocked by liquid crystal like LCDs.

There exist other technologies such as laser video projector, however they are rarely used due to high production cost and technological issues such as projector cooling.

Most available pico-projectors are produced using DLP technology, however, there are commercially available LED, LCoS and even laser projectors.

Camera sensors. Most camera sensors are manufactured using one of the two prevalent technologies: *CMOS* or *CCD*. In a *CMOS* (Complementary Metal Oxide Semiconductor) sensor chage-to-voltage conversion is done individually for each pixel employing amplifiers, noise-correction and digitalization circuits to output digital bits. On the countrary, *CCD* sensors (Charge Coupled Device) gather each pixel's charge and perform conversion once for all pixels. CCDs require special manufacturing to transmit pixels' chargers accross the chip without distortions. This makes these sensors more expensive, compared with CMOSs, but with a better quality in terms of light sensitivity, amount of image noise and so forth. For their low price, CMOS sensors are often used in mobile devices.

2.2.1 ProCam system issues

There are several difficulties one faces when using a ProCam system in a SAR application. Technically, the projector and the camera should be connected and synchronized. Because each device introduces delays, the overall performance of the ProCam system may slow down. ProCam portability remains an important issue in applications that require projection mobility. With the development of pico-projectors, the system dimensions diminish but other problems persist such as limited battery life and projector power.

Device calibration is an important problem from the scientific point of view that involves estimating camera and projector parameters both with respect to the projection geometry and color. If the estimated parameters do not change over time the calibration process can be done off-line. Geometric calibration involves the estimation of the intrinsic parameters of both camera and projector, and the extrinsic parameters between the two devices. This can typically be solved through Structured Light techniques described later in 2.3. Photometric calibration tries to estimate the response functions of both the camera and projector, and to compute mapping from the reference projected to acquired colors. This is a complicated task because of various components such as camera sensor noise and blur, limited projector gamut, and saturation. Another problem lies in the frequency differences between the projection and acquisition devices. As a result, in the captured images there may be some artifacts present. The following section discusses the photometric model of the projector-camera system.

2.2.2 ProCam photometric model

In classical computer vision applications, color changes occur mainly due to illumination variations, viewpoint changes, and the color distortions introduced by the acquisition devices. All these conditions have been extensively studied and as a result several color models have appeared [37, 38]. Besides the above-mentioned factors, in projector-camera systems the color changes between the perfect projected image I and the captured projection C are due to several successive phenomena: the biased emission of I by the projector, the reflection of the beam from a complex and possibly colorful surface, and the response of the camera when acquiring the beam emitted by the surface. Consequently, color changes are not only due to the illumination but also to the emission spectrum of the projector, the reflectance properties of the surface and the camera sensitivity curves. Commercially available acquisition and projection devices that are most suitable for most SAR applications introduce high distortions to the system. Camera color distortions and sensor non-linearity,

as well as non-linear reduced projection gamut significantly complicate the compensation process. Not to mention additional phenomena such as inter-reflection on the projection screen between neighboring pixels, blur, noise and moiré effects...

The color captured by a camera results from the integration along the visible spectrum wavelengths of several functions which are difficult to compute: the spectrum of lighting sources, the reflectance of the surfaces, and the camera response function for each channel. Unfortunately the physical equation is too difficult to be used directly for color change modeling, unless some simplifying assumptions are made, as in Shafer dichromatic model [110]. In a projector-camera system the light emitted by the video-projector represents an additional source of lighting that needs to be accounted for.

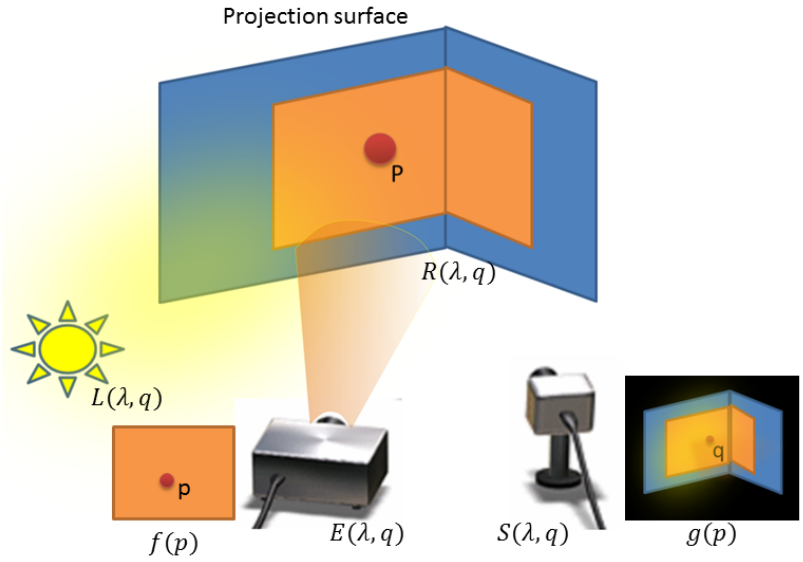


Fig. 2.6 Acquisition model in a projector-camera system

First, lets us consider the acquisition model for a camera and a projector-camera system. The following notation is used hereafter: let P be a small neighborhood around a physical point in the scene (real world) lit by a projector which produces an energy depending on its input image at a pixel p. This initial image is noted $f_k(p)$, where k is the color channel R, G, or B. The viewed color at pixel q in the acquired image, noted $g_k(q)$, is produced by the integration of the following components along the wavelengths λ of:

1. the illuminant spectrum $\mathscr{E}(\lambda, q)$, emitted by the projector plus the ambient illumination $\mathscr{L}(\lambda, q)$, if present in the scene,

2. the surface reflectance $\mathscr{R}(\lambda, q)$ which defines the proportion of incident light reflected by the surface for each λ;

3. the spectral camera sensor sensitivity $\mathscr{S}_k(\lambda)$ for each channel k.

Then, the captured intensity in channel k is defined as:

$$g_k(q) = \int_\lambda \mathscr{E}(\lambda, q)\mathscr{R}(\lambda, q)\mathscr{S}_k(\lambda)d\lambda \tag{2.1}$$

Note that p is the location of physical point P neighborhood in the projected image f_k, whereas q corresponds to the location of P in the acquired distorted image g_k. Figure 2.6 illustrates the notation.

2.3 Structured Light methods

Structured Light is a technique that extracts 3D surface shape by projecting a known pixel pattern on a scene. The camera, coupled with the projector, captures this pattern. 3D surface information can be found by computing correspondences between the projected and acquired patterns. In a general sense, the approach is not restricted to the visible spectrum and there exist imperceptible structured light that works, for example, in the infrared wavelength. However, such methods are out of the thesis scope and, consequently, are not addressed in this chapter because of the strict limitations on the maximum work distance and the presence of daylight that interferes IR emitters.

Structured-light systems are capable of handling well the reconstruction problems for untextured surfaces [103], as they superimpose the surface texture representation with the projected pattern in the captured scene. It results in an increase of the density of correspondences and, therefore, a more robust and precise reconstruction [5, 58]. However, one inconvenience may arise when using a structured light system, which is the need for device recalibration for dynamic compensation. This procedure requires projection of many additional images with calibration patterns or embedding these visible patterns directly in the projected content [97, 98].

Figure 2.7 illustrates a typical structured-light system that includes a camera and a projector that form the hardware part of the system. In order to extract 3D information from projected and acquired patterns, they are coded in a specific way. Nowadays there exist plenty of coding strategies that define the way to interpret the imaged pattern and to find the correspondences. According to the classification of J. Salvi *et. al.* [104], pattern projection techniques comprise discrete and continuous coding methods.

Discrete coding methods are based on *spatial* or *temporal* multiplexing. Spatial coding methods project spatial color or intensity patterns and then analyze groups of pixels spatially located around each pixel. The methods of this group are De Bruijn patterns [94, 95],

non-formal coding [36, 40, 53, 58, 62, 75], and M-arrays [3, 46, 85, 86, 93]. These methods are typically sensitive to noise and variations in illumination. *Time multiplexing* methods construct codewords by successively projecting several pattern images. Such techniques require several pattern projections, where the number of the images depends on the desired spatial reconstruction precision. Although Time multiplexing coding in general yields robust and precise reconstruction, it is only suitable for static scenes because of the multiple frames required for obtaining the 3D data.

Continuous coding methods represent projected patterns as continuous variations in contrast or in color. Periodic patterns are used in time-multiplexing phase-shifting methods (based on several projected images [104, 113, 126]) and in frequency multiplexing (decoding is performed in the frequency domain using Fourier or Wavelet transform [68, 74, 131]). Another continuous coding type, absolute patterns, is used in spatial grading methods [19, 114]. Because they represent the entire codeword by a single pixel value, the reconstruction resolution is defined by the resolution of the devices. Such methods are highly sensitive to different types of noise, and cannot address small surface variations because of their low resolution.

In the literature there have been some works that achieve a real-time projection compensation for dynamic surfaces by imperceptibly embedding patterns into the projected image sequence. H. Park *et. al.* [97] presented a projector-camera system which is capable of dealing with dynamic surfaces, however at the expense of succeeding to avoid pattern perception. As reported in the paper, the reasonable compensation quality could only be achieved with visible patterns in the projected content.

2.4 Passive geometric compensation methods

In contrast to Structured Light techniques, passive geometric compensation methods do not require projecting any additional patterns. They rely only on the projection content itself to match the reference image with the captured projection in order to estimate and compensate the transformation between the reference and projected images. Because it is not intrusive this approach is a good alternative to Structured Light methods when used in mobile SAR applications. The research presented in this thesis, investigates further this domain and proposes new algorithms for passive geometric compensation.

However, in a real system it is very difficult to achieve good compensation results. The passive methods rely on the definition of local features extracted from the matched images. If the reference image has a low spatial frequency, the transform estimation becomes very

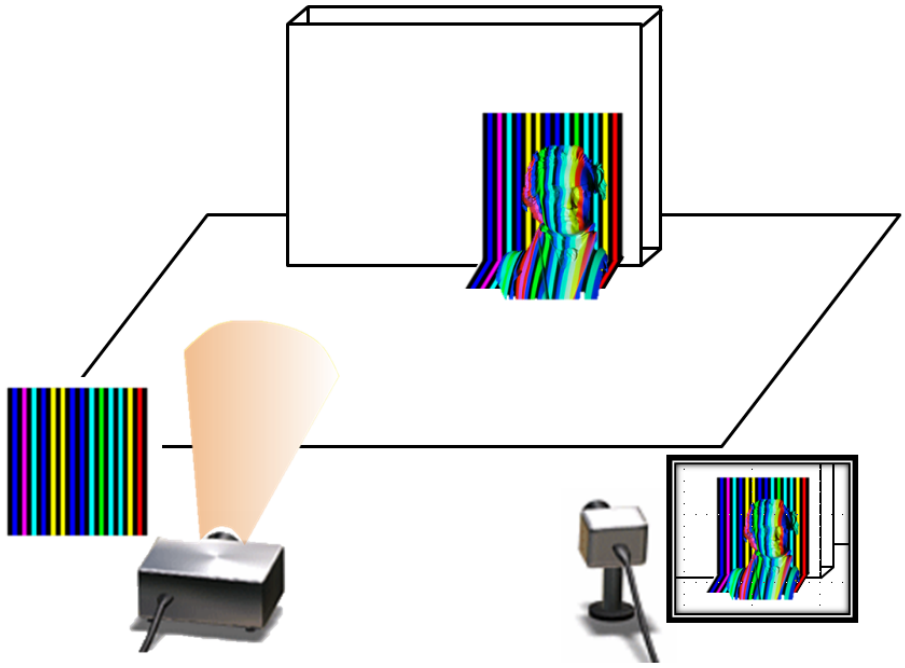


Fig. 2.7 A structured light system.

difficult due to the lack of extracted features. Besides, there are other different components that introduce noise into the system, for example the surface color or the scene illumination. For these reasons, only a few works have been presented up to now [31, 63, 127, 129]. R. Yang *et. al.* [129] in their approach estimate the transformation iteratively, starting from a rough estimate. At each frame they extract several features from the compared images and match them by means of a Kalman filter and template matching. However, the above-mentioned method cannot work with dynamic surfaces unless only small displacements occur. Moreover, the method is very sensitive to lighting changes and surface background color. T. Yamanaka *et. al.* [127] developed a method to perform geometric estimation on non-planar surfaces. Their method is based on the fundamental matrix for the camera and projector that is computed during the off-line calibration process. The captured projections and the reference images are matched on epipolar lines and then the surface shape is computed by fitting B-spline surfaces on the matching epipolar curves. The considered projection surface in this case should be made of a number of B-spline curves.

A new approach for passive projector-camera matching was shown by M.-A. Drouin *et. al.* [31]. To estimate and compensate geometric distortions their system performs several steps. Firstly, the devices are calibrated off-line in order to estimate intrinsic parameters. This process has to be done once and does not depend on the projection surface. The devices

are placed so that their epipolar lines are horizontally aligned. Secondly, a binary motion field is computed for each frame from the projected and captured video stream by means of a background subtraction algorithm. Active pixels with high values are quantized. At the next step, the system matches active points from the projected and captured videos. To that end, for each spatial point quantized values are concatenated over the whole image sequence I_t. The correspondences are found by a dynamic programming. Finally, the surface equations are obtained by fitting the computed sparse 3D points. RANSAC is used to cope with outliers. RANSAC[2] as well as other algorithms for surface point fitting are discussed in Section 3.3.1 of this chapter.

T. Kooi *et. al.* [63] propose a calibration-free method based on color-invariant feature matching and probabilistic outlier filtering. Among several tested color invariant transforms, C-color invariant, a derivation from the model proposed by Geusebroek *et. al.* [44], is shown to be the best performing one. SIFT feature matching, coupled with probabilistic RANSAC (MSAC) [116], is used for geometric surface estimation.

S. Zollmann *et. al.* [138] present a hybrid approach that makes use of both active and passive estimation techniques. The first one, a structured light, is used for the first precise estimation of the projection surface. Then, if the dynamics of the scene does not introduce complex geometric distortions, a low-weight optical flow is exploited to estimate the relative transformation between consecutive image frames.

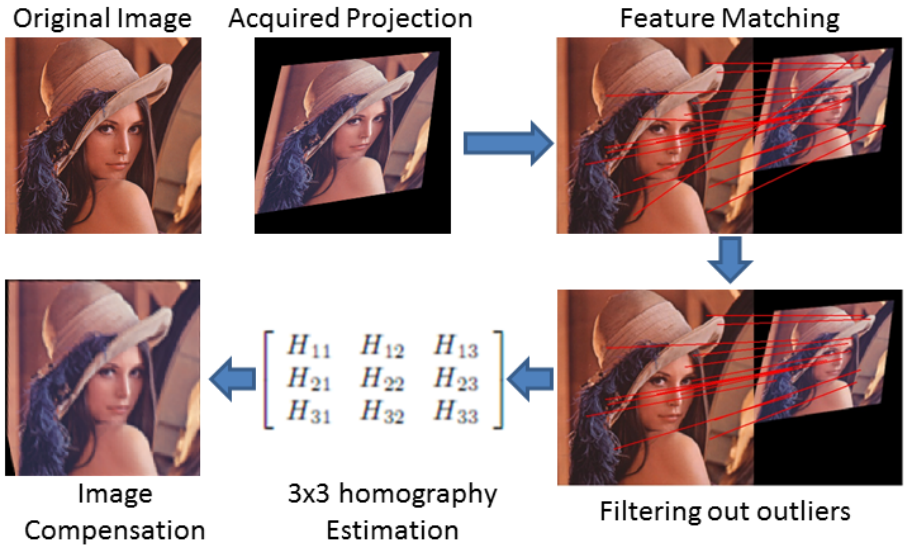


Fig. 2.8 Feature-based homography compensation.

Inspired by the above-presented works, in my PhD thesis I continue to investigate this

[2]RANdom SAmple Consensus

problem of unintrusive geometric compensation that has received little attention until now. My approach bases on color invariant feature matching performed for each pair of images, the original image and its projections. This will be explained in more detail in the following chapter. The estimation of the homography transformation can be done through RANSAC covered in 3.3. The simple illustration of the compensation is presented in Figure 2.8. Finally, Optical Flow tracking is employed in 3.3.4 to reduce the computation time of the compensation framework.

2.5 Conclusions

In the first part of the current chapter state-of-the-art works in SAR were reviewed. Most of the works make use of projector-camera system to handle video projections on everyday surfaces. Geometric and color compensations are the most difficult problems which hinder the systems from smart projections. The second part of the chapter presented a review of the methods that approach the problem of geometric correction. Structured-light techniques while being *de-facto* the standard for surface recovery, have a number of disadvantages, like the need to project several images with calibration patterns or to embed the patterns in the projected content in a visible way. For these reasons in this PhD work it was decided to investigate more the problem of unintrusive geometric compensation that relies only on the projected content. Therefore, the first part of the next section discusses the Local feature matching, chosen as the basis for geometric compensation. Then, color-invariance extension to feature matching is presented. In the second part addresses the problem of geometric compensation of projective transformations in ProCam systems.

Chapter 3

Color Invariant Feature Matching And Geometric Compensation

As mentioned in chapter 2, the basis of my research work is local feature matching, that is proven to be robust to local deformations. Furthermore, color invariance is applied to the matched images, because photometric differences between the reference and projected images are typically high to be handled by the traditional intensity-based methods. Color invariance, therefore, aims at providing more robustness to the feature matching process.

This chapter is dedicated to three main subjects. Firstly, it considers the classical feature matching problem describing the most relevant algorithms for keypoint detection, descriptor computation, and feature matching. Secondly, this chapter relates to the extension of intensity feature matching approach for the purpose of color invariance. After motivating the use of this approach, I successively present several most common color change models and color invariant transforms. Histogram Equalization (HE), a technique for color invariant feature description, is introduced along with its improved version that implies local computation. Finally, the problem of geometric projection compensation is addressed in this chapter by the example of single- and double-homography compensations. Furthermore, a combination of Feature Matching and Optical Flow is presented to reduce the execution time of the compensation.

3.1 Feature Matching

This first section, dedicated to the problem of feature matching, gives an overview of different algorithms for keypoint extraction and description.

3.1.1 Feature Matching problem

Local salient regions have long been used to find correspondences between images. Contrary to larger regions or segments, they are likely to be retained in the images in the presence of different geometric transformations that images undergo, and partial occlusions. For this reason, most low level computer vision techniques are based on local regions that are extracted from the images and further used, for example, for image matching, object recognition, or tracking. Since in real applications matched images may undergo various geometric and photometric changes, much effort has been made to improve such characteristics as distinctiveness and invariance of the extracted local regions.

The most fundamental works in the field are the SIFT descriptor introduced by D. Lowe [71], and its speeded-up version SURF [6] presented by H. Bay *et. al.*. Both works describe the methods to extract features from images and describe them, as well as the way to efficiently perform their matching. For my work I chose the SURF because of its lower complexity, i.e. faster execution, over the SIFT [73]. However, the developed approach for geometric compensation can be also used with different other feature matching algorithms.

To achieve geometric compensation this PhD work considers a classical feature matching problem that can be described by the following pipeline:

1. At the first step local feature points (also called keypoints, interesting points or regions) are extracted. The extraction algorithms seek for the regions that retain enough information to be further properly identified.

2. Once all features have been extracted from the image, at the next step, each feature region is assigned a vector of values (called descriptor) that should uniquely describe it. There are two problems that a descriptor should address: (a) the region description should be invariant (or robust) enough to allow correct matching of the same region under different transformations or distortions that are likely to occur in the sequence: blur, noise, contrast/lighting changes, geometric changes (scale, homography); (b) the descriptor should be distinctive enough so that two regions of the same object can be matched with a high probability and *vice versa*, a region of one object should not be matched with a region of another object. Unfortunately, the gain in invariance is usually made at the price of a lower distinctiveness or of a higher computation time. One of the difficulties is then to find the good trade-off between these parameters.

3. At the next step a matching process is performed. Given two images with described local salient regions, it consists in finding for each identified local region in the first

image a correspondence with another local region in the second one using some distance function or a similarity measure between the descriptors such as Euclidean, Hamming, or other distance. Besides, among all computed matches weak ones are found and discarded. One of the strategies, explained in [71], rejects a match if a keypoint is close to more than one keypoint in the second image, *i.e.* the ratio between the two smallest distances is greater than a threshold 0.8.

4. Finally, to remove outliers from the matching results, a filtering process can be exploited. This process imposes some temporal, spatial or geometrical restrictions which depend on the application in which feature matching is performed. For instance, in case of planar transform estimation addressed in this thesis, the matches are rejected as outliers if they lie out of the estimated surface.

3.1.2 Feature Point extraction

Feature point extraction (or detection) consists in searching for the points of interest located at local extrema, typically, in spatial and spatial-scale domains.

Keypoint detection in spatial domain

The keypoints detection in the spatial domain can be made from the intensity derivatives or by detecting corners in a binary image. The *Harris Detector* [49] and the *Hessian Detector* [80] detect keypoints at image locations that have strong derivatives in two orthogonal directions. They provide invariance to affine geometric transformations. The Harris Detector searches for points where the second-moment matrix C has two large eigen values. The matrix C can be computed from the first derivatives in a window, weighted by a Gaussian kernel. In a similar idea, the Hessian detector uses the matrix of second derivatives and detects keypoints in the image where the determinant of the matrix represents a local maximum. *Non-maximum suppression* is applied on the image of the determinant values. Finally only those locations are kept that have responses greater than a pre-defined threshold.

Binary Detectors

Binary corner detectors consider binary relationships between the center and each pixel at the periphery of the circular area around the keypoint. SUSAN keypoint detector [112], one of the most famous methods among the binary detectors, searches for corners in the image by considering the radial region R around each point p

and compares the size of the areas with the higher ($\sum I(p_i) > I(p) + T$) or lower ($\sum I(p_i) < I(p) - T$) intensity (see Figure 3.1). Here, $I(\dot{})$ is the intensity of the pixel, p_i are the pixels in the circular area around the keypoint p, and T is the threshold. FAST [101, 105], an accelerated version of SUSAN corner detector, considers a circle of $N = 16$ pixels around each point. The point p is classified as a corner if there exist $n = 12$ pixels p_i such that $I(p_i) > I(p) + T$ or $I(p_i) < I(p) - T$ (see Figure 3.2).

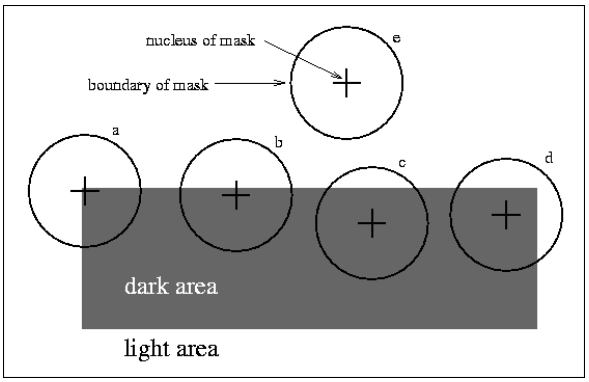


Fig. 3.1 SUSAN detector: a region can be classified as flat zone (a,c,e), as an edge (d) or as a corner (b). Image taken from [112]

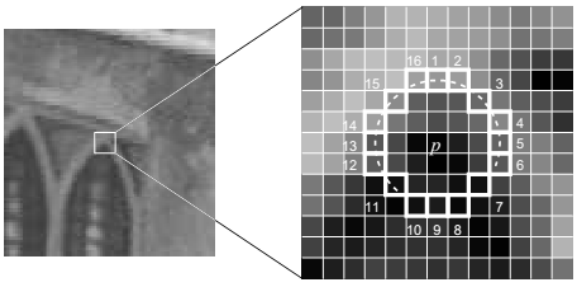


Fig. 3.2 12-point FAST detector: Point p is detected as a corner if 12 contiguous pixels around p are brighter than p by more than the threshold. Image taken from [101]

All the works presented before belong to the group of corner detectors. Such methods cannot detect features at different scales because they work only in the image space. An alternative group of methods has appeared that searches for keypoints in the image scale-space. The most important works are presented below.

Keypoint detection in spatial-scale domain

It was shown in [80] that finding maxima and minima of *Laplacian of Gaussian* (LoG), that can be written as $\sigma^2\nabla^2 G$, produces the most stable image features compared to most of other image functions. T. Lindeberg in [69] showed that LoG can be

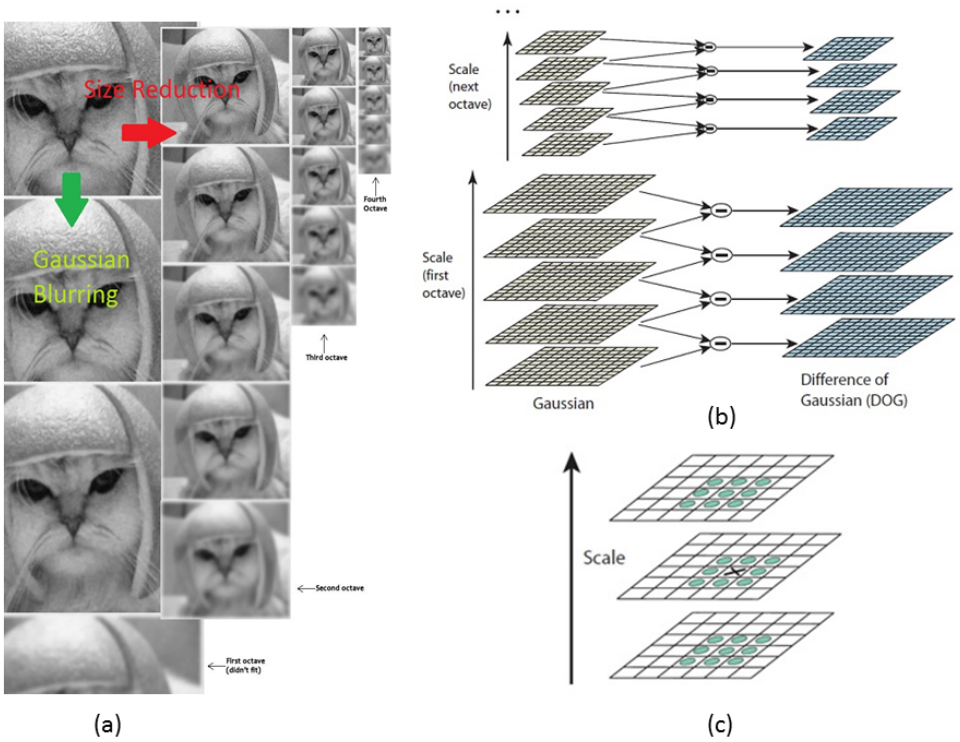


Fig. 3.3 SIFT detector: (a) - Gaussian scale space, constructed by progressively blurring and resizing the initial image; (b) - Difference of Gaussian is computed and salient points are detected by non-maximum suppression (c). Images (b) and (c) are taken from [71]

approximated by the Difference of Gaussians (DoG) of nearby scales which makes the keypoint detection more efficient. This method D. Lowe later used in SIFT [70] to extract salient points (see Figure 3.3).

More recently, K. Mikolajczyk and C. Schmid [81] proposed *Hessian-Laplace* and *Harris-Laplace* detectors that consist in using the Harris and the Hessian detectors respectively to detect keypoint location and to select the scale. *Fast-Hessian*, used in SURF [6], is a fast approximation of the Hessian-Laplace detector. It makes use of integral images to reduce the computation time. Instead of convolving the image with the Gaussian second order derivative, a more simple box-filter approximation is used. Integral image format ensures a constant integration complexity regardless of the integration area (see Figure 3.4).

Detection of region boundaries

MSER (Maximally stable extremal regions) [76] is a keypoint detector that finds areas that are stable with respect to changes of intensity thresholds (see Figure 3.5). The

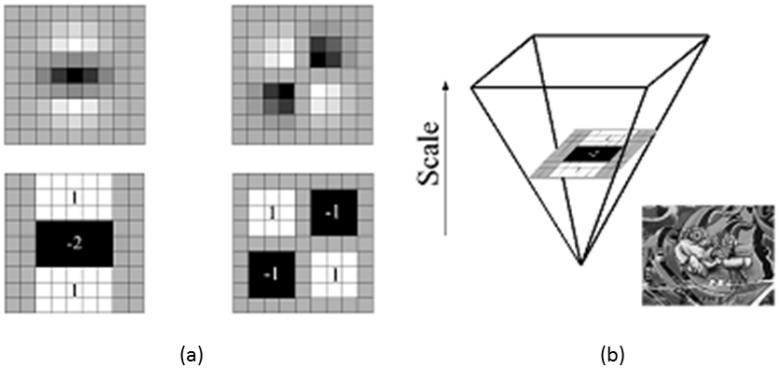


Fig. 3.4 SURF detector: Gaussian second order partial derivatives ((a) top row) are approximated by box filters ((a) bottom row). To build a scale-space pyramid the box filters are upscaled and convolved with the original image (Image (b)). Images taken from [6]

region is considered as stable if its size does not change under intensity variations. **PCBR** (The Principal Curvature-based regions) detector [29] extracts stable regions within the multiscale principal curvature image (see Figure 3.6). This method is robust to local intensity variations within regions by focusing on region boundaries rather than on the appearance of region interiors.

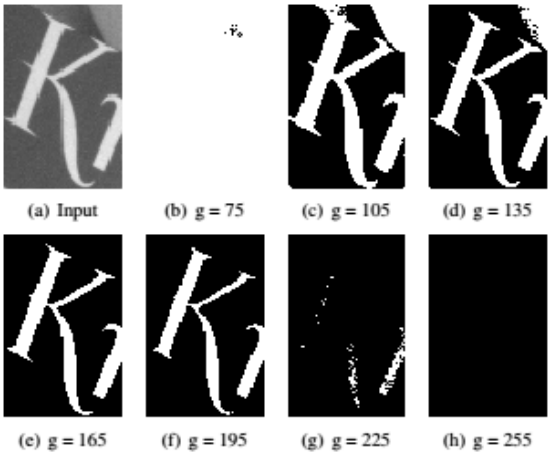


Fig. 3.5 MSER detector: The letter *K* is detected as MSER because the size of this region does not change under different thresholds applied to the image. Image taken from [30]

When a projector-camera system is used the information of the projected image is characterized by large geometric distortions. In this instance, SIFT/SURF detectors appear to be the best candidates to extract keypoints for feature matching. The keypoints extracted by these methods are robust not only against geometric distortions

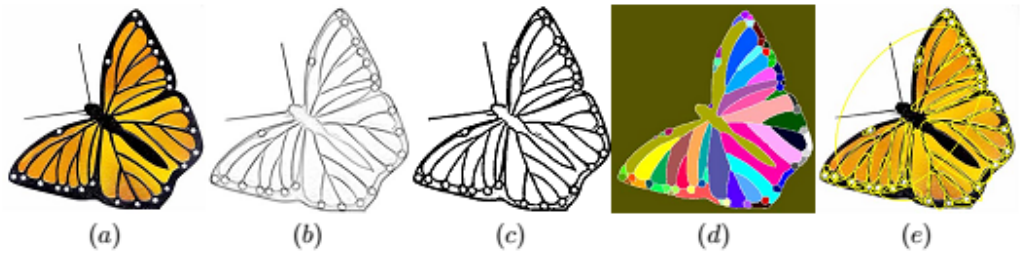


Fig. 3.6 PCBR detector: (a) - initial image, (b) - detected principal curvature structures; "cleaned" binary image of principal curvatures; (d) watershed regions and (e) detected regions represented by eclipses. Image taken from [29]

and partial occlusions, but also against contrast changes. The contributions of this thesis on the robustness of feature point detection and matching against color variations, even though they are based on SIFT/SURF detectors, can however be generalized to other feature detectors.

3.1.3 Descriptor computation

Each detected keypoint P_i is assigned a float array $D_i = [d_{i1}, \dots d_{in}] \in \mathscr{R}^n$ that characterizes the region around this point. This vector, called feature descriptor, is used to compare this point with the others by computing a similarity metrics $F(D_i, D_j)$ for each pair of features P_i and P_j. Typically, the Euclidean distance between two vectors is used as the matching metrics. As explained in section 3.1, the descriptor should be discriminative and robust in order to ensure a good matching quality. Moreover, the length of the descriptor vector represents an important criterion that affects the computation matching speed.

There exist plenty of feature descriptors that can be subdivided into several groups: pixel blocks, histograms and histograms of gradients (HoG), wavelets and binary descriptors. Among them the most prevalent are HoG and wavelets used in SIFT and SURF.

Pixel blocks. Methods in this category describe keypoints as a set of pixel intensities located in the keypoint neighborhood. This descriptor does not tolerate geometric changes between the pixel blocks of the keypoints in the original and transformed images to be matched.

Histograms and histograms of gradients. SIFT[71] can be computed as follows. The area around a feature point, weighted by a Gaussian window, is spatially subdi-

vided into several cells and for each cell a histogram of gradient orientation is computed and weighted by the gradient magnitudes. Trilinear interpolation was proposed in order to reduce all boundary effects. Each bin entry of a cell is multiplied by a weight of $1-d$, where d is the distance of the point sample to the bin center. d is measured in units of the histogram bin spacing. The descriptor vector is formed from orientation histogram values of each cell. Typically, the algorithm uses 8 orientations and the grid contains 4×4 cells, which results in a descriptor vector of the length $8 \times 4 \times 4 = 128$. Finally, to compensate the effect of illumination changes, the vector is normalized to unit length. Non-linear illumination changes due to camera saturation are reduced by thresholding each value in the unit vector to be no larger than 0.2 and then by normalizing the vector again. The SIFT descriptor computation process is illustrated in Figure 3.7.

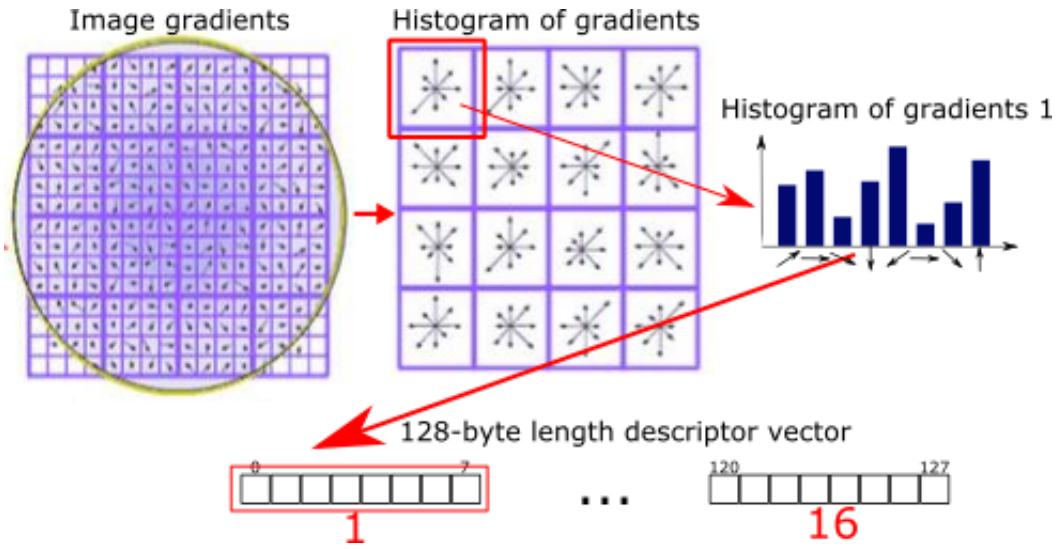


Fig. 3.7 SIFT descriptor.

Histograms of Oriented Gradients (HoG) [26] is a variation of SIFT descriptors that operates with larger image patches and uses more cells grouped in descriptor blocks that can overlap so that a cell can belong to more than one block. Each histogram is normalized to reduce the effect of illumination changes. Unlike SIFT, the HoG descriptor is generally computed for dense keypoints.

Further, many extensions have shown up that are intended to reduce the computation time and to improve the robustness against certain types of transformations. ASIFT [130] extends SIFT by simulating the camera axis orientation and, thus, making the SIFT descriptor robust to affine transformations. GSIFT [87] adds a global texture

vector to the SIFT descriptor in order to discriminate features with similar local appearance. PCA-SIFT [59] uses PCA to replace the gradient histogram method in SIFT making a new vector significantly smaller ($n = 20$ in the original work) than a standard SIFT vector. GLOH descriptor [82] first considers more spatial regions for the histograms, then high dimensionality of the descriptor vector is reduced to 64 through PCA. Therefore, because of dimensionality reduction the results descriptor becomes more descriptive as compared with other description vectors of the same size.

Binary descriptors. The interest of using such descriptors is to significantly reduce the time needed to compute and match keypoints. Indeed, the standard Euclidean distance, used to match most descriptors, can be replaced by time-efficient bitwise XOR. BRIEF (Binary Robust Independent Elementary Features) descriptor, introduced in [16], is obtained by comparing 512 pairs of pixels after applying a Gaussian smoothing to reduce the noise sensitivity. The pixels positions are preselected randomly according to the Gaussian distribution around the patch center because BRIEF is not invariant to scale and rotation changes (see Figure 3.8 (a)). A. Rublee *et al.* [102] proposed the Oriented Fast and Rotated BRIEF (ORB) descriptor which is invariant to rotation and robust to noise. BRISK (Binary Robust Invariant Scalable Keypoints), proposed by S. Leutenegger *et al.* [66], provides both scale and rotation invariance. To compute the descriptor, sample points are positioned in concentric circles surrounding the feature, with each sample point representing a Gaussian blurring of its surrounding pixels. The pairs are divided in short-distance and long-distance subsets. The long-distance subset is used to determine the keypoint orientation while the short-distance subset is used the to build binary descriptor after rotating and scaling the sampling pattern (see Figure 3.8 (c)). This descriptor was inspired by another feature descriptor, called DAISY [115] (Fig3.8 (b)).

FREAK descriptor [90] (Figure 3.8 (d)) is an improvement of BRISK that uses 43 weighted Gaussians computed around the keypoint. The pattern structures were inspired by the retinal pattern in the eye. The average pixels overlap and are much more concentrated near the keypoint. A cascade of binary strings is computed by efficiently comparing image intensities over a retinal sampling pattern.

SURF [6], similarly to SIFT, splits the keypoint region into several cells but, instead of computing gradient orientations for each grid cell, computes Haar-wavelet responses along the axis directions d_x and d_y. For efficient computation, integral images, already computed at the detection step, are reused. The wavelet responses are summed up over

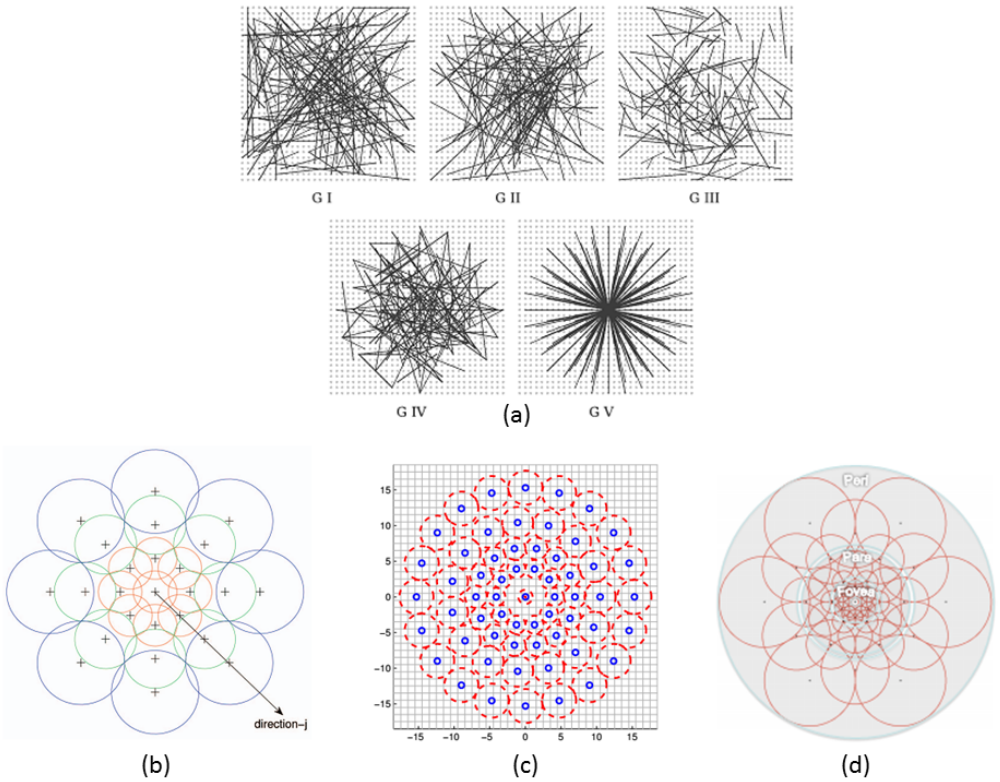


Fig. 3.8 Binary descriptors. BRIEF descriptor (a): different examples of random selection of test points (except the rightmost sampling). Image taken from [16]. DAISY descriptor (b): each circle area centered at sampled pixel locations is smoothed with the Gaussian kernel with the standard deviation proportional to the circle area. Image taken from [115]; BRISK descriptor (c): sampling with 60 points, where the circle radius corresponds to the standard deviation of the Gaussian kernel used for smoothing. Image taken from [66]; FREAK descriptor (d): sampling pattern similar to a retinal pattern. Each circle represents a receptive field, where the image is smoothed. Image taken from [2]

each cell which forms the first half of the descriptor vector. The sum of the absolute response values are also included in the descriptor in order to bring in information about the polarity of the intensity changes. Thus, for each of 4×4 cells, four values $(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ are used to form a descriptor of length 64. The computation process is depicted in Fig 3.9.

Dense-SIFT (DSIFT) is a method, widely used in computer vision, especially in image classification [12], that bypasses the keypoint detection procedure and computes SIFT descriptors densely and uniformly at each K-th pixel location. This reduces the computation time and ensures that enough points are extracted. Thus, some applications may benefit of such an approach. PHOW-SIFT, a variation of DSIFT when

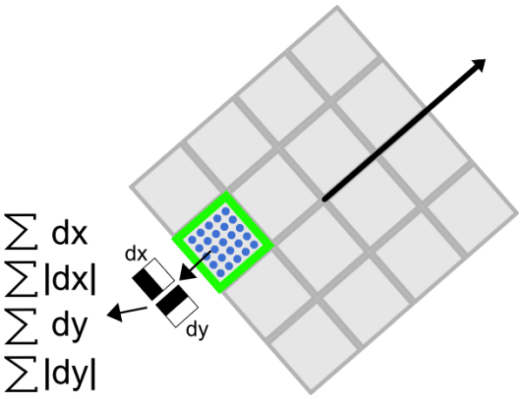


Fig. 3.9 SURF descriptor computation. The green square represents one of 16 subregions and blue points denote sample points at which wavelet responses are computed. The responses are computed with respect to the dominant orientation. Image taken from [33].

descriptors are computed at several resolution, is widely used for object recognition applications.

Color descriptors. All above-described methods work only with intensity (grayscale) images. A breakthrough has been made by adding color information to the descriptors in order to improve descriptor robustness to various illumination changes. The color extension of feature matching is discussed in the second part of this chapter.

In our work, the SURF descriptor [6] is used because it provides a good invariance to rotation and scale changes while performing faster than SIFT [71]. Moreover, SURF is less sensitive to noise.

3.1.4 Feature Matching

The Feature Matching process aims at finding one-to-one correspondences between keypoints extracted from a pair of matched images. There exist different strategies for doing that. One of the widely used approaches to match feature points, described in SIFT [71], performs all-vs-all comparison of two sets of keypoints from the matched images. To compare two descriptor vectors, the Euclidean distance is computed for each keypoint pair and the correspondences are established by the simple nearest neighbor search.

Errors in matching can be caused by many factors, among them noise, occlusions, or deformation of the objects in the matched images. To reduce the number of false matches, SIFT considers for each point the distance to the second closest point in the

matched image. The ratio of the distances to the first and the second closest points are examined. If it is greater than a threshold value t, then it means that there is only one corresponding point giving a small error and the matching is, therefore, unambiguous. Otherwise, in case of ambiguous matching, two or more correspondences that have small distances, are likely to be considered as outliers and discarded. Experimentally, a threshold value of 0.8 was found to be yielding the best results. However, in many cases, the described matching procedure cannot handle well faulty matches because of local distortions, whereas the filtering threshold is imposed globally. Techniques to impose additional constraints on the found matches and therefore to improve the matching quality are discussed in subsection 3.3.1. SURF takes over this approach, although it uses a different threshold value of 0.7.

Besides the Euclidean distance, some other functions may be used. To match binary descriptors, the Hamming distance is used which makes the matching process time-efficient. Some works make use of the Mahalanobis distance [134] that is the general case of the Euclidean metrics. Unlike the Euclidean distance, the Mahalanobis distance uses both the mean vector and the full covariance matrix which can be more efficient in some cases.

3.1.5 Performance evaluation

Feature detectors and descriptors have been extensively studied for targeting many different applications. Each application can take advantage of using one method or another and, thus, it is very difficult to define one single, generally the best, descriptor or detector. Both in the literature and in industrial implementations, SIFT and SURF have gained a lot of success and have been proven to produce stable results for a wide range of applications [50, 83]. In the present PhD work, the choice was given to SURF because of its better performance over SIFT [73].

However, the compensation approach developed in this thesis is not restricted to only one type of descriptors and whatever method can be applied. For instance, in one evaluation presented in this PhD work SIFT descriptor is used. Moreover, most works on image stitching, whose main task is to find correspondences between partly overlapping images, ofter use SIFT descriptors [14, 21, 133].

In terms of computation time, the leaders among feature matching methods are binary detectors and descriptors. However, as it was shown in some works [17, 50], in

general they are outperformed by SIFT and SURF which makes their use more advantageous. The drawback of high complexity can be compensated by modern optimization techniques such as GPU parallelization. More work on this topic is presented in chapter 6.

3.2 Color invariant Feature Matching

The algorithms described in the first part of this chapter are intended to work with grayscale intensity images discarding the color information. The descriptors computed this way are neither invariant, nor robust to most photometric changes in the matched images. In some applications, such as image stitching, the difference in color between the consecutive frames is not high, and thus the lack of robustness to color variations of the matching algorithm has little importance. However, in the case of projector-camera systems considered in this PhD work, photometric differences become an obstacle for a precise image matching. The rest of the chapter, therefore, considers a color invariant extension of feature matching algorithms intended to cope with the high color variability in the matched images.

Various recent works explore color feature matching. Basically, the approach consists in, first, applying a color invariance transform to the matched images. Keypoint detection can be performed either on intensity images or on color invariant image representations. Next, for each keypoint a descriptor is computed for each image channel by a conventional intensity algorithm. The final descriptor is, therefore, formed as a concatenation of two or more descriptor vectors corresponding to each color channel. Finally, feature matching is performed using longer descriptor vectors.

Inspired by the recent study of K. van de Sande *et. al.* [118], I commence by defining the color change models to approximate the photometric distortions which can occur in acquired projection images. It is followed by a review of existing color invariant transformations that may be used for descriptor computation. I conclude by presenting the Histogram Equalization (HE) method as a color invariance transform and, eventually, its local extension called Local Histogram Equalization (LHE).

3.2.1 Color change models

In order to define color invariant features, it is important to study and model photometric distortions that can occur in matched images due to different factors such as camera sensors, projector gamut or changing illumination conditions. Lets us now consider two color

change models to approximate the real physical equations described in chapter 2. These models are used to construct color descriptors and to validate their invariance properties through a series of synthetic experiments. The first model is the diagonal-offset model of illumination introduced by G. Finlayson *et. al.*[38] and used in color descriptor evaluations [118] and [108, 109]. The second model is the *gamma* color function that corresponds to most projector response functions.

Diagonal-offset model is chosen in this research work as a simple and commonly used equation that can model both the linear camera response function and different illuminations in the scene. If the system includes a video projector, the device, to be characterized by the above-mentioned model, has to have a linear response function.

The Diagonal-offset can be written as:

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix} \tag{3.1}$$

where (R_c, G_c, B_c) represent acquired colors after illumination variations, (R, G, B) are object colors under the canonical illuminant. The multiplicative parameters a, b, c represent contrast changes, and o_1, o_2, o_3 model the color shifts and were introduced to address the more difficult case of diffuse lighting.

It is important to mention that both models assume the R,G and B channels are independent. As it will described later in Section 3.2.4, the projector response can be considered as independent on the three channels.

Gamma model.

Most acquisition devices have a nonlinear response, which is the function that relates the irradiance of the scene to the corresponding image brightness. M. Grossberg and S. Nayar [47] have collected a database of camera response functions[1] and showed that they all have an exponential form. Likewise, many papers on projector-camera systems [27, 89] show that projectors have very similar non-linear response functions. Thus, in case linearization is not performed during the system calibration step, the gamma model can provide a good approximation of non-linear device responses both for camera sensor capture and projector emission:

$$\begin{pmatrix} R_c \\ G_c \\ B_c \end{pmatrix} = \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{pmatrix} \cdot \begin{pmatrix} R^{\gamma_1} \\ G^{\gamma_2} \\ B^{\gamma_3} \end{pmatrix} \tag{3.2}$$

[1]This database can be downloaded at http://www.cs.columbia.edu/CAVE

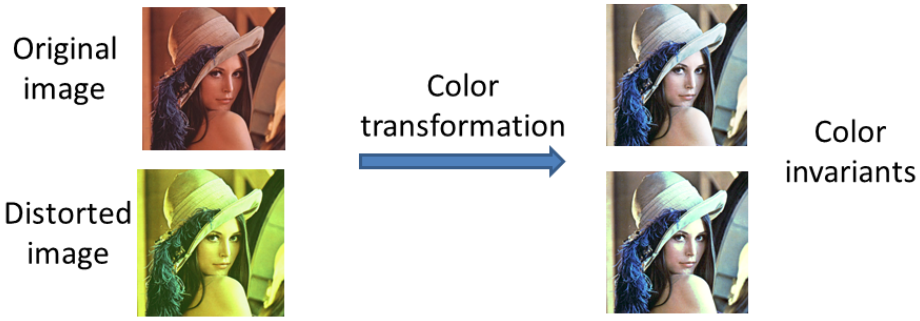


Fig. 3.10 Color Invariant transform applied to two images with different illumination.

where c indicates the color distortions that pixels undergo, and (α, γ) are the parameters of the model that respectively address varying linear and nonlinear color changes.

3.2.2 Color invariant descriptors

Following the presentation of the color change models, this subsection reviews and discusses existing color descriptors, characterize them according to their invariance and robustness properties with respect to the considered models. At some point this overview refers the work of K. van de Sande *at. al.* [118] in which a thorough evaluation of color descriptors is presented. The difference is that, in my PhD work, I focus mainly on the descriptor performance when used in a projector-camera system, rather than in object recognition systems.

In this review, the descriptors are mainly characterized according to their *color invariance* property which can be defined theoretically. *Color Invariance* to an illumination transform I guarantees that the descriptor vector remains unchanged under any color change defined by I. Figure 3.10 demonstrates this principle. Other properties, such as *distinctiveness and accuracy*, are experimentally computed and studied further in the thesis.

OpponentSIFT. This descriptor is computed for each channel (O_1, O_2, O_3) of an image in the Opponent color space. The linear transform, applied to an RGB image, is described below:

$$\left(O_1, O_2, O_3\right)^T = \left(\frac{R-G}{\sqrt{2}}, \frac{R+G-2B}{\sqrt{2}}, \frac{R+G+B}{\sqrt{2}}\right)^T \tag{3.3}$$

The O_3 channel retains the intensity information, while the other two channels represent chrominance. Thus, the descriptor is a concatenation of three SIFT descriptors computed on each Opponent color channel. Normalization, performed in SIFT, makes the chrominant channels O_1 and O_2 invariant to changes in intensity, *i.e.* when $a = b = c$ in (3.1). OpponentSIFT does not provide any invariance against variations in α and γ in (3.2). In the

evaluation [118] this descriptor produces the best results for image categorization applications.

C-SIFT descriptor, introduced in [1], uses the C-invariant which is the normalized 2D opponent color space defined by: $\frac{O_1}{O_3}$ and $\frac{O_2}{O_3}$. The normalization eliminates the residual intensity information in the chrominant opponent channels and, therefore, makes the C-SIFT invariant only to illumination changes: $a = b = c$ in the diagonal-offset model. However, for non-saturated colors the C-SIFT descriptor lacks discriminative power. In the evaluation [118] the performance of this descriptor was much lower than OpponentSIFT. For this reason it is not evaluated in this work.

HSV-SIFT, first used in [13], computes SIFT descriptors over three image channels in the HSV color space, where only the Hue provides invariance to scale and shift intensity changes. However, due to the combination of HSV channels, the descriptor has no invariance properties. Moreover, the H channel is instable for low saturation.

HueSIFT. H. van de Weijer *et. al.* [119] introduced a concatenation of the hue histogram with the intensity SIFT descriptor. By weighting the hue histogram with its saturation values the descriptor becomes more stable than HSV near the gray axis. HueSIFT is invariant to light intensity changes when $a = b = c$.

HUE+SIFT combined descriptor. An alternative to the previous descriptors, presented by B. Mazin *et. al* [77], implies instead of computing one hue histogram for each feature point, splitting the area into 9 sectors and constructing histograms for each of them. As the results, the descriptor is made of 9 local hue and one intensity histograms. This descriptor is similar to HueSIFT in many senses, but more local.

Color Constancy-based SIFT. Color constancy is the ability to identify the colors of the objects independently of light source color. Because the Human Visual System possesses this ability, many algorithms attempt to mimic the retina and cortex color perception. Retinex [42], the most widespread Color Constancy algorithm, represents a local approach that assumes that a given pixel's brightness depends on its own reflectance and the brightness of the neighboring pixels. Retinex does not rely on any of the above-introduced models, but locally estimates the illuminant color and subtracts it from each color in this local area. In [56] this method was used in feature matching and scored best among other methods in object and face recognition under changing light conditions.

Histogram Equalization SIFT. Histogram equalization (HE) is a technique to increase the contrast of grayscale images by making the distribution of the intensity histogram uniform. Recently, G. Finlayson *et. al* [38] showed that HE when applied to color images results in rank ordering preservation. It means that if, for instance, an object is acquired

under two different illuminants with the same camera sensor, the pixel ordering will be preserved for both images. This assertion works only if the color changes are monotonic for each color channel which is the case for both considered color change models that represent monotonic functions. Using HE together with the SIFT descriptor, therefore, should yield the desired invariance. As shown in [108] where this method was applied for color feature matching, HE-based SURF outperformed other methods under certain distortions.

In this PhD work some other color invariants could have been considered, for example, **rgSIFT** or **normalized RGB SIFT** [118], but in our previous evaluation they did not show good performance.

The next section provides a more detailed description of HE-SIFT and introduces an improvement to this method that locally computes HE called Local Histogram Equalization (LHE). Besides, it is shown that the use of LHE color invariant may be advantageous when used for feature matching in a system with complex photometric transformations between matched images such as the ones found in projector-camera systems.

3.2.3 Histogram Equalization based color invariant

Previously, HE was shown to provide color invariance to different photometric distortions as long as they represent monotonic functions. Thus, it becomes a more generic approach than most existing color invariants. However, there are two conditions that limit the use of HE in projector-camera systems. Firstly, both image areas in which equalization is to be performed should have similar content. Otherwise, if the regions undergo geometric distortions then rank preservation cannot be ensured anymore. Secondly, photometric distortions are not homogenously spread over the whole image, which is typically due to inhomogeneous illumination, blending with the color background, or diffused light from other surfaces. This also can affect rank ordering.

In view of the described facts, I suggest using a local version of HE (LHE) rather than just global equalization. Like in descriptor computation, the local area around a keypoint can be extracted and used for LHE, which means that the equalization needs to be performed only for the pixels that lie in the keypoint neighborhood. Moreover, from the definition of a keypoint it follows that this neighborhood is robust to various geometric distortions. This guarantees that in the LHE area (1) color distortions are likely to be homogenous and (2) local image content is preserved. Figure 3.11 illustrates the process of LHE for descriptor computation.

LHE is not only suitable for matching two images acquired by the same sensor, but can

Fig. 3.11 Example of LHE for a feature point

also be used in camera-projector systems to match original images with acquired projections. The following section justifies this affirmation.

3.2.4 Discussion on Local Histogram Equalization for projector-camera systems

This section gives a more strict reasoning of the use of LHE in projector-camera system. The spectrum $\mathscr{E}(\lambda,q)$ from equation (2.1) depends on the technology used in the projector, which in general can be LCD[2] or DLP[3] (for more details refer to Section 2.2). The lamp inside a DLP projector is typically based on mercury vapors which produces a discrete emission spectrum of three or more monochromatic beams. The same assertion of the discrete nature of the spectrum holds for LCD projectors. Inside these projectors, the white light beam, typically emitted be a cold-cathode fluorescent bulb, passes through a group of dichroic mirrors that break off only some specific wavelengths. Therefore $\mathscr{E}(\lambda,q) = 0$ except on some wavelengths. Thus, in the case of one wavelength per channel k, formula (2.1) can be simplified as:

$$g_k(q) = \mathscr{E}(\lambda_k,q)\mathscr{R}(\lambda_k,q)\mathscr{S}_k(\lambda_k) \tag{3.4}$$

[2]This technology is transmissive and is based on tiny and transparent LCD screens (0,55" to 0,9")
[3]DLP technology is reflexive and is based on thousand tiny mobile mirrors

In that case, the energy at wavelength λ_k represents a monotonic function F_k of the original intensity in that bandwidth, generally a gamma function as said previously:

$$\mathscr{E}(\lambda_k, q) = F_k(f_k(p)) \tag{3.5}$$

Finally, under these assumptions, equation (2.1) becomes:

$$g_k(q) = F_k(f_k(p))\mathscr{R}(\lambda_k, q)\mathscr{S}_k \tag{3.6}$$

where $\mathscr{S}_k$ is a constant that describes the sensor gain, but can vary over time due to the aging of the system. Let us now discuss on the validity of the rank preservation, with respect to the type of projection surface.

White Lambertian surfaces. Consider two pixels q and q' localized on the same surface. When the surface is white and Lambertian then $\mathscr{R}(\lambda_k, q) = 1$ and thus equation (3.6) boils down to:

$$g_k(q) = F_k(f_k(p))\mathscr{S}_k$$

Since function F_k is monotonic, for every $f_k(p) < f_k(p')$ we have $F_k(f_k(p)) < F_k(f_k(p'))$ and $g_k(q) < g_k(q')$. It means that the ranks are preserved and, consequently, HE may be used as a color invariant in this context.

Uniform surfaces. If the surface is uniformly colored and has homogeneous Lambertian reflectance, then it means that the surface reflectance is constant in any point $\mathscr{R}(\lambda_k, q) = \mathscr{R}(\lambda_k, q')$, $\forall q, q'$. Therefore, if $f_k(p) < f_k(p')$ then $F_k(f_k(p)) < F_k(f_k(p'))$ and $g_k(q) < g_k(q')$. On a uniform projection surface, the rank of colors is preserved between the initial image f_k and the distorted image g_k.

Non-uniform surfaces. Lets us consider two neighbor pixels p and p' located on two areas with different reflectance properties. In that case the rank is preserved only in some situations. In other words, $g_k(q) < g_k(q')$ when:

1. $f_k(p)$=$f_k(p')$ and $\mathscr{R}(\lambda_k, q) < \mathscr{R}(\lambda_k, q')$;
2. $f_k(p) < f_k(p')$ and $\mathscr{R}(\lambda_k, q) < \mathscr{R}(\lambda_k, q')$;
3. $f_k(p) < f_k(p')$ and $\mathscr{R}(\lambda_k, q) = \mathscr{R}(\lambda_k, q')$.

It yields that, for non-uniform surfaces, the rank is preserved only locally, either when the surface region is uniform in terms of colors (item 3) or when the colors of both image and surface vary in the same way (item 2), *i.e.* when both of them increase or decrease. If we

assume if that the surface varies smoothly in terms of reflectance $\mathscr{R}$, then its acquired color is locally constant in a small surface neighborhood W, *i.e.* $\mathscr{R}(\lambda_k, q) = \mathscr{R}(\lambda_k, q') \; \forall q, q' \in W$. Then the color rank is preserved locally in W.

Taking into account the provided formulations it can be concluded that global Histogram Equalization (GHE) (*i.e* computed once in the whole image) provides color invariance only when the projection surfaces are of uniform reflectance everywhere in the scene. A similar property holds for LHE, with the difference that the equalization areas are local and, in general, are significantly smaller than the whole image. However, the invariance is not guaranteed when the equalization area is situated at the edges between surfaces of very different colors, for example between two different color patches.

It is worthwhile to mention that, since this work uses local feature descriptors, the area around each keypoint, that needs to be equalized, can be obtained from the descriptor definition. However, if *a priori* knowledge about the image content or surface properties is available, it is possible to better adjust the size of the local neighborhood chosen for LHE which would result in a better color rank preservation.

3.3 Geometric compensation

In the last part of this chapter I present the geometric compensation part of my research work. As it is already mentioned in the previous chapters, I base my work on color feature matching to estimate the projective transform between the original image and its acquired projection. RANSAC is used as a homography estimation method. Therefore, this chapter continues with a review of homography compensation methods. Then, it presents an extension to the case of multiple projective transforms, by the example of double-homography transform. Finally, temporal projection compensation is addressed by presenting a new spatio-temporal compensation framework, based on feature matching and Optical Flow tracking.

3.3.1 Homography estimation

If projection is performed on a planar surface, the geometric relation between projected reference images and acquired projections can be described by a planar projective (homoraphy) transform. This 3×3 transform relates any two images of the same plane in 3D space. It can be estimated from a set of point correspondences (minimum 4 point pairs), obtained through feature matching. Employing a projective transform constraint, the transformation can be

estimated by optimization techniques. Although feature matching results can contain a lot of erroneous matches, homography estimation methods are capable of dealing with noise in data.

Most of the approaches for detecting a projective transform from a set of matches are based on RANSAC (RANdom SAmple Consensus) [39] or Direct Linear Transformation (DLT) [135] that both can be refined by Levenberg-Marquardt (LM) optimization [67].

RANSAC [39] is an iterative method that estimates parameters of a homography transform from a set of point correspondences that can possibly contain outliers. Matches are considered as outliers if they do not fit in the projective transform model. The algorithm iteratively takes a sample of point correspondences (4 pairs), computes the transformation between the points, and checks the number of matches that fit in the computed transform within some error threshold. The number of iterations as well as the error threshold represent the RANSAC parameters.

DLT [135], given a set of correspondences, estimates a homography by minimizing an algebraic cost function. The n matches form a a $n \times 9$ matrix A, and the homography h is represented as a 9×1 vector. The optimization function is the follows:

$$\min_h ||\mathbf{Ah}||^2 \tag{3.7}$$

Homography in general can only be determined up to scale. The solution h may have an arbitrary scale defined by the requirement $||h|| = 1$.

Data normalization is an essential step that reduces the effect of the arbitrary selection of origin and scale in the coordinate image frame and makes the algorithm invariant to a similarity transformation of the image.

LM [67], a dynamically damped version of Gauss-Newton algorithm, is frequently used to refine the estimates [9, 18, 24]. It iteratively reestimates the 9 parameters of the homography transform from the set of inliers.

3.3.2 Homography estimation methods from Image Stitching

Homography estimation is a key task not only in projector-camera systems, but in many other applications, for example in image stitching. Since image stitching is a well studied field in which many research works have been appeared, it was decided to study the state-of-the-art in stitching to approach the problem of geometric compensation in projector-camera systems.

Among recent works on geometric compensation for image stitching, two methods par-

ticularly stand out from the state-of-the-art because they outperform most of the existing techniques. The first one, proposed by J. Zaragoza *et. al.* [133], is the as-projective-as-possible warp which is globally projective while allowing local non-projective deviations for better alignment. This method makes use of Direct Linear Transformation (DLT) technique as a basic method for homography estimation from a noisy set of point correspondences. Prior to DLT, the data are normalized in order to avoid issues with numerical precision [52], and denormalized afterwards. The VLFeat[4] library was used to extract feature points and to match them. The second work, by C.-H. Chang *et. al.* [21], combines projective and similar transformations to produce a parametric warp. Homography estimation through RANSAC algorithm is followed by an additional refinement that performs a non-linear optimization of the residual function. This method also uses the VLFeat library for feature matching and RANSAC computation for homography estimation.

3.3.3 Multiple projective transform estimation

Estimation of multiple surfaces requires having at least two images to obtain the relative image transformation. This, however, represents a complex problem due to the lack of, and possible errors in the computed point correspondences related to each planar projection surface. Most of the works use image pairs (in case of stereo-reconstruction and multi-view [57, 111]) or a sequence of consecutive frames (structure from motion [137]) to obtain surface equations. However, these methods can hardly be adopted in our context of matching reference images with acquired projections, addressed in this PhD work, for two reasons. First, depth information about the scene cannot be retrieved, since only one projection image is considered. Second, high photometric differences between the reference image and its projections complicate the matching process, which is indispensable for homography transform estimation.

To solve this problem I propose to use a combination of several RANSAC calls to estimate all the transformations in the scene. This approach is similar to sequential RANSAC [57, 139] with some differences in estimation and compensation procedures, essentially, in points filtering and in finding intersections between 2D projections. The full algorithm is illustrated in Figure 3.12 and detailed below.

Feature matching, performed on a pair of images with detected feature points p_1 and p_2, gives a set of correspondences $m = \{i, j\}, i = [0, |p_1|], j = [0, |p_2|]$ that are then used by RANSAC to compute a homography transformation H_1 with the highest number of

[4] http://www.vlfeat.org/

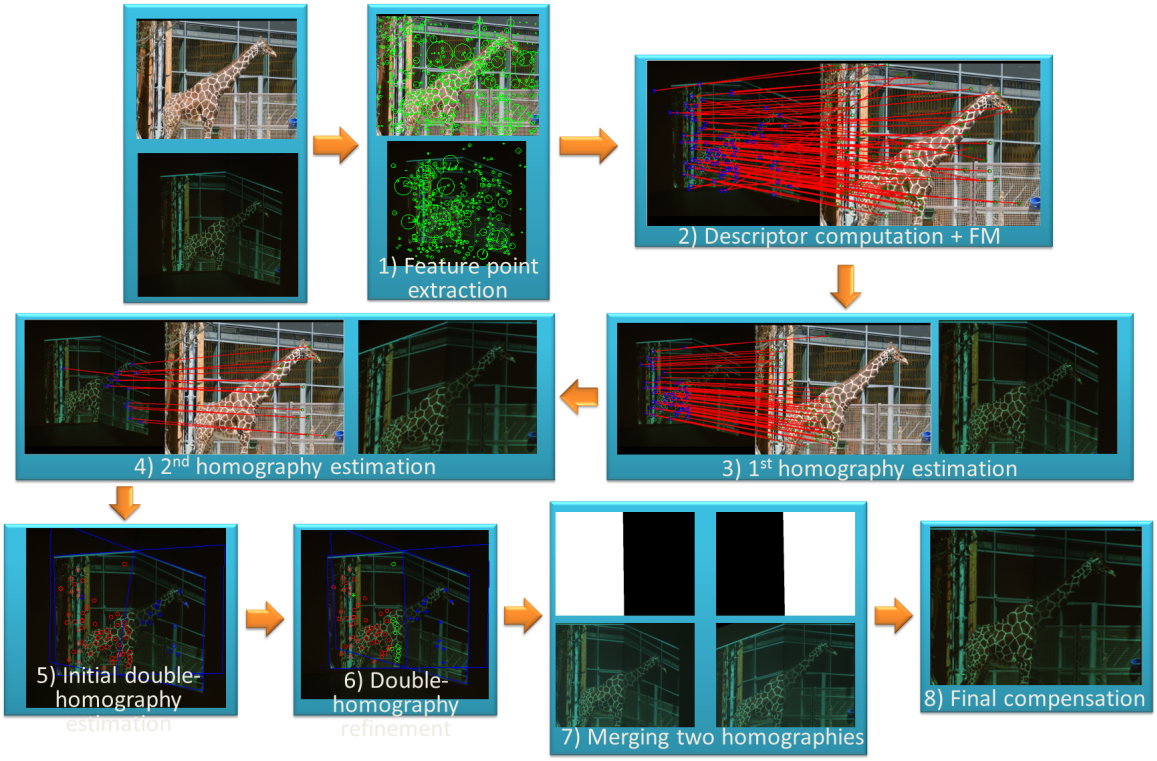


Fig. 3.12 Block diagram of double-homography projection compensation.

inlier points p_{in_1}. To estimate the second transformation H_2, the homography estimation procedure is repeated on the set of matches excluding the previously detected inliers $m = \{i, j\}, p_i \neq p_{in_1}$. When H_1 and H_2 homographies are computed, it is important to ensure that the all inliers p_{in_1} and p_{in_2} in reality belong to homographies H_1 and H_2, respectively. For this purpose, first, the orientation (vertical, horizontal, or diagonal) of the intersection line is found. It can be done by comparing the relative positions of the point centroids for each surface. Then, for each set of points the points that lie close to the border are removed. A 10% margin was found to provide good results on a set of test images. If the surfaces intersect along a vertical line, the margin is computed as 10% of the difference between the x-coordinates of the leftmost and the rightmost points in the set. Similarly, if the intersection line is horizontal, the difference between y-coordinates is computed. In the diagonal case the margin is computed as follows:

$$0.1\sqrt[2]{(\max_x(p_{in}) - \min_x(p_{in}))^2 + (\max_y(p_{in}) - \min_y(p_{in}))^2} \tag{3.8}$$

where p_in represents the inlier points belonging to the first or second planar surface.

After border points are filtered, RANSAC is applied again to recompute the homogra-

phies. In the experiments the margin of 10% was empirically selected. Note, that some other approaches can be applied, for instance a linear classification, to separate sets of points.

Once the homography transformations are refined, at the next step the regions that belong to each surface are defined in the compensation image. Since photometric difference is very high between the reference and projected images, comparison of re-projected image pixels [121] does not work well in this case. Therefore, to solve this problem the image coordinates are re-projected and the intersection of 2D surfaces are computed (steps 5 and 6 in the flowchart in Figure 3.12). Finally, binary image masks corresponding to each projection surface can be computed from the found points. These masks allow to merge the two compensations obtained through the refined H_1 and H_2 homographies in one compensation image (step 7).

This algorithm potentially can be extended to an arbitrary number of surfaces. Several conditions have to be addressed.

- There should be enough correctly matched points (at least 4) lying on each surface, and the precision should be good enough to increase the probability that RANSAC estimates a precise homography;
- Intersections should be correctly computed between all 2D surface projections. It can be done by iteratively finding one-vs-all intersections of the estimated 2D surface projections.

The described procedure is exploited in Section 5.2.1 that discusses the descriptor evaluation, performed on a set of real projections with both 1 and 2 homography transforms. The compensation has two uses: first, to obtain ground-truth transformations for the test images, and, second, to evaluate the descriptors. Ground-truth multi-homography compensation can be obtained in a similar way with the only difference that instead of feature matches, robust correspondences between checkerboard corners are used. For more details refer to section 4.2.3.

3.3.4 Temporal projection transform estimation

In the rest of the chapter I explore the use of Optical Flow tracking methods for geometric projection compensation in a sequence of video images. Exploiting temporal coherence of the projected content provides additional information that can be used to estimate transformations and to reduce the execution time. In that way, this section provides a short overview of Optical Flow methods that can be applied for dense, semi-dense or sparse point tracking.

Further, we propose a compensation system based on a combination of Feature Matching and Optical Flow methods, abbreviated hereafter as *FM-OF* compensation, in order to benefit from the advantages of each technique. Optical Flow is fast but can produce drifts in the precision of the projective transformation since the estimation errors can accumulate during time. FM is more time consuming, but it provides more reliable results. It can be run from time to time to tackle the shortcomings of Optical Flow. The presented work on *FM-OF* was done in collaboration with T. Rakotomalala in the context of a student internship.

Optical Flow algorithms for point tracking

Optical Flow (OF) methods have long been used for motion estimation and object tracking. First described by J.J. Gibson [45] in 1950, the OF represents the distribution of apparent movement velocities of brightness patterns in an image. The method estimates the spatial displacement of image pixels assuming that the properties of consecutive images are similar in a small region as shown in:

$$I(x,y,t) = I(x+\delta x, y+\delta y, t+1). \tag{3.9}$$

where $\delta x, \delta y$ are displacements along horizontal and vertical axis, respectively; $I(x,y,t)$ and $I(x+\delta x, y+\delta y, t+1)$ are intensity values of pixels (x,y) and $(x+\delta x, y+\delta y)$ at time moments t and $t+1$, respectively. Based on the previous assumption, called *brightness constance constraint*, the OF constraint can be formulated as:

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t}\frac{\delta t}{\delta t} = 0 \tag{3.10}$$

which results in:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \tag{3.11}$$

where V_x, V_y are the x and y components of the OF velocity and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the partial derivatives of the image $I(x,y,t)$ in the corresponding directions, which can be shortened as I_x, I_y and I_t. The OF constrain then assumes the following form:

$$I_x V_x + I_y V_y = -I_t \tag{3.12}$$

This last equation has two unknown values and, thus, requires additional constraints. B. Horn and B. Schunck [51] introduced a global smoothness constraint to estimate the OF over the whole image and to favor solutions that show smooth motion. However, this method is

less efficient when displacements are small. B. Lucas and T. Kanade [72] developed a robust local method for OF estimation that assumes that the motion is constant in a local area. This constraint allows solving the equation (3.12) for all pixels in that neighborhood. Moreover, this method is less sensitive to image noise. For these reasons, Lucas-Kanade OF is nowadays the most widely used method. The pyramidal implementation of the Lucas-Kanade OF [132] from OpenCV was used in the further experiments.

Projective Transform Compensation based on Feature Matching and Optical Flow

The rest of the current chapter introduces a new compensation system that exploits both spatial and temporal image coherence in order to estimate and compensate homography transformations. The developed approach bases on color invariant feature matching, thoroughly discussed in 3.2, and Optical Flow methods, described in 3.3.4. The idea of such temporal-spatial combination is similar to what was presented in [64], color invariant SIFT descriptors were used together with M-estimator SAmple and Consensus method (MSAC) to improve the compensation quality. MSAC [117] is a modification of RANSAC that computes the likelihood of the consensus set to find the minimum of the loss function. Contrary to the described method, in the compensation system I use color invariant feature matching only for the initial transform estimation and for further adjustments introduced from time to time. In the rest of the time the robust points extracted during feature matching step are tracked by an optical flow method. This represents a more lightweight solution for image compensation because the points can be reused from one frame to the next.

A block diagram in Figure 3.13 illustrates one iteration of the compensation system. I^t_{ref} and I^t_{proj} refer to reference and projection images in iteration t, or, in other words, *t-th* frames of the reference and acquired projection video sequences. In the first step, the color image histograms of the current and the previous reference images are analyzed. To detect a change of scene, the distance between successive histograms of the reference images is computed on the R, G and B channels separately. To measure the distance between two histograms, a correlation coefficient is used:

$$d_C(H^t, H^{t+1}) = \frac{E[(H^t - \mu_{H^t})(H^{t+1} - \mu_{H^{t+1}})]}{\sigma_{H^t}\sigma_{H^{t+1}}} \tag{3.13}$$

where d_C is the correlation between histograms H^t and H^{t+1}; μ and σ stand for the mean and the standard deviation; E is the expected value operator.

If the correlation value at least for one color channel c is below a certain threshold Thr_c, then a scene change is flagged. In this case feature matching (FM) is performed in order to

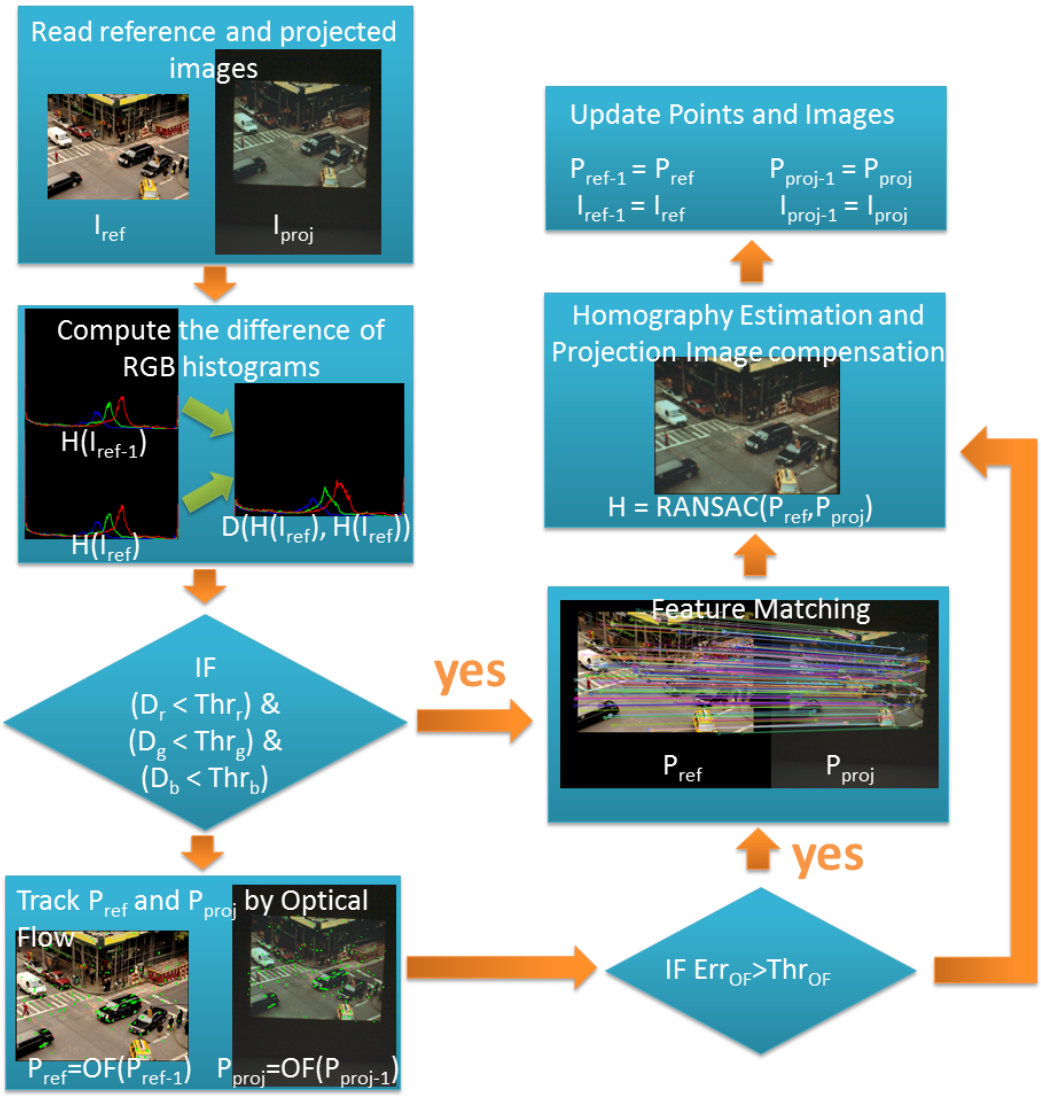


Fig. 3.13 Block diagram of the FM-OF compensation system.

reestimate the set of feature points in the current projected and reference images.

If I^t_{ref} and I^{t+1}_{ref} are close enough in terms of their color histograms, Optical Flow (OF) is executed twice, once for a pair of consecutive reference images, and then for a pair of acquired projections. OF tracks only the point vectors P^t_{ref} and P^t_{proj}, detected in the previous iteration (from FM or OF), which reduces the overall time required for tracking compared with the dense OF. Next, the average error produced by OF is examined. It is computed as the average $L1$ errror between tracked patches. If the error value is higher than threshold Thr_{OF}, then FM is executed. Otherwise, if the points are correctly tracked in the current image, homography transformation can be estimated from the obtained P_{ref} and P_{proj} set of

points. RANSAC estimates the homography transformation which is used to compute the compensation. Finally, the point vectors are updated before going to the next iteration.

If two consecutive reference images are different in terms of content but have similar color histograms, then a change of scene is flagged falsely. OF, executed at the following step, will most likely fail yielding a high error value, which means that FM will be executed.

The presented compensation framework was applied to compensate acquired video projections described later in Section 5.3.

It is important to discuss some limitations of the FM-OF method. Namely, geometric compensation performance of the method depends on the performance of the used OF method. The pyramidal Lucas-Kanade OF implementation cannot deal with high scale changes, like in case of *camera forward motion*. Similarly, the FM-OF will most likely not work well in the case of non-rigid transformations in the projected video.

3.4 Conclusions

In this chapter several state-of-the-art techniques in feature matching were presented. Two groups particularly stand out among all the methods. The first one includes SIFT and SURF that are used in most of the applications for feature matching and object recognition. The second group represents binary descriptors that provide significant speed-up when compared with their floating-point competitors. Nevertheless, in some cases they are outperformed by SIFT-like methods, that is why it was decided to consider SURF or SIFT methods in further evaluations.

In the second part color invariance extension of feature matching was discussed. Among existing color invariance transforms, I selected Opponent and HSV because they possess invariance properties against the considered models. Moreover, LHE-RGB was introduced as a color transform that is invariant to various monotonic color change functions. The performance of the chosen invariance transforms is studied in an evaluation framework in chapter 5.

The final topic addressed in this chapter is the problem of homography estimation in the context of ProCam systems. In the first place, static projective transform estimation was considered. Single- and multiple-homography compensation through RANSAC was discussed. Next, a temporal extension of geometric compensation to a temporal sequence of images was described which makes use of the Optical Flow point tracking method. The both compensation frameworks are evaluated on different datasets. The evaluation results are reported in chapter 5. The following chapter in turn presents several datasets used for

algorithms evaluation.

Chapter 4

New Datasets for Projector-Camera Systems

One part of my research works aims at evaluating the performance of different algorithms in the projection-acquisition scenario. For that purpose it is important to perform experiments in conditions that are close to real ones occurred in a ProCam system. Since there is a lack of such test data available, a part of the thesis was dedicated to prepare several dataset suitable for algorithm evaluation by addressing various conditions.

This chapter presents three datasets, purposefully prepared in this work. Firstly, to evaluate the performance of different color-invariant descriptors from 3.2.2 in homography estimation, several sets of synthetic images were prepared that simulate various photometric distortions and projective warpings [109]. Such test data allows to assess the descriptor performance by analyzing the quality of feature matching and homography compensation. Secondly, it is important to make evaluations on real-world projections. To that end, I prepared a large dataset of acquired projections covering a wide range of condition, such as ambient illumination and different surface color and geometry [107]. The dataset also contains additional data that allow computing ground-truth homography transforms and performing color corrections of the acquisitions. To my knowledge, this is the first database of acquired projections which can be useful for various evaluations. Several applications are shown later in chapter 5. Finally, the third dataset represents an extension of the previous one which implies video projections obtained with a dynamic ProCam system rather than the case of static projections in the previous dataset. This new test data is necessary for evaluating different algorithms used in projector-camera systems, non only static, but those that exploit temporal coherence between video video frames.

4.1 Synthetic Images

The idea of using a dataset of synthetic images was to make a fast automatic dataset generation and evaluation of various descriptors under different controlled distortions. Color warping and homography transformations allow to provide simulated conditions close to what is encountered when using a projector-camera system. Photometric conditions, generated in accordance with several color change models, address such components as varying illuminations, camera and projector responses. Each image in the dataset undergoes a geometric warping that models a homography transform. The six types of synthetic distortions, denoted *S1* to *S6* for sake of clarity, are listed in Table 4.1. Examples of test images are illustrates in Figures 4.1 and 4.2.

Synthetic database	Brief description	Number of test images
S1	Photometric distortions (3.1) and (3.2). (Fig. 4.1, column 1)	100
S2	*S1* + Gaussian noise (Fig. 4.1, column 2)	100
S3	*S1* + Homography transform (Fig. 4.1, column 3)	100
S4	*S2* + Homography transform (Fig. 4.1, column 4)	100
S5	Normal mapping (Fig. 4.2, top row)	15
S6	*S5* + background color blending (Fig. 4.2, bottom row)	45

Table 4.1 The six types of synthetic distortions used for the evaluation.

The first distortion type *S1* simulates real photometric conditions under the assumption that the real physical equation can be approximated by the Diagonal Offset (3.1) or the Gamma models (3.2). All the parameters of these models were randomly generated as uniform random variables with the standard deviation of 0.5, *i.e.* γ for each channel varied in the range from 0.5 to 1.5. Thus, the generated parameters address most of device functions [47]. The "Lenna" image was used as a single reference image for the whole set. In total, 100 images were generated with different distortion of *S1* type. Then, *S2* distortions were made by applying Gaussian noise to the previously distorted *S1* set of images. The Gaussian transform with uniform random parameters (a variance of 40) aims to simulate camera noise.

Images of *S1* and *S2* have only photometric distortions without any geometric warping. In *S3* and *S4* synthetic homography transforms are introduced so that they simulate non axial projections on a single planar surface. Homography warping was applied to the images from *S1* and *S2*. In each case to generate a homography transform, four corner points are warped so that each point was translated at a distance $d \in [-100, 100]$ and rotated by an angle $\alpha \in [0, 360^\circ]$. Examples of synthesized distortions *S1*-*S4* can be seen in Figure 4.1.

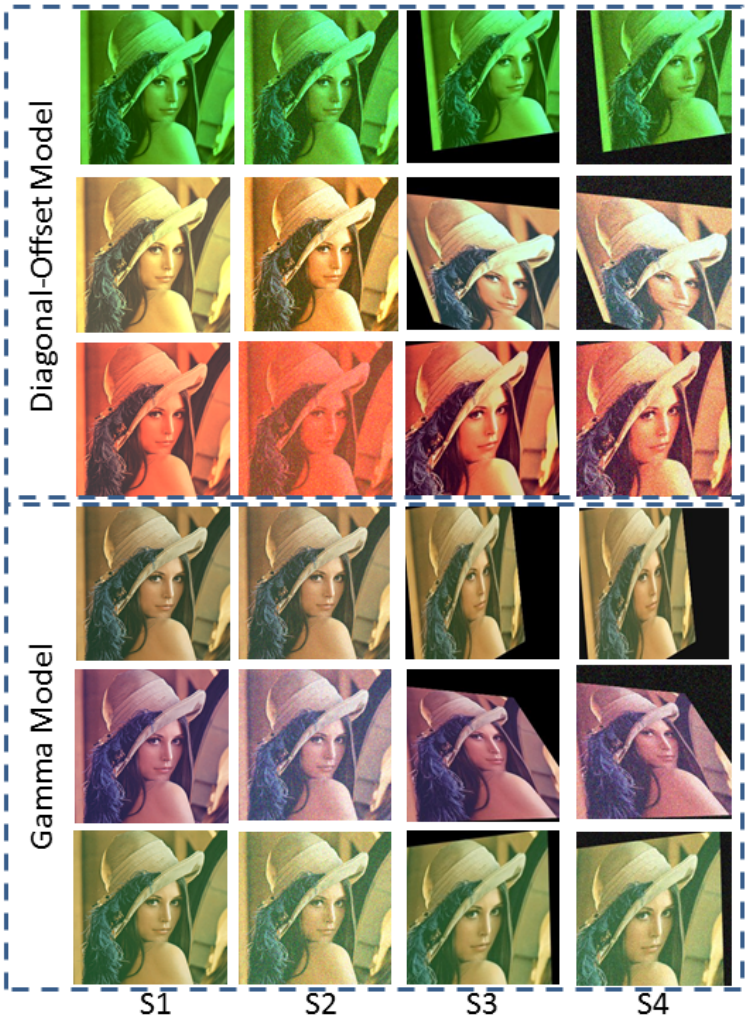


Fig. 4.1 Examples of synthetic photometric and geometric (homography) distortions. Columns represents *S1-S4* distortions types. The first three rows correspond to the images distorted according to the Diagonal-Offset model, while the last three rows obtained in accordance with the Gamma model.

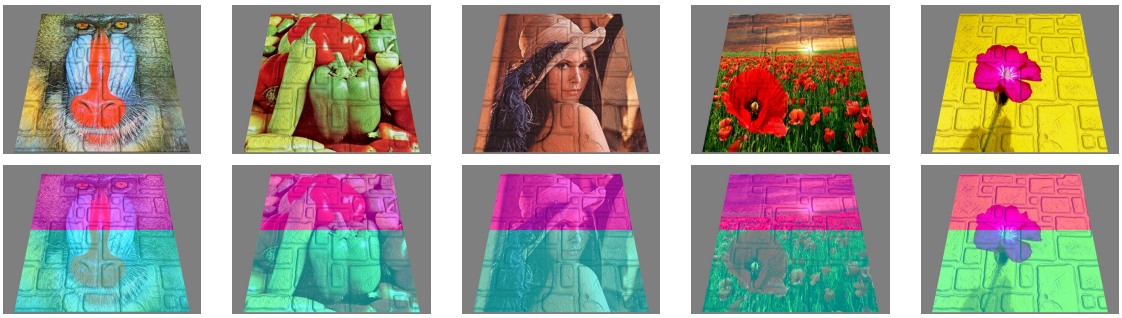


Fig. 4.2 Example of synthesized distortions *S5*: normal mapping. The top row illustrates an example of normal mapping applied to reference images. The bottom row shows the results of blending with a color background texture to produce *S6* test images.

To go further in the analysis, I introduce *S5* distortions that correspond to normal mapping. This technique enables pixel lighting variations according to normal displacements and, in such a way, can model background surfaces with small bump structures like wood, brick or metal. Using an OpenGL[1] rendering pipeline, test images were distorted and ground-truth coordinates were obtained. Several popular materials were chosen, such as metal, brick, concrete, or wall. Figure 4.2 shows some examples in the top row. Each of the five images underwent normal mapping specified by the displaced normal map. Note that normal mapping does not cause any change in pixel coordinates and thus does not take in consideration bigger geometrical distortions of a surface that entail parallax effects and pixel displacements. Only lighting of each pixel is modified through normal displacement.

In the last type of distortions *S6* photometric variations are added to the previous distorted images of normal mapping *S5*. To simulate color background blending and, thus, to make the simulated distortion even harder, selected color or black and white images were blended with the reference images before applying normal mapping. A blending ratio of 50% was used, which means that both background and reference images equally contribute to the final texture color. Some results can be observed in Figure 4.2 in the bottom row.

The total number of test images per testing scenario is shown in the rightmost column of Table 4.1.

[1] OpenGL (Open Graphics Library) is a cross-language, multi-platform application programming interface (API) for rendering 2D and 3D vector graphics: https://www.opengl.org/

4.2 Static Images Projected In A Static Environment

The database of real-world acquired projections, prepared with the use of a ProCam system, addresses various geometric and illumination conditions. Moreover, it provides additional data for ground-truth homography transforms computation and color correction of the acquisition and projection devices. Projected images were specifically collected to make the database tailored for the use in different scenarios. Therefore, the database description goes as follows. First, the experimental setup is detailed. Second, the set of reference images chosen for projections are described. Next, the projection-acquisition pipeline is presented along with the database structure and some details on the acquisition process. Finally, ground-truth preparation is explained.

4.2.1 Experimental Setup Description

The experimental setup, used in this work to make real-world projections, consists of three main components :

1. Acquisition-projection devices (form a projector-camera system);
2. Projection scene (different geometric and color characteristics);
3. Illumination (several sources of illumination are considered);

Projector-Camera system

The first component, a projector-camera system, combines one projection and one acquisition devices connected to a desktop computer (refer to Figure 4.3 for more detail). The devices are rigidly fixed with respect to each other in order avoid mutual displacements throughout the acquisitions. The devices are mounted on metal bars and placed on top of a rolling wheel support. The desktop with the acquisition software is placed under this support (not seen in Figure 4.3).

The following projection and acquisition devices are used in the setup:

- an LCD-projector Epson EB-X11
- 1 Basler firewire camera A641FC with an 8-mm lens.

Figure 4.4 illustrates the processing pipeline which is typical for a projector-camera system. First, an image I is projected by the projector that introduces color distortions $P(I)$.

Fig. 4.3 Experimental setup. Top-left figure illustrates the acquisition and projection devices. Bottom-left figure shows the projection scene with an installed background poster. Three ARToolKit markers in the scene are used to save the exact setup positions throughout the acquisitions. Right figure - the acquisition system with two installed illumination devices: fluorescent lamps (on the top of the setup) and an incandescent lamp bulb.

Then, the camera acquires the geometrically warped projection $C(G(P(I)))$ where C stands for the camera matrix and G defines the geometric transformation. Steps 3 and 4 consist in geometric and color compensations that can be performed in any order. Finally, the system projects the computed compensation $P^{-1}(P(I))$.

The acquisition process is automatic for a given projection scenario, *e.g.* for a given projection surface, setup position, illumination, and background. The camera and the projector are connected to the computer on which the acquisition software is run. The system projects and acquires in series test reference images from the database. The software code is written in C++ using the following libraries:

- OpenCV to read/write images and for simple image processing[2];
- Basler API to access and control on-the-fly internal camera properties from C++ code:

[2] a library of Computer Vision programming function https://http://opencv.org/

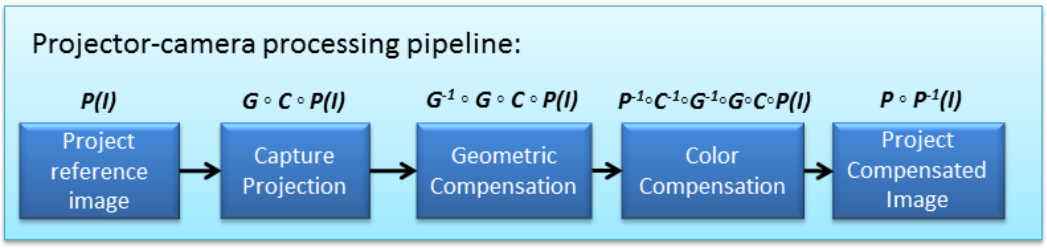


Fig. 4.4 A sequence of steps performed inside a ProCam system.

Fig. 4.5 Schematic projection modeling on one and two planar surfaces.

the exposure time, the image format (RAW or demosaicked), white balance, gamma correction[3];

- ARToolKit library to detect ARToolKit markers in the scene[4].

Projection scene

The projection scene is made of three white planar cardboards that form a concave corner on the junction. The projections are made on one and two projection surfaces to have both single- and double-homography transformations.

The distance between the projection surfaces and the acquisition system is around 1-2 meters so that the projection size does not exceed 1 meter for each projection surface. This constraint was chosen in order to facilitate the design of the projection scene. The camera is placed near to the projector at an arbitrary distance and angle so that the projection is well situated in the captured image. In the acquired images (resolution 1624×1234) projections occupy approximately 10% of the image surface (one third of the acquired image length and width).

[3] Basler software can be download from http://www.baslerweb.com/en/products/software
[4] A library for building Augmented Reality applications http://www.hitl.washington.edu/artoolkit/

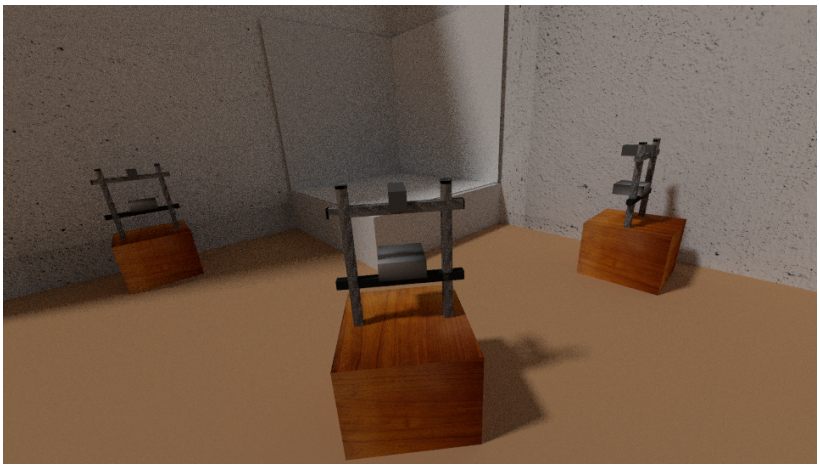


Fig. 4.6 Schematic representation of three differerent positions of the acquisition setup. From left to right: position 1, position 2 and position 3.

Projections were made on each surface configuration from several viewpoints. In case of static projections, three different positions were chosen (see Figure 4.6). Whatever the position, the projection/acquisition devices and the illuminations (except for daylight) are rigidly fixed with respect to each other.

It is important to mention, that the setup positions others than the orthogonal one introduce additional photometric distortions in the final projection. The projection surface is differently lit by the projector because of the different distances from the projector lamp to various parts of the projection surface.

Background posters. In the acquisitions I attempted to model such parameters of the projection surface as color and texture. To that end, several printed projected surfaces were installed in the projection scene and used in the acquisitions. Physical installation consists in attaching paper posters to the cardboards that serve as projection surfaces. The geometric parameters of the scene do not change when a background poster is added. The trivial case, when the projection is performed on white cardboards, simulates the absence of color background. The following images (see Figure 4.7) were used to produce background posters used in the acquisitions of static projections: Gray checkerboard, Colorchecker image and Dalí's painting "The Persistence of Memory"

Illumination conditions

During the acquisitions it was important to simulate lighting and projection conditions corresponding to a typical projector-camera system. For that purpose four illumination scenarios were considered for all acquisitions:

Fig. 4.7 Posters that model color background: a Macbeth colorchart, a Dalí's painting, a grayscale checkerboard.

- Dark room. The only light source in the scene is the projector;
- 2 fluorescent lamps. This type of illumination was chosen as an example of the white light. The lamps are mounted on the top of the projector-camera system (see Figure 4.3);
- Incandescent lamp is a source of reddish illumination which significantly differs from the fluorescent light. The bulb lamp is mounted on the projector-camera system (see Figure 4.3);
- Daylight illumination. The choice of this illumination can be explained by the fact that daylight is often present in the projection scene which complicates the normal perception of the projected content. Though it is difficult to characterize this source of illumination, most acquisitions were made in a sunny weather with the clear sky at around midday which allows its approximation with D60 illuminant.

4.2.2 Database of static projection images

The database of static projections aims at providing enough test data to perform evaluation of various algorithms used in projector-camera systems, at the same time modeling different conditions that typically take place in a real scene. In the course of database preparations, the following distortions were considered (refer to subsection 4.2.1 for more details) :

1. Color distortions that can be due to one or several factors:
   (a) Illumination in the scene
   (b) Color Background
   (c) Projector Radiance

Category	Subcategory	N images
Landscape images	sea	3
	mountain	3
	forest	3
	desert	3
	waterfall	3
	field	3
People	Actions (sport, meetings, etc.)	4
	Others	4

Category	Subcategory	N images
Urban scenes	vehicles	5
	buildings	4
	bridges	3
	roads	4
Animals		6

Table 4.2 Summary of 162 projected natural images in the database : different categories and numbers of images.

2. Homography transform: geometric transformation when the projection is made on one or two planar surfaces. Both single- and double-homography transformations are considered when preparing the data set.

Projected reference images

The developed dataset is intended to be a universal tool for evaluation of different smart-projection algorithms regardless of the target application. On the one hand, the choice of the original images to be projected is very important in order to make the dataset useful in many PAR applications. On the other hand, the size of the database should be reasonable.

It was decided to make the dataset on the basis of the following three types of images:

1. **Natural images.** The principal idea is not to impose any constraints on possible target applications of the evaluated algorithms. Thus, a generic Flickr image database[5] was exploited as the source of reference images. Image selection process consisted of several steps. Firstly, several main image categories were defined according to expert appraisal (several experts in computer vision, image processing and computer graphics made a joint decision on several issues). Secondly, I arbitrarily picked up 20 images of each category. Finally, expert appraisal was used again to select several images of each considered category. In total 48 images were included in the set of projected reference data. The categories and the number of selected images are summarized in table 4.2.

2. **VideoConferences.** Another scenario that might be of interest to projector-camera systems is video teleconference. Human faces are most probably the main objects

[5] https://www.flickr.com/

present in the projected video in such an application. Thus, it was decided to use the database of color indoor images of human faces called McGill Real-World Face Video Database[6]. Based on expert appraisal, from this database I chose 8 video streams according to the presence of complex background, various human faces (different age, nationality, glasses etc.) as well as different poses. Then, from each stream 10 frames were randomly extracted and included in the set of projected reference images. The resolution of each frame is 1024×768 pixels.

3. **Powerpoint presentations.** The dataset also aims at evaluating projector-camera systems for corporate presentations. For this purpose 5 publicly available educational presentations were selected from Coursera[7] and MIT webpages[8]. In total 34 slides were selected that represent the following cases: title slides, slides with plots, slides with text, and slides with images.

The total number of projected images is 162. Figure 4.8 shows some samples of the projected images.

Database building pipeline

Figure 4.9 illustrates the sequence of processing steps to build the dataset of acquired image projections. The acquisition process consists in projecting and capturing projected images in series for each setup configuration. Each configuration is defined by several components: the projector-camera setup position (one of the three predefined positions), the number of projection surfaces (1 or 2), the illumination (one of the four illumination sources), and the projection background (if used).

Preparation of the setup for acquisitions was done as follows. First, geometric surfaces needed to be prepared to make projections. The setup was aligned in order to make projections on the desired number of surfaces (one or two). Second, a color poster was mounted in the scene (if color background is used). Next, I placed the setup at one of the three predefined positions. The previous steps altogether form the geometrical configuration. For further geometrical device calibration and ground-truth computation (more details go further in this chapter), a checkerboard image was projected and acquired with the camera, once for each geometrical configuration.

[6] https://sites.google.com/site/meltemdemirkus/mcgill-unconstrained-face-video-database

[7] https://www.coursera.org/

[8] http://ocw.mit.edu/

Fig. 4.8 Examples of projected images. The first row - natural images of different categories; the second row - images with human faces (used for the video conference scenario); the third row - images of presentation slides.

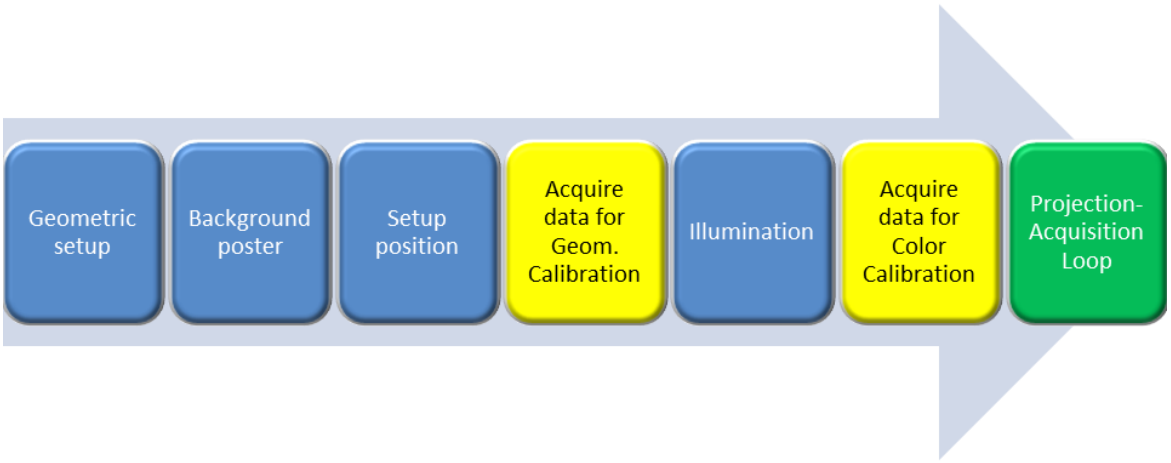


Fig. 4.9 Acquisition pipeline. Blocks in red denote the acquisition process of data used for geometric and color compensation. The green block is the projection/acquisition loop. Blocks in blue describe the preparation steps for acquisitions.

Next, illumination was to be prepared in the scene. For color calibration I acquired additional data before to start projections. It included projections of images that form an RGB cube of 256^3 colors to compute the projector response function. Moreover, projections of white and black images were performed that could be useful in color image compensation. For the camera color calibration and illumination estimation, a colorchart and a white board were acquired. More detailed explanation of the device calibration and ground truth computation is presented later in this chapter.

All 162 reference images were projected in the loop as many times as there are different setup configurations. Since changing a background poster represents a more difficult task than changing the setup position, it was decided to put each poster once for all setup positions and illumination sources. Once the poster is mounted, the setup can be at one of the three positions. To keep exactly the same position for all background images, I developed an ARToolKit-based automatic alignment. The system could find the precise position by detecting the three visible markers in the scene and then comparing their coordinates with those preliminarily stored in the system.

To summarize, the following types of data are included in the database:

1. Acquired raw projection images;
2. Acquired images of the colorchart and white board for color data post-processing;
3. 256 projected acquired images that represent an RGB cube for color projector characterization;

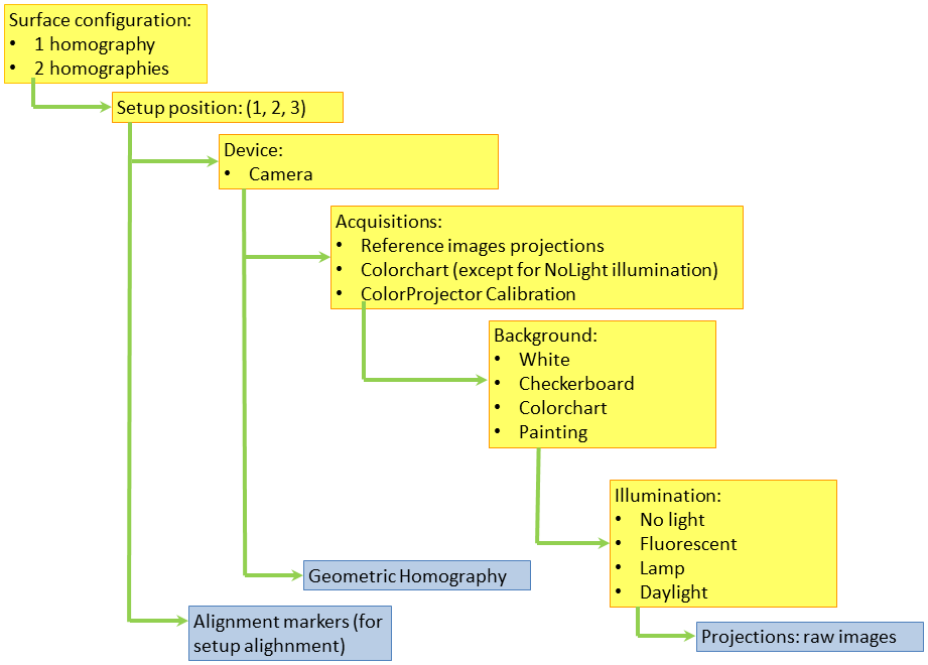


Fig. 4.10 The structure of the Database. Yellow blocks denote acquisition conditions and blue blocks correspond to acquisitions.

4. Acquired projected white and black images for color projector characterization;

5. Projected and acquired checkerboard image for geometric device calibration;

Types 2-5 were acquired once for each geometric configuration, position and illumination in the scene. Type 6 was acquired once for each geometric configuration and position.

The total number of projections can be estimated as a product of the number of projected images, 4 illuminations, 3 positions, 4 posters, and 2 homographies: $162 \times 4 \times 3 \times 4 \times 2 = 15,552$ projections. The structure of the database is illustrated in Figure 4.10. Figure 4.11 shows some acquisition examples of projections that include single- and double-homography transformations, different illumination in the scene and color background images. Note, in the figure the presented images were demosaicked and underwent a simple color compensation in order to make the images clearer.

Camera Acquisitions

In this section several technical details on image acquisitions are presented:

Image format. It was decided to choose RAW image format for all images acquired with the camera. This allows manual demosaicking that significantly improves the results

Fig. 4.11 Examples of acquired images. The top row illustrates 4 color background posters used in the acquisitions. The second row represents 4 illumination conditions addressed in the acquisitions. The third and the fourth rows shows 3 setup positions for single- and double-homography transform, respectively.

since the built-in algorithm produces well-marked artifacts. The demosaicking procedure, therefore, can be done offline as a part of the data post-processing procedure.

Exposure time. The exposure time varied throughout the acquisitions. I used one value for all acquired projections, and another for non-projection acquisitions (colorchart, white board). Because of the big difference in projector luminosity as compared to the scene lighting, it became necessary to use different parameters. If not varying the shutter speed, the captured images would be saturated in case of acquired projections, and would have dark noisy areas when acquiring colorcharts.

Gamma correction. Although the technical documentation from the camera manufacturer clearly states that the camera has a linear response, in practice, we could not get a response that can be considered as linear. All the acquisitions were done without any external gamma correction, but it could be performed, if necessary, as post-processing, from the acquired colorchart images.

Acquired mean image. The colorchart and white board images were acquired 10 times and the mean image was then computed and stored. This helps to reduce the level of noise in the acquired images and, therefore, to enhance the compensation quality. Figure 4.13 shows the camera sensor response measured from 6 captured gray values of the colorchart. The left and right plots represent two different illumination scenarios.

4.2.3 Ground Truth

Geometric Compensation. During the whole process of acquisitions additional data on the geometric and photometric setup parameters were collected. It was done for the purpose of computing geometric ground-truth information and color compensation of the images. Geometric ground-truth was used in this work to assess the performance of the evaluated algorithms. Acquired colorchart images allow to perform color post-processing on the acquired data, for example for gamma correction, white balance or distortion compensation due to camera sensor or illumination.

As mentioned above, for each geometric configuration, a checkerboard image was projected and captured with the camera. Precise correspondences between the corner points are extracted from the projected and captured images using Computer Vision System Toolbox in matlab. The computed points were further used to compute a precise homography transform which serves as ground-truth in this work. The case of double-homography transform represents a more difficult task than a single-homography, because the precise regions corresponding to each projection surface should be obtained. Ground-truth compensations were

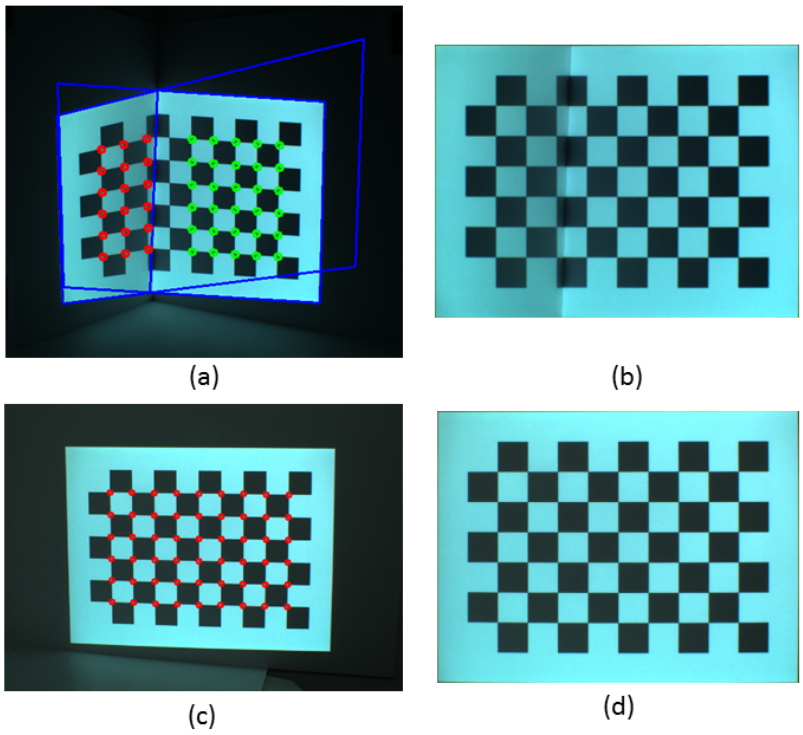


Fig. 4.12 (a) Acquired checkerboard image projection on two planar surfaces with the detected corners and the projection surface intersection; (b) Double-homography compensation; (c) Acquired checkerboard image projection on a single planar surface with the detected corners; (d) Single-homography compensation.

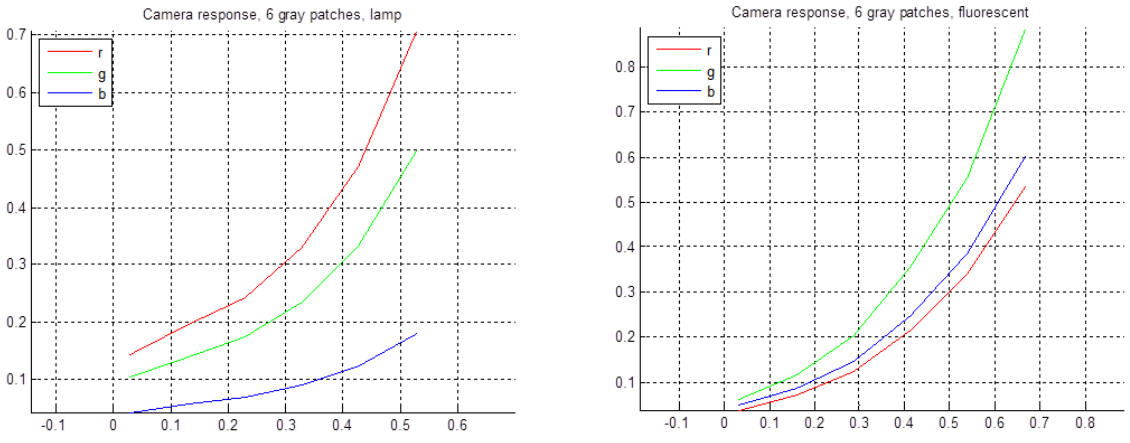


Fig. 4.13 Measured camera gamma respose functions. The left plot corresponds to the incandescent lamp illumination; the right plot - fluorescent illumination. Intensity values are normalized in the range [0,1].

	pos1	pos2	pos3
left	0.0337	0.0218	0.0291
right	0.0597	0.0502	0.074

Table 4.3 The average Euclidean Distance between the checkerboard corners and their projections on two planar surfaces (double-homography transform). The values are presented separately for the left and the right surface.

obtained through the algorithm from sub-chapter 3.3.3. The least-squares optimized spatial transformation function, provided by Matlab Image Processing Toolbox, estimated a projection transformation from point correspondences. The precision of the obtained homography transforms mainly depends on the accuracy of the detected corner points. Table 4.3 shows the average $L2$ distances between the corner points for each homography trasform in the image. The results are presented for 3 different setup positions. Figure 4.12 illustrates acquired checkerboard projections with the detected corner points, as well as projections compensations.

4.3 Dynamic Video Projections

In this last part of the chapter I present another series of acquisitions intended to reproduce more challenging scenarios of the use of projector-camera systems. First, the previous acquisitions are extended by increasing the number of projection surfaces so that to have simultaneously 3 or 4 homography transforms in the captured images. Next, static images previously used as reference content are replaced video projections. Finally, I extend the experiments by making the projector-camera system moving while projecting and acquiring images.

4.3.1 Acquisition Scenarios

The main used acquisition types are listed below:

1. Static ProCam system + static projections + 3 or 4 projection surfaces
2. Static ProCam system + video projections + 1 or 2 projection surfaces
3. Dynamic ProCam system + static projections + 1 or 2 projection surfaces
4. Dynamic ProCam system + video projections + 1 and 2 projection surfaces

Fig. 4.14 Videos sequences used in the acquisitions. Video 1 represents natural scenes; video2 - urban scenes; video 3 - news report.

The first type is essentially an extension of the static projections described in 4.2. Addressing simultaneously 3-4 homography transforms allows modeling such real projection scenarios, as complex wall corners and ledges. The second type of acquisitions supposes projection of video sequences on 1 or 2 projection surfaces. The reason why video projections are considered in this work as against static images, is to exploit temporal coherence in video sequences which is important for such algorithms as Optical Flow or background subtraction. The third and fourth acquisition scenarios imply that the projector-camera system is no longer static, in other words projections are captured while the system is in motion. In a simpler case only one static image is projected so that only geometric warping changes in the captured images due to ProCam system displacements. Then, a video sequence replaces the projected static image to make the compensation task more difficult because homography transforms and projected images change simultaneously throughout each series of acquisitions.

4.3.2 Setup description

In case of dynamic projections (types 3 and 4), it is important that the system processes enough frames per second to ensure smooth video acquisitions. Because the previously used Basler camera had a very low acquisition speed at the highest resolution, it was decided to replace the camera and use Ethernet IDS UI-5240CP-C-HQ camera [9]. By using such configuration the speed of 8fps was achieved in the experiments which is however sig-

[9] camera specifications can be found at https://en.ids-imaging.com/store/ui-5240cp.html

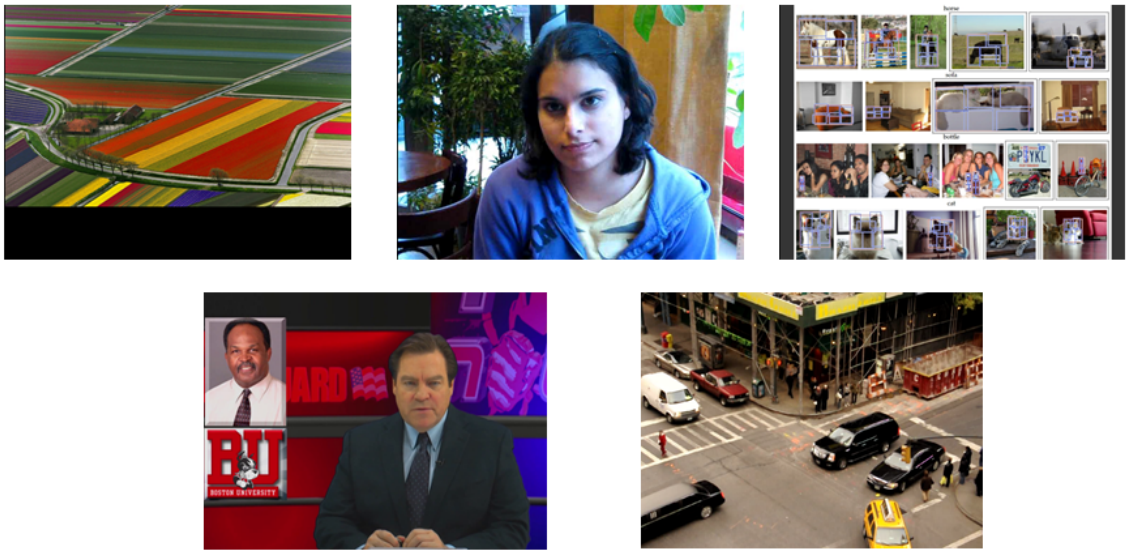


Fig. 4.15 Static images that were projected by the dynamic setup (scenario 3).

nificantly lower than the one from the camera specifications. This loss in performance arose from several reasons, among them the most important are projector-camera synchronization issues, difference in camera and projector frequencies, and limited memory size to buffer the images. In practice, the obtained videos were found to be smooth enough to be used in further evaluations.

To simplify the acquisition process, only white background was considered in all the acquisition scenarios. Hence, color background was out of scope for these experiments. Scenario 1, when separate image frames were projected, included 3 and 4 homography transformations. The former transformation was made by three cardboards that formed the concave corner of the projection scene, while the later was ensured by projecting on a complex wall corner that resulted in 4 homography transformations without discontinuities (see Figure 4.16). Moreover, 4 homography transform acquisitions were performed only from 2 setup positions to reduce the total amount of acquired data.

4.3.3 Projected content

In the first acquisition scenario the set of images was exactly the same as in 4.2.2. As for scenarios 2 and 4, several video sequences were selected according to its projection content. They belong to the following categories: news, natural and urban scenes, which corresponds to teleconference or movie projection scenarios. Screenshots from the 3 chosen creative common licensed videos are shown in Figure 4.14.

Fig. 4.16 The surface configuration used in the experiments to produce 4-homography distortions in the captured images (scenario 1). Left and right images represent two setup positions.

In scenario 3 several static images were captured and projected by the dynamic setup. In total, 5 frames were used, among them 2 were taken from videos 2 and 3 (see Figure 4.14), and three frames from the set of projected images in scenario 1 and in 4.2.2. Figure 4.15 shows these images.

4.3.4 Database structure

Let us now summarize all the acquisition conditions that were addressed in the series of video-projections and static projections on three and four surfaces. Scenario 1 essentially repeats the acquisitions described in the first part of the chapter (see scheme in Figure 4.10) with the difference that color background was not used at all for the case of 3 and 4 homographies, and 4 homography projections were restricted to be made from two setup positions.

In scenario 2 (Figure 4.17.(a)), video projections were made from 3 static setup positions. Like in the previous acquisitions, besides acquired projected images, the checkerboard was projected for each setup position, and colorchart and white board were captured for each illumination. The same four illumination conditions as in 4.2.1 were employed in this scenario.

In scenario 3 (Figure 4.17.(b)) several frames were projected when the setup was continuously moving. The setup was being moved in two different manners. First, the camera was fixed and the projector constantly changed its position with a high amplitude which resulted in strong projective transformations in the captured images. Projections were made on one,

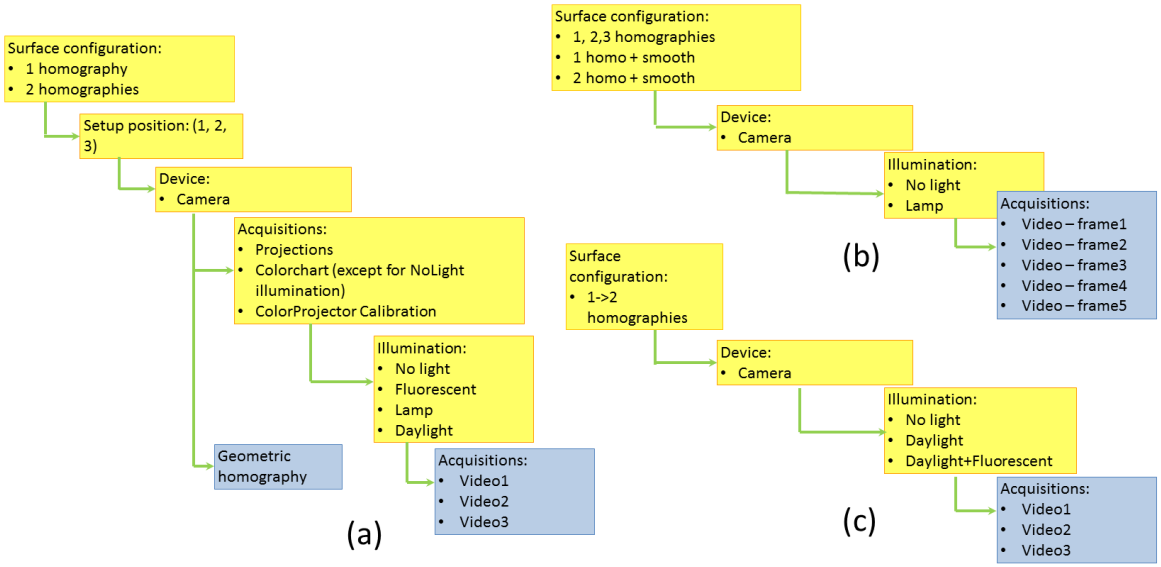


Fig. 4.17 The structure of the acquisitions in scenarios 2 (a), 3 (b), 4 (c). Yellow blocks represent different acquisition conditions and blue block denote acquisition processes

two and three surfaces so that in each acquired video the number of simultaneous projective transforms is constant. To reduce the amount of acquisitions, only two illumination conditions were addressed in this case, that is, dark room and incandescent lamp.

Finally, scenario 4 (Figure 4.17.(c)) represents the most complicated case when videos were projected and acquired with the dynamic setup. During each acquisition series projections underwent both single- and double-homography transformations. Three different ambient illumination conditions were addressed: dark room with the projector light, daylight and the combination of daylight and fluorescent light. The acquisitions were limited to these three configurations because they are most likely to be present in the real situation.

4.3.5 Ground-truth computation

Ground-truth data computation for the first acquisition scenario is the same as in 4.2.3 with the only difference that the number of projective planes is now 3 or 4 instead of single- and double-surface configurations in 4.2.3. It means that RANSAC homography estimation method, previously applied to compute ground-truth double-homographies in 4.2.3, can can be extended so that to estimate a set of planar surface equations and to precise intersection between them. Similar compensation approach can be applied to the acquisitions made in scenario 2, since the setup is static in each series of acquisitions and thus the homography does not change.

Regarding dynamic setup projections (scenario 3 and 4), it becomes difficult to obtain precise ground-truth information for each frame. Projecting calibration checkerboard after each frame might give a precise approximation however at the cost of double reduced frame rate. That is why it was decided to sacrifice per-frame ground-truth geometric information for the acquisition speed. Obviously, one can obtain ground-truth transformations by manually labeling the four corners of the projection area for each frame.

In scenario 3, however, I computed ground-truth in a different way. Instead of obtaining a homography transform for each frame, the projections only for the first and the last frames were compensated and used as the ground-truth data to estimate pixel-wise color distances between images. More details about how ground-truth was used to assess the compensation quality is given later in sub-chapter 5.3.1.

4.4 Conclusions

This chapter presented three datasets of distorted images. In the fist one, synthetic distortions model real photometric and projective distortions typically occurred in a ProCam system. This dataset is used in the next chapter evaluate descriptor performance under various generated distortions. The second dataset, ProCam database, contains real-world projections and covers various conditions, such as varying homography transformations, light sources, and color background. It also provides some data for ground-truth computation. Although ProCam database fully corresponds to only one particular setup, most of the addressed conditions, typical for real ProCam systems, can be encountered in various SAR applications. To prove its usefulness, the following chapter demonstrates two applications aimed at evaluating color descriptors performance for homography estimation and object recognition. Finally, an extension of the ProCam database was presented. Video projection acquisitions were added in the database to make it suitable for different methods that use temporal information between the frames. In the next chapter the prepared video acquisitions are used to evaluate a spatio-temporal compensation method from 3.3.4.

Chapter 5

Evaluations and Applications

After presenting three datasets of synthetic and real-world projections, in this chapter I present several evaluations and applications when these test data are used to assess algorithms performance in the projection-acquisition scenario. The structure of this chapter is, therefore, organized as follows. First, an evaluation of color descriptors for homography compensation is presented. It makes use of the synthetic dataset presented in Section 4.1 to compare descriptor performance. Second, the ProCam database from Section 4.2 is used in two applications: (1) for homography compensation and (2) for object recognition. Finally, an evaluation of two compensation frameworks is presented on a set of acquired video projections described in Section 4.3.

5.1 Preliminaries : evaluation of color descriptors for geometric projections compensation

To examine different characteristics of the state-of-the-art color descriptors from chapter 3, an evaluation framework was developed. The performance of different descriptors was assessed on a set of synthetic images from Section 4.1, that were chosen because of a relatively simple and fast process of test data preparation. This is the first quality evaluation performed on synthetic images before to go to the real-world projection scenarios covered later in this chapter.

This section, first, describes the prepared evaluation framework. Then, the evaluation criteria used in this work are explained. Finally, the evaluation results and shown and each descriptor is discussed giving conclusions on its efficiency if used in a projector-camera system with similar distortions to the ones generated in the dataset.

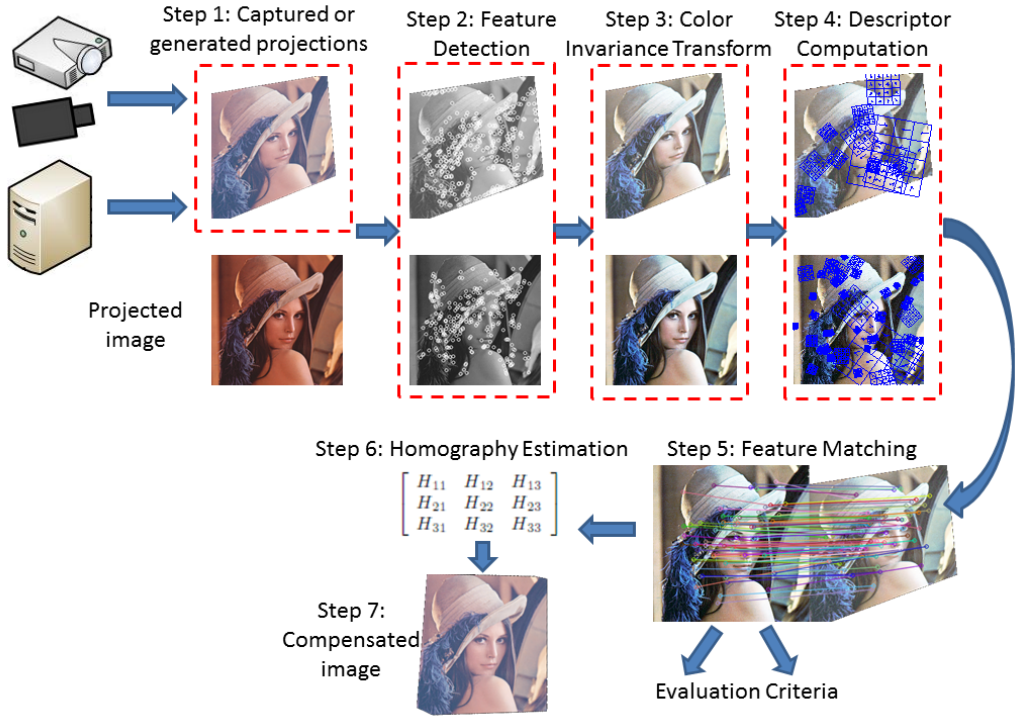


Fig. 5.1 Processing pipeline of Color Descriptors evaluation.

5.1.1 Evaluation framework

Figure 5.1 illustrates the evaluation framework which consists of several main steps.

The first stage consists in preparing a set of test images that includes two parts: original (reference) and distorted (simulated projected) images. It was decided to introduce synthetic distortions to model real photometric and geometric conditions that are typical for a projector-camera system. In order to build a dataset, different synthetic distortions were applied to each reference image (original image to be projected) in order to imitate real acquired projections. Thus, each reference image has a set of distortion copies that are considered as simulated projections. The preparation of the dataset is detailed in 4.1.

The next step implies feature point extraction from the set of reference and distorted images. Features were computed from the intensity channel in order to have the same set of points for all tested descriptors, whatever color invariance is used. In the evaluation framework I included SURF implementation of E. Oyallon and J. Rabin[1] [91] for feature matching. SURF parameters were found by executing the method on 100 training images. The parameters that gave the best performance, were fixed and experimentally kept unchanged

[1]The SURF implementation is available at http://www.ipol.im/pub/pre/69/.

throughout the experiments: 3 octaves, 4 intervals and a Hessian threshold equal to 1050.

Steps 3 and 4 consist in computing color invariant descriptors. This implies applying a color transform to the reference and distorted images and computing SURF descriptors for each color channel independently. The description vectors for each channels are concatenated into one large vector which serves as a feature point descriptor. The outcome of this step is therefore a set of feature descriptors computed for each point.

The next step connotes matching feature descriptors computed from the reference and distorted images. The standard matching strategy, described by D. Lowe [71], defines correct matches based on the nearest neighbor ratio of 0.8 which was used in the experiments. For each color feature descriptor, I computed statistics over all matched image pairs from the test data set. These results were then aggregated in order to capture the general performance of the compared methods. The evaluation criteria are described more extensively in 5.1.3.

In the final step the geometric transform between the matched images is found and the compensation image is computed. In this scenario only single-homography transform was addressed. The 3×3 homography matrix was estimated through RANSAC method as described in 3.3.1 and then the inverted transform was applied to the distorted image to get the compensation. I evaluated the descriptor performance by estimating the errors between the obtained and ground-truth compensations. The process of ground-truth preparation is discussed in the following section.

5.1.2 Evaluated color descriptors

The evaluation framework compares the performance of several color invariant transforms, described in chapter 3, when used for local feature description. First, I choose the traditional Intensity-SURF, and its color variations RGB-, HSV- and Opponent-SURF. Then, I select for the evaluation Color Constancy-based descriptor, which is represented by Retinex algorithm. All aforementioned color descriptor computation methods are described in 3.2.2. Finally, HE-based and LHE-based descriptors, proposed in 3.2.3, are added to the evaluation.

Table 5.1 summarizes the color invariance properties of the addressed descriptors with respect to the color change models used in the evaluation. "+" sign in the table indicates that a descriptor is invariant to color distortions defined by a model, "-" shows lack of invariance, and "+/-" indicates that only part of the descriptor is color invariant. The Retinex SURF is not invariant to the presented color change models because the Retinex theory relies on the

Table 5.1 Color invariance properties of the addressed descriptors

	Diagonal-Offset Model	Gamma Model
I-SURF	+	-
RGB-SURF	+	-
HSV-SURF	+/-	-
Opponent-SURF	+/-	-
Retinex-SURF	-	-
HE-RGB-SURF	+	+
LHE-RGB-SURF	+	+

adaptation rule of von Kries [124] that asserts than the sensitivity adapts to color changes by controlling the amplitude of each spectral distribution. Even though Retinex employs a local approach which can discount non uniform illumination, it is invariant only to contrast changes in the Diagonal-offset model (3.1).

5.1.3 Evaluation metrics

This section overviews several metrics used in the color invariant descriptors evaluation. The performance of the feature descriptors was studied in terms of feature matching quality and homography compensation (warping) accuracy. Two evaluation criteria, used in the experiments, directly reflect the FM quality. The Correct Detection Rate (CDR) and the FM Precision, defined below, reveal two important descriptor properties: distinctiveness and color invariance. The first one shows how many feature matches can be detected using this descriptor, while the second represents the fraction of correct correspondences among all the extracted ones. As for the warping accuracy, I compared the estimated compensations with the images transformed by the ground-truth homographies. In fact, the last criterion is the most important one for choosing a feature descriptor, because it shows how well the method performs in a homography estimation task which is the target application in this work. Nevertheless, the first two metrics helps to understand the performance and the behavior of each descriptor in different testing scenarios.

The definition of each evaluation criterion is presented below.

Detection Rate. This criterion reflects the *distinctiveness* property of feature descriptors. A feature descriptor is distinctive when the similarity measure is high (distance is low) between two homologous points and significantly lower (or higher for the distance) for any other potential match. In order to examine this property of descriptors, *CDR* values were averaged on the whole dataset. The obtained value represents the ability of each descriptor

to produce enough matches under each type of distortions.

$$CDR = \frac{1}{N}\sum_{i=1}^{N+1}\frac{\#\text{ correct matches (i)}}{min(\#fp_1,\#fp_2)_i} \tag{5.1}$$

where N is the number of image pairs, # correct matches (i) is the number of correctly matched feature point pairs in image i, fp_1 and fp_2 denote the two sets of feature points detected from the matched images.

FM Precision. This criterion corresponds to the *color invariance* descriptor property which indicates how much matching quality deteriorates when one or several types of distortions are introduced in the matched images. It can be computed as a ratio between correct and all detected (correct and false) matches in the image pair. To define the formula of FM precision, I first introduce Detection Rate (DR):

$$DR = \frac{1}{N}\sum_{i=1}^{N+1}\frac{\#\text{ computed matches (i)}}{min(\#fp_1,\#fp_2)_i} \tag{5.2}$$

where # computed matches (i) is the number of matched feature point pairs.

Finally, FM precision can be computed as CDR/DR ratio.

Warping accuracy. The two metrics described above were used to investigate the quality of feature matching which, however, is not the objective of this work. To evaluate the descriptor performance in geometric compensation a third metrics was introduced, called *Warping accuracy*. It can be computed in a simple way by comparing the obtained and ground truth compensation images. Ground-Truth homography matrices H are given since the geometric transforms are simulated. For each point $[x\ y]$ in the first image the corresponding point in the second image can be obtained through a simple multiplication $[x\ y\ 1]\cdot H$. Therefore, the *Warping Accuracy* is computed as the average Euclidean distance between pixel coordinates of the corrected image and the corresponding ground-truth image.

$$d(H_e,H_r) = \frac{1}{W\times H}\cdot\sum_{\substack{X_e=0,Y_e=0\\X_r=0,Y_r=0}}^{W,H}\sqrt{(X_e-X_r)^2+(Y_e-Y_r)^2} \tag{5.3}$$

where W,H are the width and height of the images; H_e and H_r are 3×3 estimated and ground truth inverse homography matrices. This criterion was only computed for synthesized distortions (*S1*-*S4*).

5.1.4 Evaluation results

The evaluation results are presented and discussed below. I start from synthetic distortions that model simple single-homography transformations, two color change models and Gaussian noise (*S1* to *S4*). To conclude, the results for *S5*-*S6* distortions are given.

Simulating the projection on an ideal surface

Tables 5.2, 5.3 and 5.4 show the descriptor performance results computed for the case of synthetic distortions *S1*-*S4*, when classical color distortions and homography warping were performed on the reference image. Here, only illumination changes are simulated and not the complex photometric variations due to the projection on a color surface. It is important to mention that the applied color distortions are spatially uniform in the images, which is favorable for most of the evaluated descriptors. For the sake of compactness, I excluded HSV-SURF results from the tables because this descriptor showed the worst performance among the competitors in each case. In fact, only the H channel is invariant to light color changes, what is not the case for the V and the S channels. Furthermore, the descriptor is instable in the Hue channel for low saturated colors.

Observing the Correct Detection Rate (CDR) values in Table 5.2, two main conclusions can be made. First, Intensity-based descriptors are, in general, more distinctive than their color analogs, which is proved by lower average ranks. Second, the I-HE descriptor is the best one since almost 48% of the feature points are correctly matched. This is due to the conjunction of three factors: 1) the impact of the color changes is minimized through the use of a single channel instead of three; 2) the SURF feature point is already locally invariant to uniform contrast changes; 3) the HE provides invariance against monotonous variations when these variations are spatially uniform, which is exactly the case here.

Concerning the *FM Precision* results in Table 5.3, RGB-based methods have a clear advantage over Intensity descriptors, whatever the distortion type. It means there are less matches when using color, but these matches are more reliable. Thus it can be concluded that color increases the discriminative invariance power of the descriptor. Opponent, while providing low distinctiveness (it ensures less matches which is marked by low *CDR* values), is quite robust to the distortions defined by the Gamma color model in which it produces strong *FM precision* results. Regarding Retinex, it provides poor results when geometric distortions occur (*S3* and *S4* images). As it was previously mentioned, test images underwent uniform distortions. This explains the fact that in many cases HE and LHE do not introduce any improvement in the results. Therefore, homogeneous and monotonic color

distortions correspond to a favorable case for most color descriptors. The case of inhomogeneous color distortions will be encountered and in the next chapters on real-world images.

The compensation errors (*Warping accuracy*), presented in Table 5.4, demonstrate that good feature matching performance does not necessarily improve the homography estimation. Most of the considered descriptors yield similar results considering the average error and the average rank. For the intensity-based descriptors it can be due to their better *CDR* results, which means that more points are detected in general, compared to the color methods. RGB-based descriptors, on the contrary, produce less matches but most of them are correct (high *FM Precision* values).

One more factor complicates direct relationship between feature matching and geometric compensation results. RANSAC, used for homography transform estimation, is robust itself to outliers. Even though some descriptors show worse feature matching performance, thanks to RANSAC outlier points filtering, precise compensation (low *Warping accuracy* values) can be computed. Nevertheless, the obtained statistics on feature matching results can give important clues to understand each descriptor performance in each condition.

Overall, from the results obtained for *S1-S4* distortions, I recommend I-HE descriptor as the most suitable for the tested scenario. This descriptor is supported by its good compensation performance (see Table 5.4) and its lower complexity, as compared with color invariance and I-LHE methods.

Another important observation can be made for the Gaussian noise influence on the descriptor performance. LHE-based descriptor is more sensitive to the noise than Intensity and RGB descriptors. It happens because rank ordering is corrupted by the noise and thus cannot be preserved which is the condition for its good performance. as previously stated in 3.2.4.

Simulating the projection on a colorful and rough surface

In this section I focus on the results obtained on *S5-S6* test sets, prepared through normal mapping and blending with color background. Contrary to the previous scenario, the photometric variations involved here are complex because the reference image is mixed with bump light map and a color background. It aims to well model the complex photometric changes that can appear when an image is projected on a non-ideal surface, possible colorful and textured. To the best of my knowledge, no color descriptor has been defined specifically to deal with such distortions.

The averaged over the number of test images [2] *CDR* and $FMPrecision$ errors are col-

[2] The number of images are outlined in Table 4.1.

Table 5.2 Feature matching results (*Correct Detection Rate CDR*) in the case of synthetic photometric variations (*S1* to *S4* test images).

			CDR (%)							
			I	I-HE	I-LHE	Opp	Rtnx	RGB	RGB-HE	RGB-LHE
Synthetic images database	*S1*	Gamma	83.11	83.76	83.76	79.79	84.19	83.73	84.08	**84.22**
		D-Off	72.62	**74.49**	74.20	36.54	63.28	69.50	71.40	68.82
	S3	Gamma +Homo	26.73	26.56	25.83	**28.44**	24.17	28.28	27.97	27.11
		D-Off +Homo	26.00	**26.16**	25.38	12.04	20.02	25.98	25.64	24.11
	S2	Gamma	59.92	**61.12**	57.16	40.37	54.08	55.71	56.28	52.82
		D-Off	51.94	**54.43**	49.61	18.18	40.63	43.43	44.49	40.95
	S4	Gamma +Homo	22.79	**22.85**	20.33	16.62	18.15	21.99	21.60	19.78
		D-Off +Homo	19.97	**20.55**	18.15	6.61	14.09	17.58	17.18	15.46
Average			45.38	**46.24**	44.30	29.82	39.83	43.27	43.54	41.66
Average Rank			3.1	**2.0**	4.0	7.1	6.4	4.1	3.9	5.2

Table 5.3 Feature matching results (*FM precision*) in the case of synthetic photometric variations (*S1* to *S4* test images).

			FM Precision (%)							
			I	I-HE	I-LHE	Opp	Rtnx	RGB	RGB-HE	RGB-LHE
Synthetic images database	*S1*	Gamma	99.56	99.55	99.53	99.74	99.90	99.53	99.58	**99.92**
		D-Off	99.03	99.14	99.25	98.79	99.32	99.24	99.31	**99.46**
	S3	Gamma +Homo	93.41	92.97	93.50	**97.13**	95.21	96.08	95.99	96.07
		D-Off +Homo	91.12	90.52	91.03	92.03	92.04	**94.30**	93.70	93.67
	S2	Gamma	98.96	98.92	98.93	**99.53**	99.39	99.30	99.17	99.36
		D-Off	97.96	98.30	98.30	97.78	**99.05**	98.99	98.62	98.96
	S4	Gamma +Homo	91.83	92.02	91.72	**96.98**	95.17	95.52	95.28	95.66
		D-Off +Homo	88.90	89.70	88.86	90.78	91.64	**93.56**	92.57	93.32
Average			95.10	95.14	95.14	96.59	96.47	**97.06**	96.78	97.05
Average Rank			6.3	6.6	6.5	3.9	3.1	3.1	3.6	**2.3**

Table 5.4 Homography compensation results (*Warping Accuracy*) for *S3* and *S4* synthetic test images.

			Warping Accuracy (pixels)							
			I	I-HE	I-LHE	Opp	Rtnx	RGB	RGB-HE	RGB-LHE
Synthetic images	*S3*	Gamma	3.17	1.51	1.41	1.47	1.62	1.53	**1.38**	1.28
		Diag-Off	3.58	3.61	**3.52**	50.48	7.45	11.58	3.67	3.80
	S4	Gamma + noise	2.06	2.05	2.35	3.50	5.25	1.98	**1.94**	2.18
		Diag-Off + noise	4.61	**4.28**	5.68	192.02	32.08	9.43	21.36	18.88
Average			3.36	**2.86**	3.24	61.87	11.60	6.13	7.09	6.54
Average Rank			3.75	**3**	3.25	6.75	7	4.75	3.25	4

lected in Table 5.5. For the sake of compactness, only the four top performing descriptors are displayed in this table.

First of all, in case of pure geometric distortions (dataset *S5*), HSV descriptor demonstrates a fairly good performance. High *CDR* and *FM Precision* values drop significantly when color background blending is introduced. As for the Opponent-SURF, under high color distortions it produces the most number of feature matches and they are more robust. Indeed, this descriptor has the advantage to provide pixel-wise color invariance, whereas RGB-LHE requires a windows of interest around the point, as explained in chapter 3. RGB-LHE is less distinctive, producing less matches, but with high precision. RGB-SURF, compared to its LHE version, produces more feature points but they are less precise. When background is uniformly colored, HE-based descriptors together with the others perform well.

From the presented results for *S5-S6* datasets it appears that the Opponent descriptor is the best one in this evaluation. The use of RGB-LHE can also be promising because, even though it does not outperform Opponent, it presents a good trade-off between distinctiveness and robustness.

Table 5.5 Feature matching results (*Correct Detection Rate CDR* and *FM Precision*) on synthetic images generated with normal mapping and background blending (*S5-S6* test images).

			CDR (%)				*FM Precision* (%)			
			HSV	Opp	RGB	RGB-LHE	HSV	Opp	RGB	RGB-LHE
Synthetic images	*S5*	Normal Mapping	**27.37**	27.35	25.11	23.44	**98.69**	97.87	97.99	98.08
	S6	Normal Mapping + BG Color	3.12	**10.63**	9.06	8.70	64.28	**89.16**	86.97	87.65

5.1.5 Discussion

To sum up, below I give brief inferences on the use of each descriptor, derived from the presented study.

- HSV is advantageous when no color distortion is present;
- If the projected surface is uniform and planar, Intensity-based methods produce a lot of matches (high *CDR*) which results in a good geometric compensation (*warping accuracy*), despite low *FMPrecision* values. RANSAC is capable of filtering out false matches, provided that there are enough correct correspondences. When projections are blended with color background, the use of Intensity-based descriptors is not however recommended;
- Because most of the produced distortions are global and uniform in the image, LHE does not provide any improvement. In case of projections on colorful surfaces, the results can be slightly improved. When the images are noisy (Gaussian noise applied), the results deteriorate because in this case the rank color preservation is no longer assured;
- Notwithstanding very good *CDR* and *FM precision* results in some cases, Opponent can completely fail at producing feature points matches leading to inaccurate homography estimations. It can be explained by the losses in distinctiveness when the matched images undergo large color variations. Thus, this descriptor is efficient only for moderate color distortions.

  If color distortions are moderate, the use of this descriptor can be advantageous;

- RGB-LHE appears to be the most stable descriptor in the sense that its results do not drop significantly whatever the types of color and geometric distortions. Thus, this descriptor is recommended when there is no prior information about the color distortions occurred in the matched images.

From the experiments made on synthetic and real test data, it arises that it is sometimes difficult to make a logical connection between the feature matching quality (*Correct Detection Rate CDR* and *FM precision*) and the corresponding geometric compensation results (*Warping accuracy*). Nevertheless, it was seen that having more feature matches (even at the cost of low precision) leads to good homography compensations by RANSAC. In this sense, high *CDR* values are important. On the contrary, when $FMprecision$ results are very poor, which means that feature matching fails on test data, it is very unlikely that RANSAC can correctly estimate the homography.

After presenting the evaluation of the descriptors on the synthetic data set, we now turn to the analysis of the descriptors performance on real-world projections.

5.2 Contributions for ProCam systems : static scenario

In this scenario different algorithms were tested on a set of static acquired image projections from the ProCam database described in 4.2. There are two experimental applications presented in this work. The first one is similar to the color descriptor evaluation on the synthetic dataset presented in 5.1, with the difference that the comparison of compensation methods was performed on real-world projection images. Moreover, the case of double-homography compensation was addressed in this evaluation. The second application consisted in performing object recognition from captured projected images. The problem is more complicated than classical object recognition because test images undergo more severe photometric distortions, typical for projector-camera systems, and thus significantly differ from the images used in training.

5.2.1 Geometric compensation of static images

In this first application the ProCam database was used to evaluate the performance of several homography compensation methods. Several state-of-the-art algorithms for image stitching described in 3.3.3 were considered. They are evaluated and improved in this chapter for the use in a projector-camera system.

Evaluation Framework

The objective of this framework is to evaluate several homography estimation methods in the projection-acquisition context. First, I included in the evaluation two image warping methods for stitching, described in 3.3.3, as they are. However, these methods are not adapted to deal with high photometric distortions, which occur in acquired ProCam system projections. Indeed, in most image mosaicing applications the photometric differences between the processed images are not very high and, thus color variation is not addressed explicitly. In this evaluation to make the methods be color invariant I applied Global Histogram Equalization (HE or GHE), described in chapter 3. This method was shown to provide color invariance to various uniform photometric changes. Local Histogram Equalization (LHE), an improvement of GHE, was not used here because it is not implemented in the VLFeat library and thus it would have required a significant modification of the considered methods. Finally, to be consistent with the previous work, I added GHE-based compensation [108] evaluated in 5.1.1. To resume, 5 methods were evaluated in this framework: two of them from stitching, their color invariant modifications and GHE-based homography estimation from the previous evaluation, hereafter denoted as Setkov2013. All the methods are listed again in Table 5.6. The first two works, marked with "-" in this table, are not color invariant, whereas the remaining three methods, marked with "+", employ a color invariance transform at the pre-processing step.

Method	color invariance
Chang2014 [21]	-
Zaragoza2013 [21]	-
Chang2014+GHE	+
Zaragoza2013+GHE	+
Setkov2013 [108]+GHE	+

Table 5.6 The methods for homography estimation used in the evaluation.

As stated in 5.1.4, histogram equalization is very sensitive to the noise which appears when background is included in the equalization image area. To that end, I deliberately selected the projection area which could be easily done thanks to ground-truth data provided in the database. Therefore, I ensured that only the area of image projection was equalized. Some examples can seen in Figure 5.2. Although in a real application the projection area should be detected in one way or another, this problem was considered as beyond the scope of this evaluation.

Two metrics were used in the evaluation. The first one, denoted as *color error* corre-

Fig. 5.2 Examples of projected image with Histogram Equalization applied on the projection area.

sponds to the quality of the compensated image and can be computed as the Root Mean Square (RMS) Error between the images warped by the ground-truth homography and the ones estimated by the evaluated method. This criterion is very important for SAR applications because it shows how close the compensation is to what can be ideally obtained. The second metric, called *geometric error*, reflects the quality of the estimated homography transformation and can be formulated as the RMS Error between the (x,y) image coordinates warped by the estimated and ground-truth homographies. This metrics could be of interest for such applications as image stitching and 3D reconstruction.

Evaluation Results

Figure 5.4 shows the photometric and the geometric errors of the compared methods obtained on all acquired projections from the database without color background. To make the curves more comprehensible, the cumulative distribution function (CDF) of the errors is displayed. The values along the x axis represent the compensation errors and the y axis represents the percentage of the matched images that have errors below the abscissa value. For the sake of clarity, the curves are displayed for error values lower than or equal to 20. Observing the compensated images, it turned out that the compensations with the errors higher than 20 are too much distorted to be used as projection corrections (see error examples depicted in Figure 5.3). This error value was empirically obtained through an expert evaluation of three people and is only applicable to the used evaluation setup. This error may change for different image content, image resolution, or projection and acquisition devices.

Because all the CDF curves are monotonous, the relative performance of the methods does not change on the non-displayed parts of the plots. The two top plots in Figure 5.4 correspond to the case of a single-homography transformation, whereas the bottom plots show the results of double-homograhy compensation. The left graphs show the photometric errors and the left ones represent the geometric errors.

The main conclusion that can be drawn from the results is that color invariance when applied to the homography compensation methods, significantly ameliorates the compensation results. In case of a single homography, however, only the method of Chang is improved by 10.94% for photometric error and by 13.92% for geometric compensation. For double-homography transformation the improvement is strong for both methods: the color error is descreased 8.79% and 5.46% and the geometric error is decreased by 17.82% and 6.24% for Chang and Zaragoza methods respectively. These values were computed as the average differences between the results curves with and without HE for each method.

For single-homography transformations, the method of Chang with HE produces better results according to the RMS color errors and performs similarly to the competitors in terms of geometric errors. This method in general computes less matches that can be significantly improved by HE. The method of Zaragoza, on the contrary, yields enough matches and, thus, the improvement is not significant when HE is used. GHE based method is outperformed by the other methods in terms of geometric error, while the color errors are similar for many methods.

Observing the results computed in the case of double-homography, it can be seen that Zaragoza+GHE method outperforms Chang+GHE on the whole range of errors. Both methods are substantially improved by the introduced color invariance. To estimate two homography transforms in one image, not only more correct matches are required, but also a better spatial distribution is needed so that the number of matches are approximately balanced between the two projection surfaces. To finish, GHE provides a good trade-off because it produces very good geometric transform estimations and more stable results for high error values.

5.2.2 Object Recognition

The second application consists in performing object recognition on acquired projected images. As it was explained in 2.2, the acquired images undergo various geometric and photometric distortions due to the devices, the illumination, and the projection surface. All this can complicate object recognition, if the classifier is trained on reference images, while

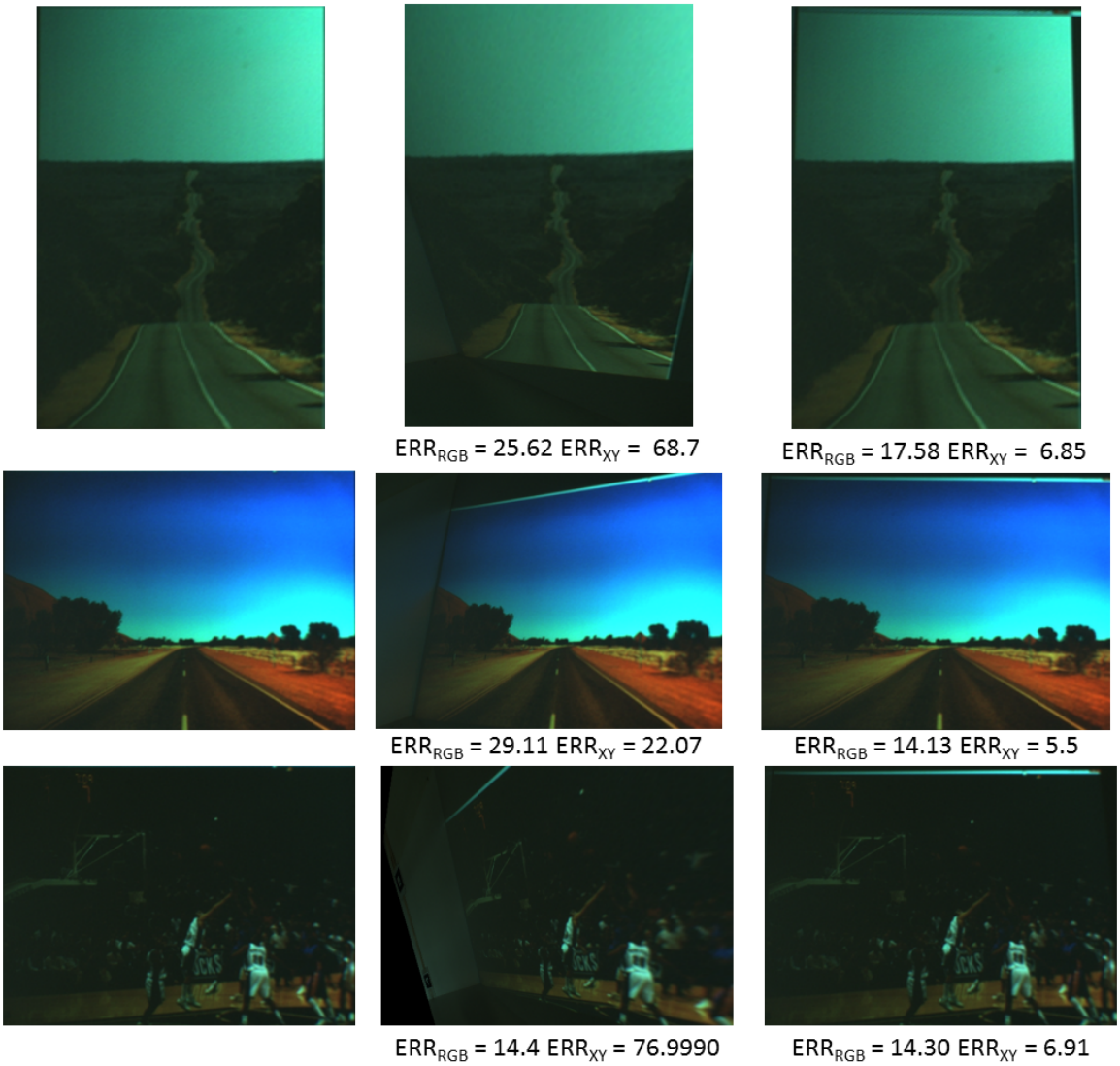
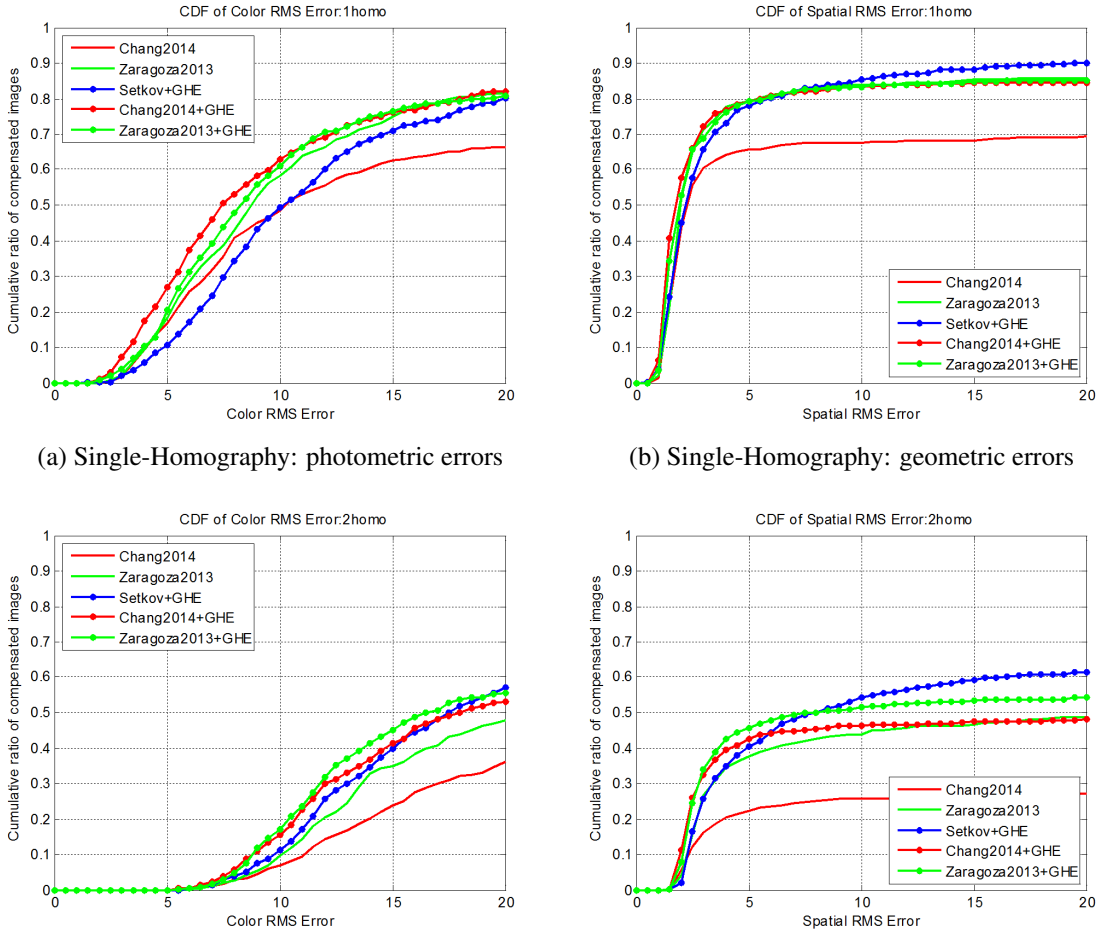


Fig. 5.3 Homograhy compensation examples with the corresponding color and geometric errors. Ground-truth compensations are shown in the first column.



(a) Single-Homography: photometric errors

(b) Single-Homography: geometric errors

(c) Double-Homography: photometric errors

(d) Double-Homography: geometric errors

Fig. 5.4 Compensation results

the recognition is done on acquired projections. This is a more difficult problem than the classical one due to the complex photometric distortions between test and training images. This new application might be of interest for such applications as home video projections or public presentations. In the first case, it can be used to recognize actors or movies, and in the second case it can recognize objects from images on presentation slides. This work was performed in collaboration with Fabio Martinez Carillo.

Object Recognition Framework

In this evaluation I examined the performance of several local descriptors applied to object recognition in a Bag-of-Words framework. A part of the database was used which corresponds to face images (for details refer to 4.2.2). The reference images were taken from the

McGill Real-World Face Video Database[3] [28]. It includes, in total 300 images for each of 8 categories. Each category corresponds to one person that is present in different images in which the pose and background varied significantly. In total $300 \times 8 = 2400$ images were used as training data.

In the image projections database only 10 images of each category were projected and acquired under different conditions. These images were exploited to test the trained recognition model. In fact only a part of the acquisitions was selected that corresponds to a single setup position and three different illuminations: daylight, fluorescent and incandescent lighting. In total, for each category, 30 images (10 projections under 3 illuminations) were retained for testing.

Several local descriptors were examined in this recognition framework. I-SIFT, RGB-SIFT, LHE-RGB-SIFT and PHOW-SIFT are the methods included in the evaluation. All of them were implemented in Matlab by means of VLFeat library [120][4]. PHOW-SIFT, a multi-scale version of Dense-SIFT, was set to compute descriptors at every 32-th pixel. It was done in order to keep the total size of feature descriptors comparable with the other methods and to reduce the execution time.

The rough projection images taken from the database represent a difficult case for feature matching, because the projection area is not selected and photometric distortions are high. In the evaluations three scenarios were examined. First, object recognition was performed on test projection images as they are. Then, with the aim of improving the recognition rate the projection area was preselected to exclude most of the background from the feature extraction. This preselection was done manually so that projections occupied approximately $1/3$ of the image surface. Finally, to reduce the affect of color distortions in the acquisition, a simple preliminary color compensation procedure was applied. It took into account only camera distortions and did not address either projector response, or ambient illumination. The gamma parameter value of 2.2 was roughly estimated from the acquired colorcharts (refer to 4.2.2 for more details about the acquisitions) and averaged for the R, G and B channels. Then, a simple white balance correction was applied by multiplying the green channel by 0.8, which corresponds to the Balance Ratio parameters in the camera. Finally, intensity factor of 0.8 was used in order to avoid saturations. The formula below presents

[3] https://sites.google.com/site/meltemdemirkus/mcgill-unconstrained-face-video-database

[4] an open source library VLFeat includes implementations of various computer vision implementations http://www.vlfeat.org/

Fig. 5.5 Example of test images used in BoW object recognition framework. (a) - an RGB image provided in the database; (b) - the cropped image; (c) - the cropped and photometrically corrected image.

the transformation as a whole:

$$\begin{pmatrix} O_R \\ O_G \\ O_B \end{pmatrix} = 0.8 \cdot \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix} \right)^{1/2.2} \tag{5.4}$$

where $I_{R,G,B}$ and $O_{R,G,B}$ correspond to the initial (input) and the corrected (output) images.

Figure 5.5 illustrates test images from three different scenarios used in object recognition.

Object Recognition through Bag-of-Words model

Bag-of-Word (BoW) model is among state-of-the-art approaches used for object recognition [32, 60, 61]. To prepare the evaluation framework, I made use of this model, which is based at the lowest level on local point description. In this experiment I evaluated the performance of several descriptors and, in addition, analyzed the performance of object recognition in the projection-acquisition context.

The BoW model works as follows. First, local features are extracted and described from the training images. Generally, SIFT-like or other robust local descriptors are used to describe visual information in the images. The next step consists in constructing a vocabulary. To do so, detected descriptors (samples) are clustered using a classical K-means approach. Each cluster center is hereafter coded as a visual word. In the evaluation a random selection of 10% of the samples was used. The size of each vocabulary was set according to the number of detected input samples, as: $k = \sqrt{(S/2)}$, with S the number of computed input samples. Then, a projected image is represented as a histogram of visual word occurrences. Each input sample is assigned a visual word that produces the lowest Euclidean distance on

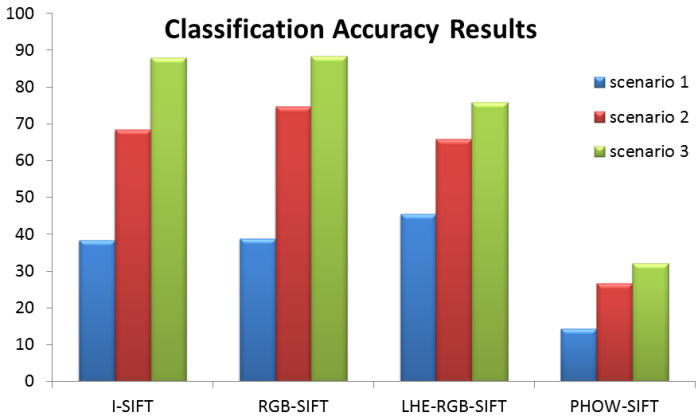


Fig. 5.6 Classification accuracy results obtained in the three test scenarios.

$\mathbb{R}^n$, where is n is the size of each descriptor.

Finally, once the vocabulary is built and all the images are represented by their histograms, the recognition of each category was carried out by a Support Vector Machine (SVM) using the standard LIBSVM [20] implementation, using the *one-against-one* multi-class SVM classification with a Radial Basis Function (RBF) kernel.

Results

To assess the quality of object recognition for each method I computed *classification accuracy*, which is a typical measure for multi-class recognition. It can be computed as the average of the diagonal elements in the confusion matrix. Figure 5.6 shows the *classification accuracy* results obtained by the evaluated methods on a set of image projections for the three tested scenarios. It can be seen that the combination of projection area selection and color correction significantly improves the recognition rates. Thus, it is highly recommended to perform this type of compensation prior to performing object recognition in the case of captured projected images.

By observing the performance of color descriptors, some conclusions can be made. First of all, PHOW-SIFT performs very poor because dense SIFT point sampling is very sensitive to the projection scene background. Even in the cropped images (scenario 2) there remains some background around the projection area which introduces a noise in the recognition. Next, LHE-RGB-SIFT descriptor performs well only on unmodified projection images. It means that this descriptor is more robust to the presence of background. When image correction is introduced (scenarios 2 and 3), I-SIFT and RGB-SIFT yield the best results. When color correction is not performed (scenario 2), RGB-SIFT outperforms all the methods,

which can be explained by its ability to better cope with color distortions than I-SIFT does. Partially it was shown by *CDR* values in Table 5.3.

Figure 5.7 shows the confusion matrices for all the methods except for PHOW-SIFT that was omitted because its poor performance. From the matrices it can be seen that the recognition framework tends to misclassify images of most of the classes as "person5" category. Some difficulties are also introduced by classes "person7" and "person8". It can be due to similar backgrounds in the acquired images and in the reference "person5" images. Color correction and projection area selection can solve this problem.

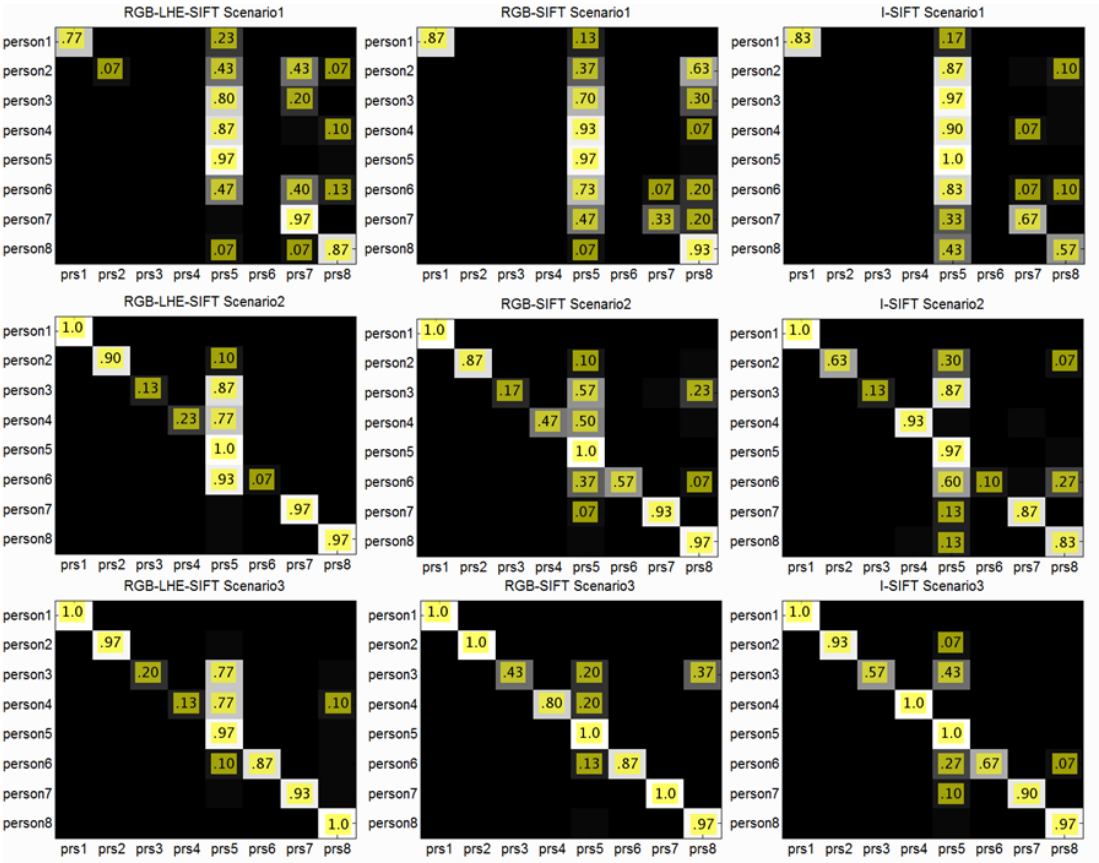


Fig. 5.7 Confusion matrices of RGB-LHE-SIFT, RGB-SIFT and I-SIFT.

Discussion

In the presented evaluation only several local descriptors were evaluated. Consideration of state-of-the-art BoW recognition works was left out of the scope of this work. To be consistent with the previous evaluation, Intensity and color descriptors as well as HE-based color invariant method were compared. Because multi-scale dense SIFT is frequently used

in object recognition, this descriptor was also included in the evaluation.

Another objective of the evaluation was to try previously tested descriptors in a new application framework for object recognition in the case of a projector-camera system. This first attempt opens new avenues for further investigations. Moreover, it shows another useful application of the ProCam database to object recognition.

5.3 Contributions for ProCam systems : dynamic scenario

The last part of this chapter presents a series of experiments directed at evaluating the compensation system described in Section 3.3.4. This system uses a combination of feature matching and optical flow tracking methods in order to temporally compensate image projections.

Prior to introduce the experiments made in this evaluation, several additional conditions that are not addressed in the block diagram of Figure 3.13 are presented. Depending on the nature of the video content, Optical Flow tracking can perform differently. If the motion is fast, some of the tracked feature points can go outside the image boundaries and thus disappear from the analyzed set of points. The decreased number of feature points can degrade the compensation quality, because the points might then become less well distributed spatially. To address this problem, in the FM-OF method I reiterate the Feature Matching procedure when the ratio of the number of currently available points to the initial number of points goes below the 0.7 threshold. This value empirically showed good results in the evaluation with video projections. High ratio values lead to a more frequent FM execution and, thus, to a more precise estimation with an increased run time. On the contrary, a low threshold tolerates the feature points reduction and, thus the quality degradation, while making the compensation faster.

Another point that needs to be addressed corresponds to the frequency of FM calls. OF easily fails at tracking the points on a long sequence of images. The compensation error gets accumulated passing from frame to frame. To alleviate this problem I employed a hard threshold that corresponds to the maximum number of compensated through OF images without running FM. In the experiments I chose this value to be 30, as it allows low distortions in compensation while keeping the results and the execution time reasonable.

5.3.1 Experimental data

The aim of this evaluation is to compare the performance of the proposed FM-OF compentasion method with the traditional per-image FM approach. Both compared methods were executed on several video acquisitions presented in Section 4.3. First test scenario implies compensating a video sequence projected by the static setup, that is with a constant homography transformation. In this case ground-truth transformation is provided along with the tested video acquisitions. The second scenario corresponds to dynamic projections of video sequences.

The evaluation was performed in terms of compensation quality and execution time. For the static projections scenario, the quality was assessed in terms of RGB color and XY-coordinates distance between the ground-truth and the computed compensations. The average per-pixel square distances are further shown. As for execution time, the time performance was measured and compared between FM-OF and FM compensation methods.

In the case of dynamic video projections, ground-truth transformations are not provided in the database, but can be computed by manually labeling the four corners of the projection area in order to get the precise homography transformations of each projected image. However, in this evaluation a different strategy was applied to assess the methods. I made use of the ground-truth compensations from the static projection scenario of the same video to have a corrected image for every projected video frame. I compared them with the compensations estimated from dynamically projected video acquisitions. Because the photometric differences are quite high between the dynamic and static projections due to inter-reflections, the projector distance to the surface, etc, a simple Gray World color compensation was employed to make the results closer. Gray World assumption [15] argues that the average reflectance of the scene is close to achromatic, *i.e.* the mean red, green and blue values are almost equal. Having all this said, the following transformation was applied to the obtained geometric compensations before to compute the errors:

$$O^i = 128 \cdot I^i / \overline{I^i} \tag{5.5}$$

where i indicates one of the R,G,B image channels, I and O are the images before and after color correction. Here the image intensities are assumed to be in the range from 0 to 255.

Figure 5.8 shows examples of geometric compensation results with and without color correction, as well as the corresponding error values. It can be seen that in some cases it improves the results when photometric differences are high between the static and dynamic acquisitions. It also better highlights the differences between the compensations that, due

to low intensity, can seem quite similar despite different applied homography transforms. It should be noted again, that in these experiments I was not interested in the absolute compensation errors, but in the relative performance of the compared methods which was well captured.

5.3.2 Results

This section presents the experimental results obtained on several videoprojections performed by the static and the dynamic ProCam system. These videos correspond to acquisition scenarios 2 and 4 in Section 4.3. The performance of the two compensation was compared using several evaluation metrics. Furthermore, the execution time of the compensation framework was measured.

Static Projections

Figure 5.9 shows the compensation results obtained using FM-OF (red curves) and FM (green curves) approaches. For the sake of simplicity, video sequences "Nature", "Street" and "News" are denoted as video1, video2 and video3 (see Figure 4.14), respectively. For each video three plots are presented. The top one corresponds to the Root Mean Square Errors ($RMSE_{XY}$) of the pixel coordinates between the computed and ground-truth compensations. This measure shows the warping accuracy of the obtained compensation. Similarly, the second plot represents the RMSE of RGB compensation image values ($RMSE_{RGB}$), which represent visual differences between the original image and its projection. Finally, the third plot depicts the number of feature point pairs computed or tracked for each frame. This data give more clues on the performance of each compensation method at each frame. Blue timestamps indicate when FM was executed inside the FM-OF method. Table 5.7 summarizes the average RMSE values for each video.

Observing the results, one can make several conclusions. The compensation error produced by OF, when applied to compensate images, constantly grows over time with the increase of the number of processed frames. An example of such error propagation is depicted in Figure 5.10. However, if FM interlaces from time to time OF tracking to introduce corrections in the feature points vectors, the error growth can be moderated. Figure 5.11 shows several examples of image compensations[5]. Depending on the projected content, the difference in performance between the compared methods can vary significantly. More pre-

[5]Sample videos illustrating how the FM-OF method works can be downloaded at: https://perso.limsi.fr/setkov/demoVideos

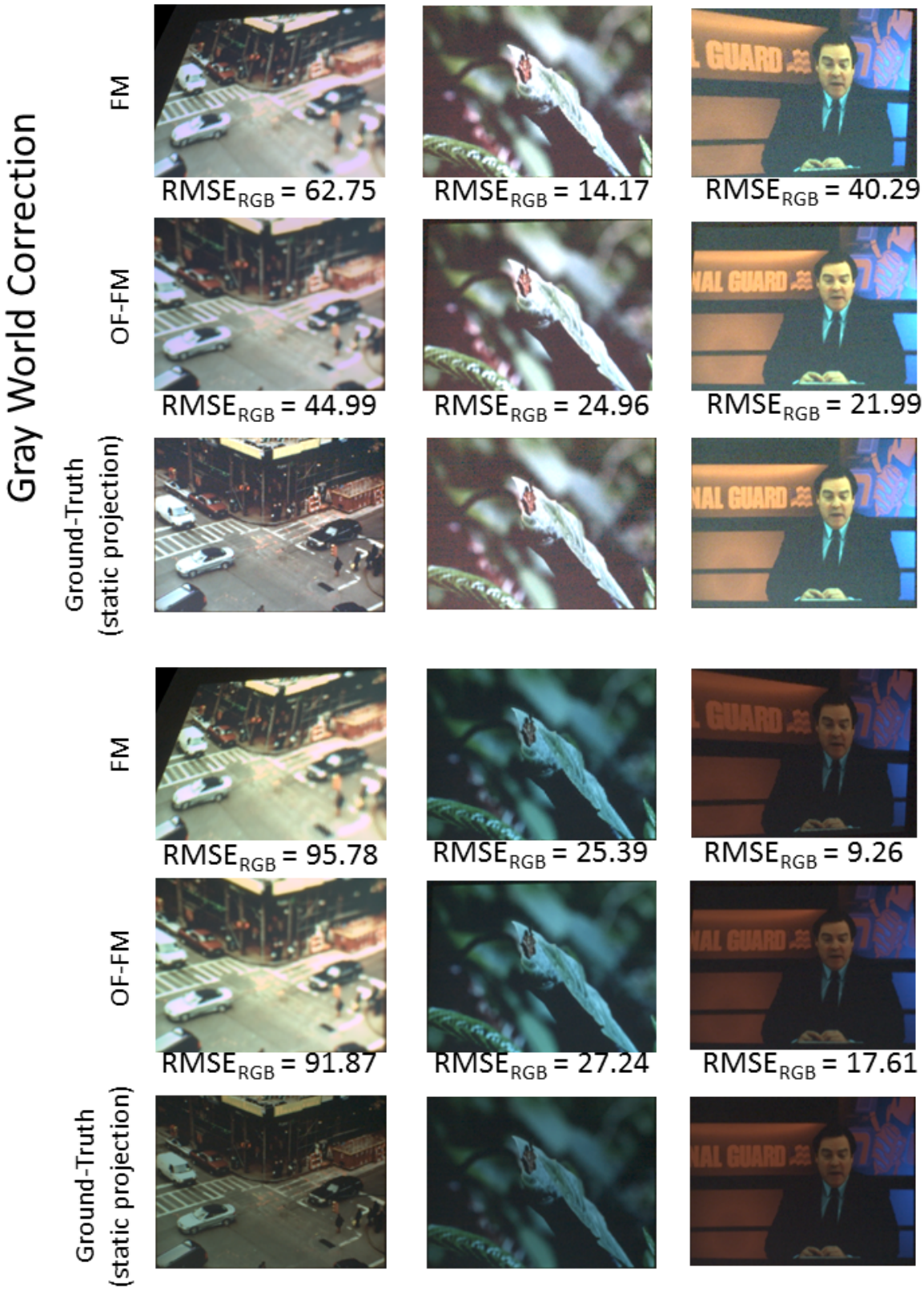


Fig. 5.8 Examples of the compensated images obtained through FM and FM-OF methods in the dynamic projection scenario with and without Gray World color correction. The third and the sixth rows correspond to ground-truth data based on corrected static projections of the same video sequence.

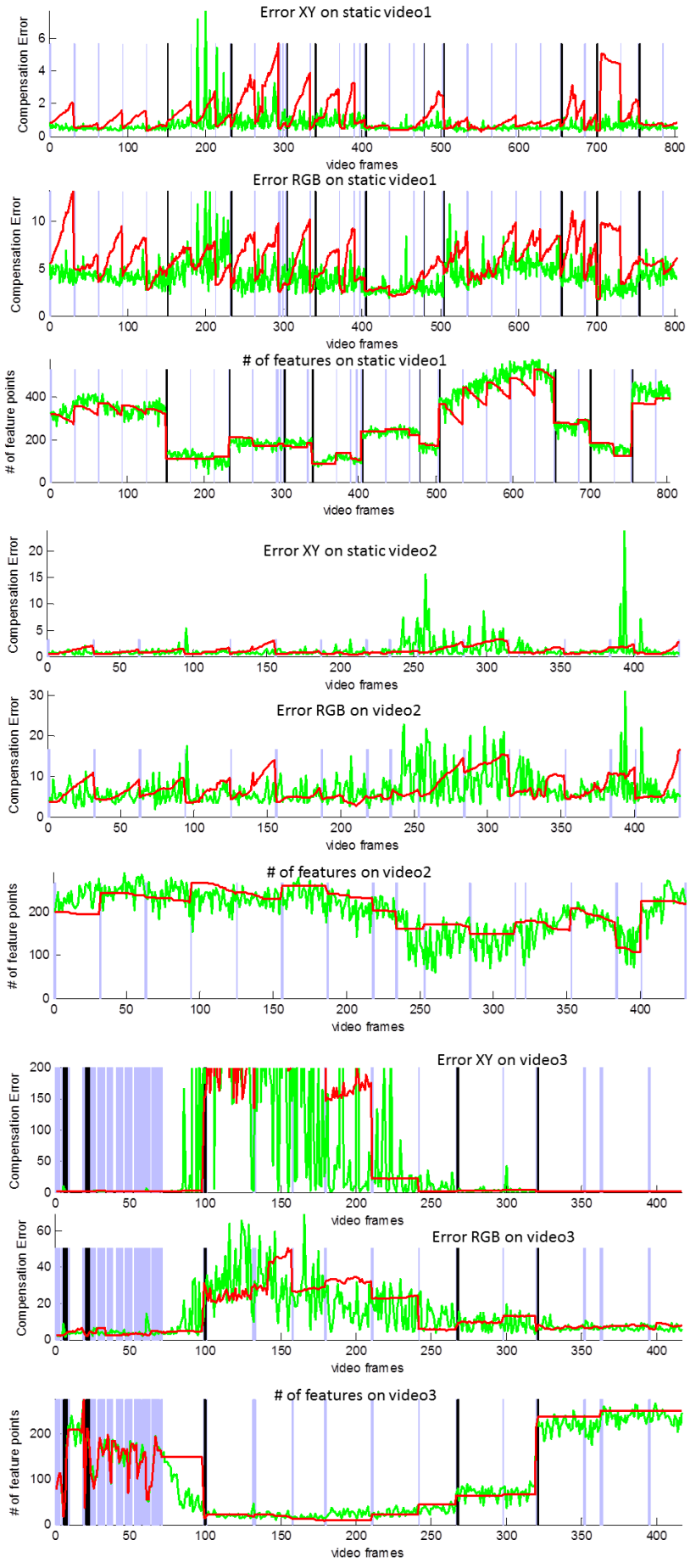


Fig. 5.9 Compensation results obtained through FM-OF method (red curves) and FM method (green curves) on statically projected and acquired videos. Blue timestamps in-

Fig. 5.10 Example of compensation error propagation by the OF-FM method. Frame t is the compensation estimated at frame t, frames $t+10$ and $t+20$ are the compensations estimated after OF tracked feature points during 10 and 20 frames, respectively.

method	video1		video2		video3	
	$RMSE_{XY}$	$RMSE_{RGB}$	$RMSE_{XY}$	$RMSE_{RGB}$	$RMSE_{XY}$	$RMSE_{RGB}$
FM-OF	1.30	5.85	1.08	7.11	245.30	14.25
FM	0.69	4.19	1.16	6.72	85.82	13.19

Table 5.7 Compensation performance of the tested approaches in the static scenario.

cisely, on video1 there is almost twice difference between the compensation results of FM and FM-OF methods. In fact, this video is difficult for OF because of many scenes with small deformable objects. Increasing the frequency of FM calls can improve the overall quality of FM-OF compensation. On video2, OF in many cases improves the compensation. This video is more favorable for OF because the objects are rigid and can be easily tracked. The third video3 represents the most difficult case for both methods. FM frequently fails because of the lack of detected matches. Consequently, OF, that bases on the FM estimation, cannot perform well.

Dynamic Projections

As it was mentioned in 5.3.1 for dynamic projections I analyzed only $RMSE_{RGB}$ errors when compared the estimated compensations with the ground-truth data obtained from the static projections. Figure 5.12 presents the error curves plotted for 3 dynamic videos[6]. A short part of each video was selected that corresponded to a single-homography transformation. Further, Table 5.8 lists the average compensated values.

Analyzing these results, it can be concluded that, on the first two video sequences, the Feature Matching method outperforms the FM-OF method. On video1 the differences are

[6]Sample videos illustrating how the FM-OF method works can be downloaded at: https://perso.limsi.fr/setkov/demoVideos

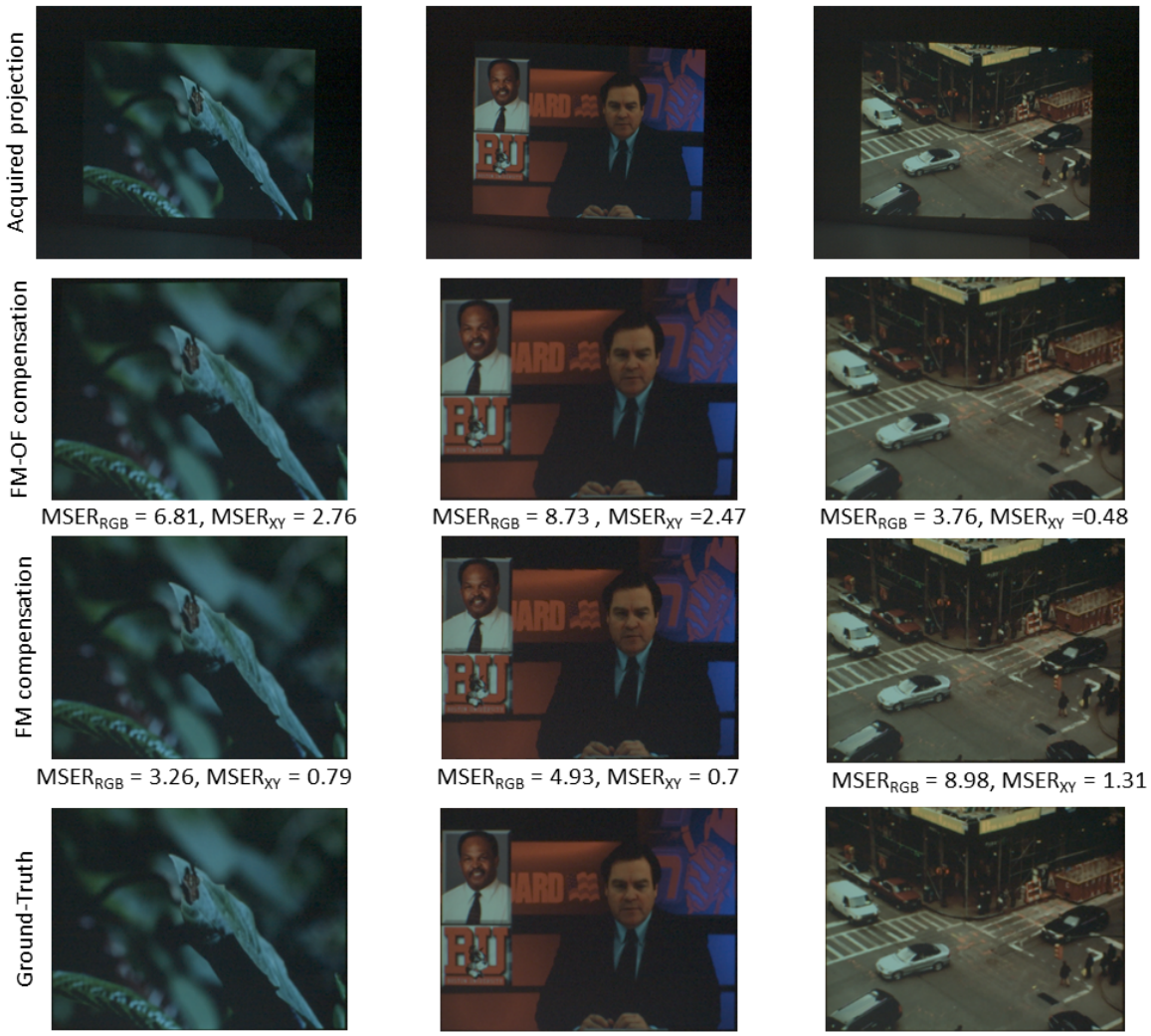


Fig. 5.11 Examples of the compensated images obtained through FM and FM-OF methods.

method	video1	video2	video3
	$RMSE_{RGB}$	$RMSE_{RGB}$	$RMSE_{RGB}$
FM-OF	14.00	55.13	25.71
FM	12.87	50.17	29.52

Table 5.8 Compensation performance of the tested approaches in the dynamic scenario.

not very high and can be solved by increasing the frequency of FM calls inside FM-OF method. In case of video 2 high errors produced by OF tracking can be explained by high error in the initial estimates by FM. Some improvement of FM procedure could alleviate this problem. Differently, on video 3 the FM-OF method shows quite good performance as compared to its competitor. While FM fails from time to time, OF tracking, provided good FM correspondences, yields good estimates which result in more a stable compensation.

Time complexity

The time complexity of both methods was measured by executing the evaluation framework for each method using the same test context. The maximum number of consecutive frames processed by OF without calling FM was equal to 30. The resolutions of the processed images were the following: 256×341 for the reference images and 350×450 for the acquired projections. All the algorithms were implemented using OpenCV library without any explicit parallel optimization. Execution time measurements were performed on a laptop equipped with Intel i5-4200U 1.60GHz processor and 8Gb of RAM. In the experiments FM-OF method required 0.146 sec, whereas FM method took around 0.705 sec per frame. These results demonstrate a significant speed-up in computations when performing OF + FM every 30 frames instead of applying FM at each frame.

5.4 Conclusions

This chapter has presented four evaluation frameworks. The first one evaluated performance of several color descriptors in a homography estimation task. Synthetically generated distortions allowed to observe how descriptor performance changes when one or another type of distortions is applied. The obtained results were analyzed both in terms of feature matching quality and homography compensation accuracy. For projections on ideal surfaces, I-HE is recommended. Concerning projections on colorful or textured surfaces, RGB-LHE provides the most stable results.

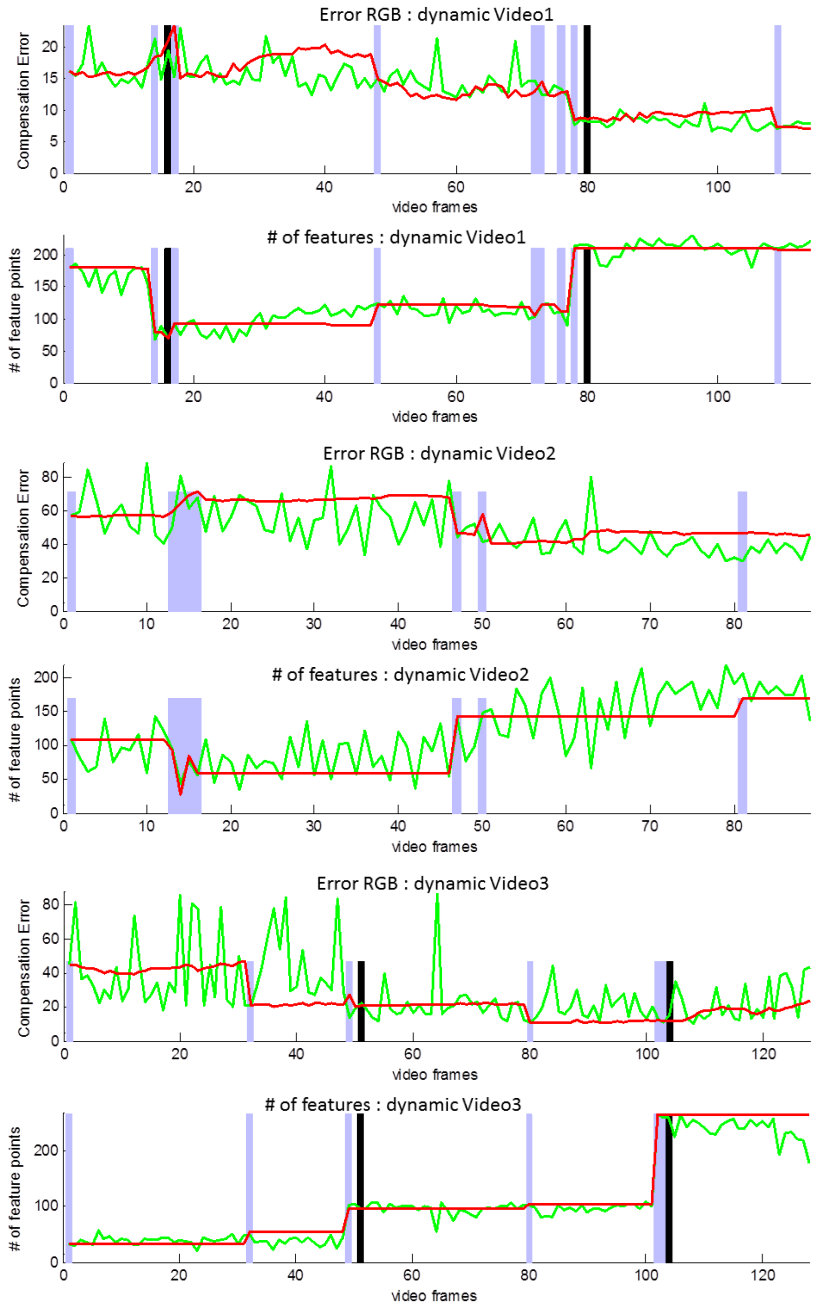


Fig. 5.12 Compensation results obtained through FM-OF method (red curves) and FM method (green curves). Blue timestamps indicate when FM was executed inside the FM-OF method. Black timestamps correspond to scene changes.

In the second framework real-world test images were used to evaluate several homography estimation methods in a projection-acquisition scenario. The results confirmed the assumption that arbitrary homography estimation methods can perform better if color invariance is applied. If the projection area can be approximately detected, then color invariance transformations can be applied at a preliminary step and thus no additional intervention in the compensation algorithms is required.

The third evaluation framework compared the performance of several local descriptors in object recognition. In this experiment, depending on whether test images will be corrected or not, I recommend the use of RGB-SIFT or LHE-RGB-SIFT for BoW-based object recognition.

Finally, an evaluation framework of the FM-OF compensation system from Section 3.3.4 was presented. The obtained results showed that, in some cases, by exploiting temporal coherence through OF it is possible to achieve results quite similar to the ones obtained through Feature Matching only. The results of this evaluation leave room for further improvements aimed at merging spatial and temporal information extracted from the images to obtain a time-efficient and high quality compensation. In terms of run time, the speed-up of the compensation obtained though FM-OF method is significant, by a factor of approximately 5, which makes the method interesting for real-time video projection compensation in SAR applications.

Chapter 6

GPU Implementation and Optimization

This chapter describes a real-time implementation of color-invariant homography compensation that was built from the algorithms that showed good performance in the previous evaluation described in Section 5.1.1. More specifically, the developed system includes LHE-RGB SURF color invariant feature matching and RANSAC algorithm for homography compensation that were implemented with GPU[1] acceleration to meet real-time requirements. In fact, the evaluation frameworks, implemented in C++ or Matlab and previously used in this PhD work, require much execution time and thus they are not suitable for real-time projection experiments.

RGB-LHE SURF was chosen as the color invariant method because of its better performance over the compared descriptors which was proved by the previous studies presented in Sections 5.2 and 5.1. To develop this compensation system, a CUDA implementation of the intensity SURF[106] was modified to perform RGB-LHE color transformation with the descriptor computation process.

The structure of this chapter is the following. First, the CUDA platform for parallel GPU computations is presented, CUDA was preferred to CPU optimization and other GPU optimization techniques. Then, I detail the implementation of the algorithms. More emphasis is paid to RGB-LHE SURF implementation which is the main contribution in terms of implementation efforts. Next, I go further in presenting the framework used to perform experiments and compare different implementations. Finally, the obtained results show that the GPU-based LHE-RGB-SURF implementation requires significantly less time for execution than its C++ implementation with insignificant losses in performance.

[1] Graphics Processing Unit

6.1 Programming platform

Over the last years heterogeneous programming model, that uses both CPU and GPU, has gained much success in scientific computations because it delivers an easy way to exploit multi-core structure of modern GPUs to make parallel implementations, while using CPU to handle critical and difficult-to-parallelize computation parts. In such programming models CPU is often called *host*, while GPU is called *device*. The *host* part, executed on a CPU, is also responsible for setting up the *device* (GPU) context, task scheduling and smooth memory transitions and results gathering from the *device* after the computations. According to the NVIDIA sources, the GPU performance reached 5500 GFLOPS/s in 2014, as against 750 GFLOPS/s achieved by Ivy Bridge CPU in 2013 [2]. Such a difference in performance as well as the need to make the implementation parallel made my choice in favour of CPU-GPU combination outsourcing most of the heavy computations on a graphics card.

There are two general purpose GPU (GPGPU) programming platforms that are commonly used for scientific computations. The first one, called OpenCL (Open Computing Language) represents a low-level API that is supported by most graphics cards. The second one, CUDA (Compute Unified Device Architecture), is a platform developed by NVIDIA and restricted to be used in their video cards. Since the workstation used for the experiments is equipped with an NVIDIA card, it was decided to use CUDA. Moreover, studies showed that CUDA slightly outperforms OpenCL because of the different memory models and NVIDIA's compiler optimizations for CUDA compared to those for OpenCL [34].

Another technique to make computations on GPUs is shader programming. Its application to general programming, however, presents a difficulty because the initial problem has to be reformulated by using basic data containers and multiple pass-rendering to off-screen framebuffers.

6.1.1 CUDA architecture

CUDA architecture consists of several layers. The lowest level corresponds to an NVIDIA graphics card with a driver that supports CUDA. Then goes the middle level represented by programming languages, typically C with some extensions, but also there are different wrappers available. At this level all algorithms implementation was performed. Finally, at the top layer there are different libraries and applications available, as for example programs for efficient matrix operations or fast Fourier transform computations.

[2] The comparison plot can be found on NVIDIA webpage at http://docs.nvidia.com/cuda/cuda-c-programming-guide/

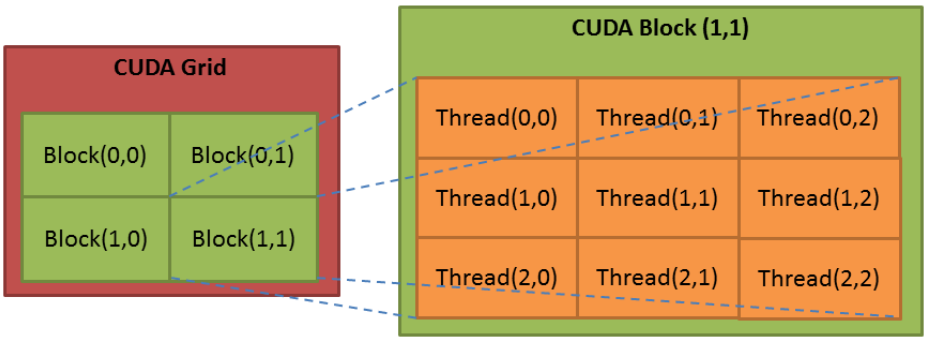


Fig. 6.1 Schematic representation of threads, blocks and grids in CUDA platform.

As it was mentioned above, the programming model comprises two concepts: *host* and *device*. The *host* program, run on the CPU, manages the computation process, including those parts dealing with GPU computations, such as GPU context set up, memory allocation, memory transfers and gathering the results. CUDA *device* functions, called *kernels*, are executed on the GPU and can be written in a C-syntax language with some extensions for parallel computations. The number of threads is specified in the *host* code when calling the kernel.

For programming convenience, CUDA platform provides a hierarchy of threads which allows grouping and executing them on different cores. Threads at the lowest level can be grouped in blocks that can operate on complex elements, for example vectors or matrices. At the highest level blocks are organized into grids. Both blocks and grids can be one-, two- or three-dimensional. Their size is specified before calling the kernel and limited by the hardware. Figure 6.1 provides an example of how threads are organized in blocks and grids.

CUDA platform has several types of memory accessible on GPUs. Each thread has its own *private* memory realized in registers. It is the fastest space that allows effective memory distribution over the threads of a block. Another type, *shared memory*, is a limited amount of memory reserved for each core (block of threads). In a similar manner, it provides a fast memory access and distribution over the thread blocks. Finally, *global memory*, accessed by all executed threads, remains persistent across kernel calls in the program and represents a significantly larger amount of GPU memory as compared to private and shared memories. Moreover, there are two types of read-only memory accessible by every thread, *constant* and *texture*. Figure 6.2 schematically shows this hierarchy.

The implementations and tests were performed on a workstation equipped with an NVIDIA Quadro 4000 video card. The NVIDIA driver version was 9.18.13.4062. Figure 6.3 presents the most relevant specifications of this card that were taken as a basis for implementation

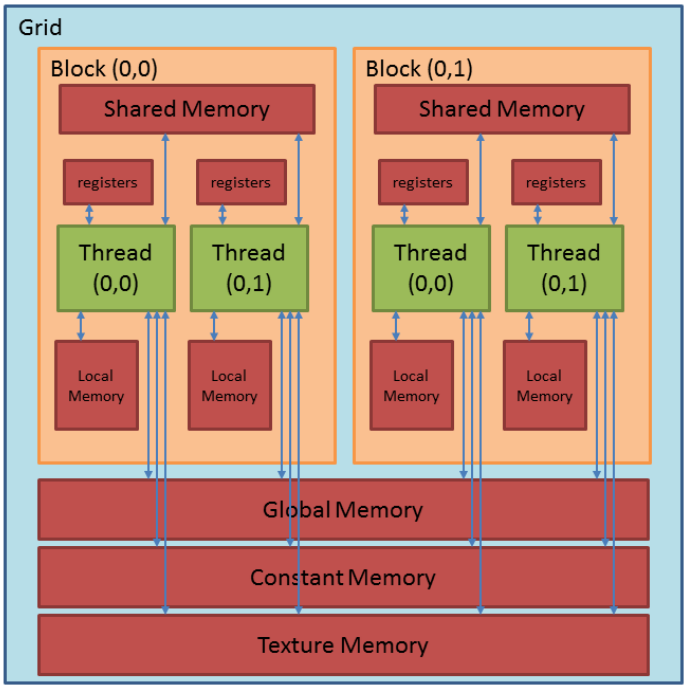


Fig. 6.2 Memory hierarchy in CUDA.

optimizations.

```
CUDA version:   v6050
CUDA Devices:

0: Quadro 4000: 2.0
  Global memory:   2048mb
  Shared memory:   48kb
  Constant memory: 64kb
  Block registers: 32768

  Warp size:         32
  Threads per block: 1024
  Max block dimensions: [ 1024, 1024, 64 ]
  Max grid dimensions:  [ 65535, 65535, 65535 ]
```

Fig. 6.3 CUDA specifications of NVIDIA Quadro 4000 graphics card obtained with driver version 9.18.13.4062.

6.2 CUDA implementation of FM-based geometric image compensation

The implemented homography compensation system represents a pipeline of several main blocks in which each part gets data from the previous step to make computations and then passes the results to the next block. The components are depicted in Figure 6.4.

Fig. 6.4 The CUDA RGB-LHE SURF implementation pipeline. The blocks in green denote the parts that were partially implemented in this work.

All the components, except for geometric warping, were implemented using GPU CUDA acceleration. Several CUDA implementations, available for research purposes, were taken and adjusted so that to be assembled together in one setup. Geometric warping was made by means of OpenCV library. Below goes the detailed description of each component.

SURF Feature Extraction. An implementation of A. Schulz[3] [106], declared to processes HD images in real-time, was taken as the basis for the further system development. In this implementation integral images are effectively computed on the GPU by means of CUDPP library[4]. The code is optimized to work with the standard number of octaves (2 or 3) and with 4 intervals, thus it was decided to use these parameters further in the system.

SURF Feature Description using LHE. The A. Schulz *et. al.* CUDA SURF implementation was used also for descriptor computation, although I have modified it significantly to support LHE-based descriptors and to work with color images. More detailed description is presented in the next section.

Feature Matching. A simple nearest neighborhood matching from CUDA SURF was extended to work with color images, that is, to compute euclidean distances between 192-dimension descriptors.

RANSAC filtering. A CUDA-RANSAC open source implementation by N. Ho[5] was embedded in the implementation. The author showed that his RANSAC is comparable in terms of accuracy with the corresponding OpenCV implementation.

Geometric warping. C++ OpenCV *warpPerspective* function served to warp the projection with the obtained homography transform.

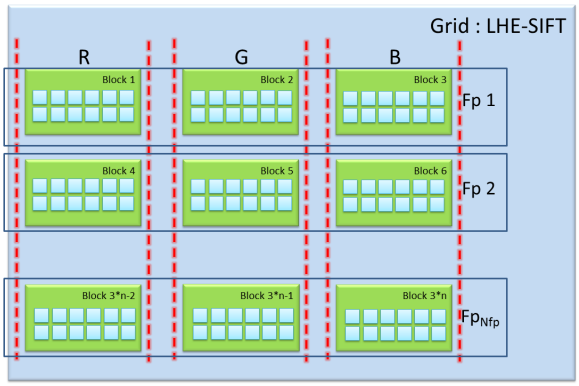


Fig. 6.5 Structure of CUDA processing units. Each CUDA block processes one of the n feature points (Fp_i) on one color channel (R, G or B).

6.2.1 CUDA Local Histogram Equalization

Embedding LHE in SURF descriptor computation process was not straightforward because SURF operates with integral images computed once for each frame. In turn, LHE-SURF requires that image region around each feature point is modified through the histogram equalization transform. Since feature point regions can overlap, integral image can no longer be used simultaneously by several parallel threads to speed-up the descriptor computation process, unless the algorithm is modified.

In order to adapt the LHE-SURF for CUDA platform with a single copy of the integral image processed in parallel, it was decided to sacrifice the effective integral image summation for the possibility of performing LHE for each feature point and for each color channel independently.

Figure 6.5 shows the task distribution over a batch of CUDA threads and blocks. Each block computes one LHE-SURF descriptor for one color channel. Thus, the total number of CUDA blocks involved in the computations equals 3 N_{fp} for an RGB image, where N_{fp} is the number of extracted feature points. Each block uses its own shared memory to store the feature point neighborhood in order to allow a faster access to this data when performing LHE and computing the Haar wavelet responses.

However, the size of the area around a feature point is variable depending on the scale, therefore, at high scales does not fit into the shared memory which is limited to 48Kb for the used video card. For the sake of performance, it was decided to skip points with a scale

[3] https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/object-recognition-and-scene-understanding/cuda-surf-a-real-time-implementation-for-surf/

[4] CUDA Data Parallel Primitives Library: http://cudpp.github.io/

[5] http://nghiaho.com/?p=490

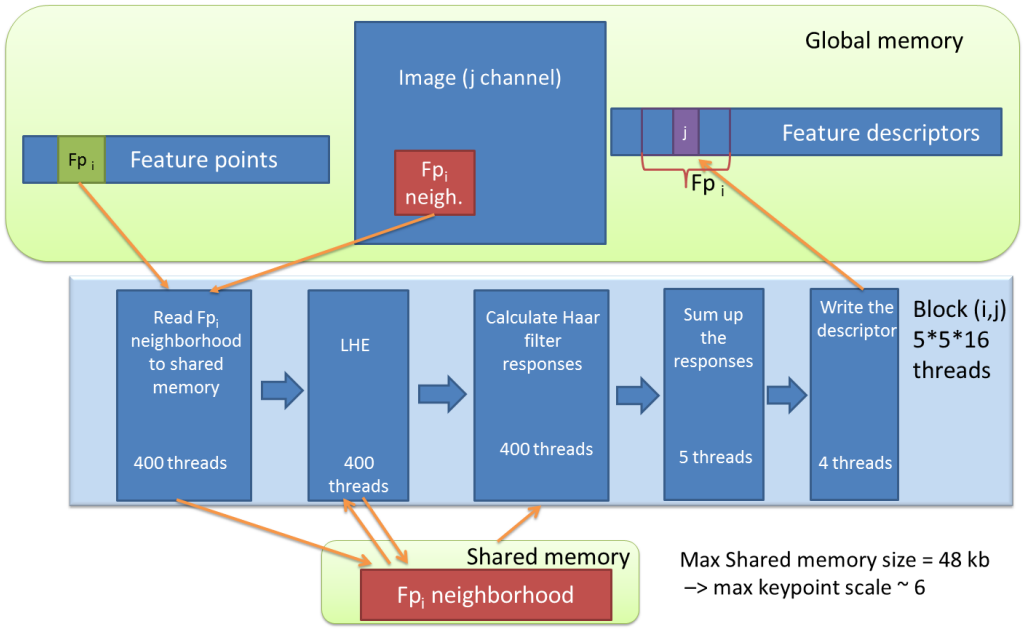


Fig. 6.6 An example of the command pipeline and memory access inside a CUDA block.

larger than 6, which means that only the two first octaves will be considered in contract to the standard number of octaves equaled to 3. As a result, on the tested videos, in average 5-10% of the feature points had to be withdrawn. As it will be further shown in Section 6.3.2, skipping points does not impact the results.

Figure 6.6 depicts a flowchart of the commands and memory accesses in a CUDA block. First, the block reads the feature point information that characterizes the image region to be used for descriptor computation. The data copy from the global to the shared memory is performed by all 400 threads.

Because each feature point region has its own orientation, the rotated coordinates of each sampling pixel of this region in the global image space should be computed and mapped to the shared memory space, as it is shown in the example from Figure 6.7. Next, Histogram Equalization is performed independently by each block. All threads in the block compute the cumulative histogram of the region using atomic summation operations to ensure that each thread can safely read and write the new value. At the next step after LHE computation, each thread computes Haar wavelet responses in the horizontal and vertical directions. Once all responses are computed, 5 threads sum them up and finally 4 threads write the obtained results in the global memory. When all descriptor parts are computed, it only remains to join the descriptors computed for the R, G and B channels and to normalize the resulting descriptor.

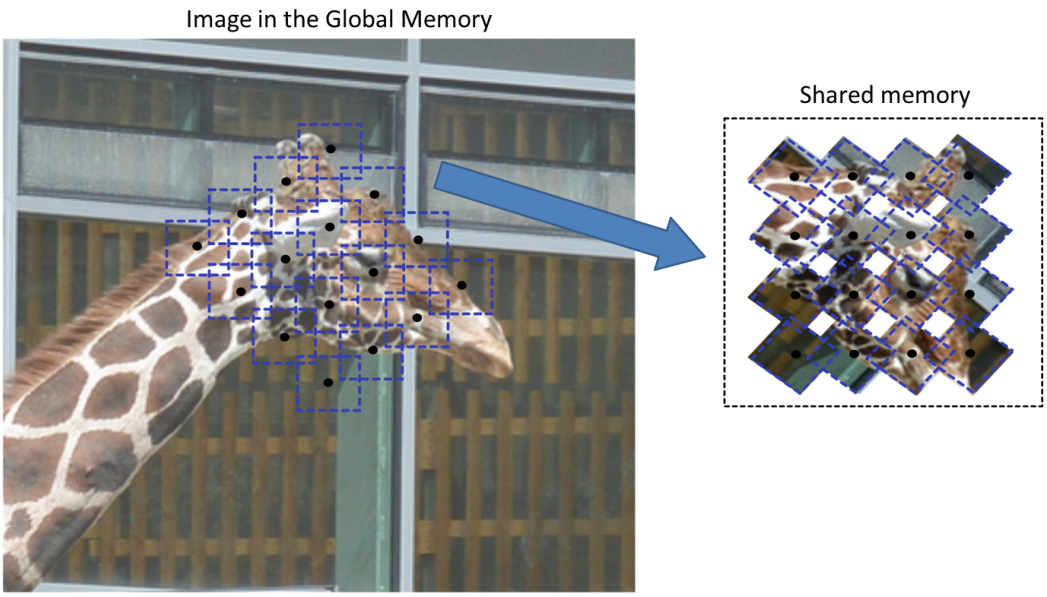


Fig. 6.7 Coordinate mapping between the global image and shared memory spaces. In this example the the neighborhood of the feature point consists of 16 subregions that are copied into the local memory after the rotation is compensated.

6.3 Quality and performance evaluation framework

This section describes the evaluation of the implemented compensation algorithms which allows to assess the quality of the CUDA GPU-based algorithm and make a comparison with C++ CPU-based implementations. First, the compared implementations are presented. For each implementation I detail how optimal parameters are calculated so as to have a fair comparison. Then, several evaluation criteria are discussed. The comparison was performed both in terms of quality and execution time. Finally, the results and conclusions are provided.

6.3.1 Compared algorithms

It was decided to compare the CUDA implementation with one CPU implementation that uses functions from the OpenCV library. Since the OpenCV descriptor computation functions do not support LHE transformation on the processed images, the comparison was performed using only Intensity and RGB images without LHE. Moreover, it was decided to test OpenCV SURF performance in two configurations, when 2 and 3 octaves are used, to show how the quality changes when the number of octaves is reduced. It has been explained in Section 6.2.1 why my CUDA implementation has to be restricted to compute two octaves.

Table 6.1 lists the compared methods.

Method	Color Transformation	# Octaves
CUDA-SURF	RGB-LHE	2
	RGB	2
	I	2
OpenCV-SURF	RGB	2, 3
	I	2, 3

Table 6.1 Summary of the evaluated methods.

6.3.2 Parameters optimization

Let us explain the procedure that has been performed to find optimal parameters for each implementation that were used in the evaluations. Since the CUDA and OpenCV implementations rely on parameters within different ranges, the optimal parameters configuration was searched separately for each method. A half of the projected reference images (61 images) from the ProCam database (for more detail refer to Section 4.2.2) was used as training data to obtain optimal parameters. The precise procedure is discussed below in this section. I start by defining the best parameters for feature point extraction. A Hessian threshold determines how many points will be extracted from the images. Then, I search for the best feature matching Nearest Neighborhood Ratio (NNR), which has an effect on the number of matches that are retained. Finally RANSAC, used to filter outliers, has two parameters to be tuned, the number of iterations and the threshold.

Feature Matching parameters

It was important to choose a good Hessian threshold, because small values may produce a large ratio of week feature points and thus many false positive matches will be detected, which will introduce noise into the system and decrease the probability of a good compensation through the RANSAC method. On the contrary, a small threshold means that only a few points will be retained, and thus the number of correct matches may be very low (less than 4 matches for homography estimation). Thus, a threshold should be defined so that to provide sufficient margin to detect a reasonable number of feature points for different images, both with small or high responses.

To find Hessian threshold parameters, I was inspired by the work of V. Nannen and G. Oliver in [88] that studied the effect of the number of detected features on the resulting

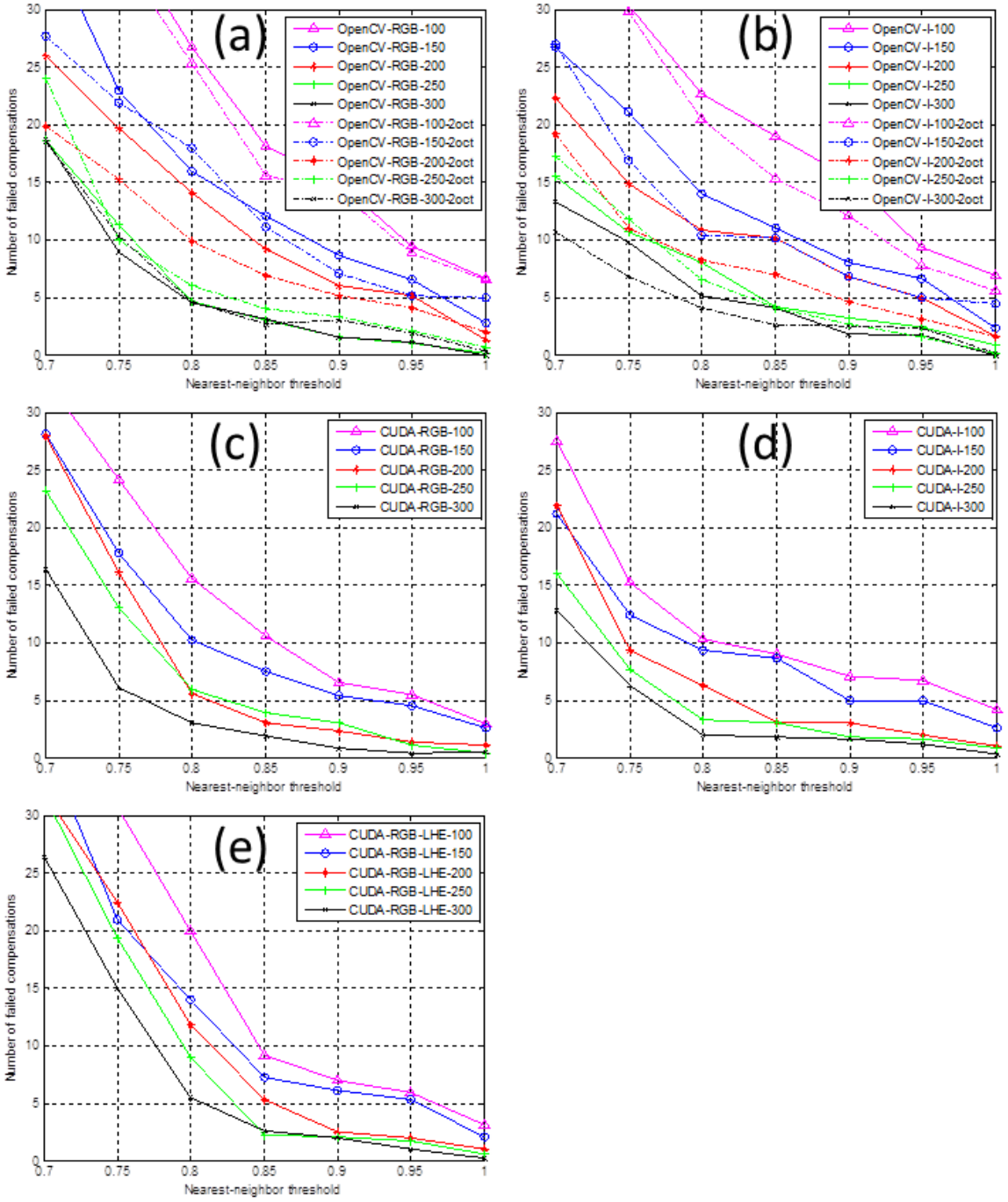


Fig. 6.8 Number of failed compensation computed by different SURF implementations with various Hessian thresholds. The following 5 methods are evaluated: intensity and color OpenCV SURF, intensity, color, and RGB-LHE CUDA SURF. Moreover, OpenCV methods were tested with 2 and 3 octaves in feature point detection.

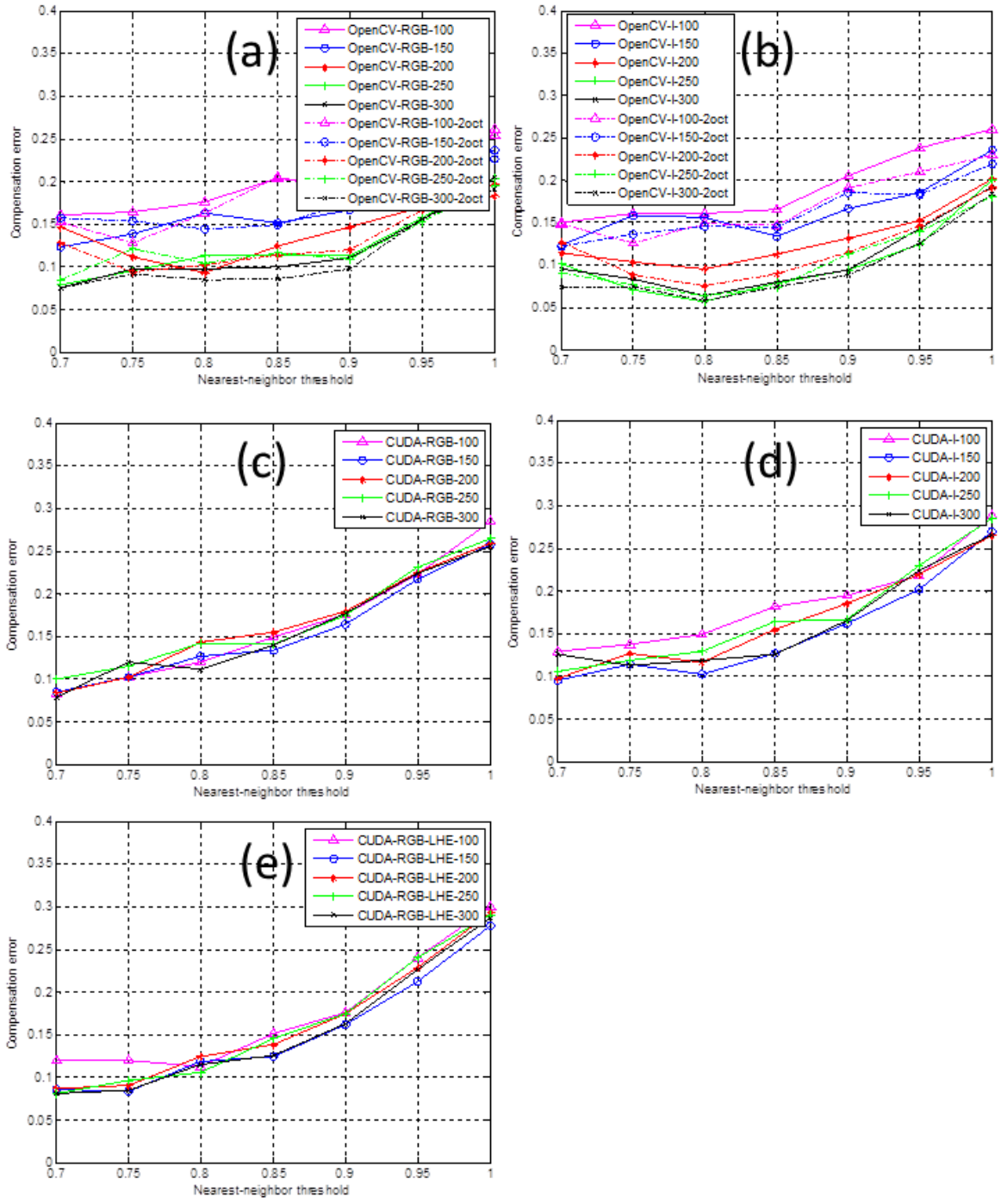


Fig. 6.9 Compensation errors computed by different SURF implementations with various Hessian thresholds. The following 5 methods are evaluated: intensity and color OpenCV SURF, intensity, color, and RGB-LHE CUDA SURF. Moreover, OpenCV methods were tested with 2 and 3 octaves in feature point detection.

Method	Color Transformation	NNR Threshold	RANSAC threshold	RANSAC # iterations
CUDA-SURF	RGB-LHE	0.85	8.5	2100
	RGB	0.85	5.0	3700
	I	0.8	4.5	4100
OpenCV-SURF (2 octaves)	RGB	0.85	6.5	4500
	I	0.8	4.0	1900

Table 6.2 Feature Matching and RANSAC parameters selected for each method based on training images.

visual odometer performance. It appeared that the results remain stable when the number of feature points lies within a certain interval (from 50 to 200 in their application). Similarly I studied the quality of feature matching with different numbers of features that I varied in the range from 100 to 300 points per image. To do so, for each range of points $[50, N_i]$, where $N_i \in [100, 150, 200, 250, 300]$, I found the best Hessian thresholds that gave the numbers of detected feature points close to the desired interval. Then, I assessed and compared the quality across all parameters configurations, independently for each method. Unlike most works that use Precision-Recall or Receiver Operator curves to evaluate feature matching quality, I compared the performance directly on the compensation results.

In fact, *Precision* and *Recall* metrics, even though providing some evidence on the behavior of one descriptor or another, cannot clearly reflect the compensation quality (refer to the study presented in chapter 5.1), because RANSAC is executed after feature matching. This method estimates the homography with a certain probability and is capable of filtering out false correspondences. To that end, I suggest considering the compensation error while keeping track of the number of compensation failures (when less than 4 matches are computed).

The compensation error was computed as the average SSD distance between the RGB pixel values in the estimated compensated images and their ground-truth compensations. A compensation is considered as failed if less than 4 matches are computed as a result of feature matching. In this experiment I fixed RANSAC parameters for all the methods, i.e. the number of iterations were 200 and the threshold value equaled 2. The plots were made by varying NNR threshold k from 0.7 to 1 with a step of 0.05. Because RANSAC is not a deterministic algorithm, for each parameters configuration I repeated the experiments 10 times and averaged the results. Figures 6.8 and 6.9 show the compensation results of CUDA and OpenCV implementations for different Hessian thresholds. Several major observations can be made from the obtained results. First, when there are not enough features extracted

from the images, the number of computed correspondences is not sufficient for homography estimation. Low Hessian thresholds, corresponding to 250 and 300 extracted feature points, allow having more than 4 matches almost for every image (less than or around 5 failures for OpenCV (plot in Figure 6.8.(a)-(b)) and CUDA implementations (plots in Figure 6.8.(c)-(e)) for NNR threshold value 0.85). CUDA implementation has a small number of compensation failures even for the threshold corresponding to 200 features. Second, if we consider the compensation errors, OpenCV implementation performs well (6.9.(a)-(b)) when a large number of features are extracted. As this number becomes smaller, the performance decreases. The results are different for CUDA implementation (6.9.(c)-(e)). Except for the highest threshold (that corresponds to 100 features), all the error curves are similar, which means that the implementation is quite robust against the variations of feature point quantities.

Several other observations can be made from the results. Using only 2 octaves to detect feature points improves the results (see Figures 6.8.(a)-(b) and 6.9.(a)-(b)). It means that in the training data there are no big differences in scale that cannot be coved by 2 octaves. Considering points at a larger scale introduces noise into the feature matching process. In the further evaluations OpenCV implementation with only 2 octaves was used.

In general, the compensation results on the training data are very similar for intensity and color descriptors. OpenCV implementation produces lower compensation errors which, however, can be explained by slightly worse results in terms of the number of compensated images.

Analyzing the plotted curves, each method was assigned a configuration of feature matching parameters (Hessian and NNR thresholds) that gave satisfactory results on the training images. The lowest Hessian thresholds corresponding to 300 features were chosen for each method. Table 6.2 shows the values of the selected NNR thresholds in the third column.

RANSAC parameters

The next step consists in finding optimal RANSAC parameters: the number of iterations and the threshold. The increase of the number of iterations up to a certain value can improve the compensation quality, although the execution time will be increased as well. RANSAC threshold defines the admissible distance between the planar surface estimated from a set of inliers and the rest of the points.

To find a good RANSAC threshold value, I executed all the methods several times, varying the threshold parameter in the range from 1.0 to 10.0 with a step of 0.5. To make

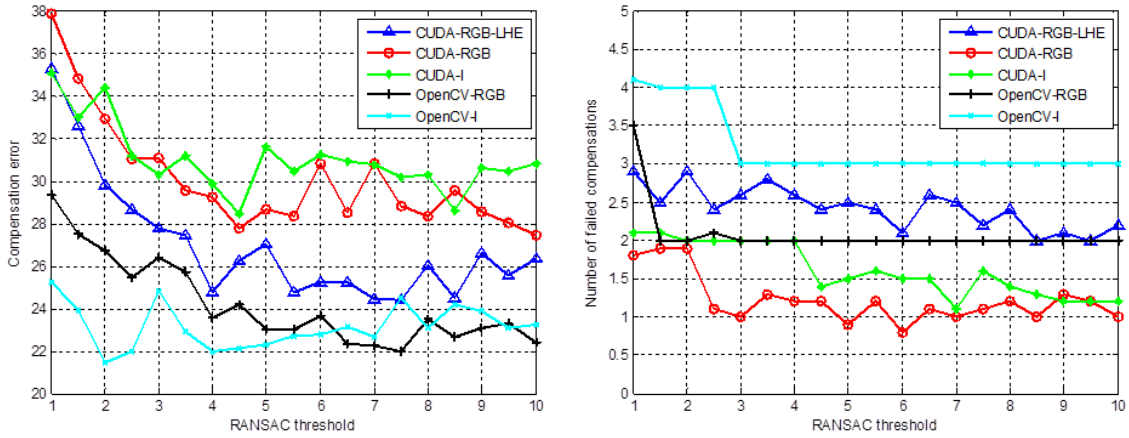


Fig. 6.10 Compensation errors and the number of failures obtained by the 5 evaluated methods with various RANSAC thresholds.

the results more precise, 1000 iterations in RANSAC were used. Each configuration was executed 10 times to get an average result value. Figure 6.10 shows compensation results of the methods with a different RANSAC threshold. It can be seen that increasing the threshold until a certain value has a positive effect on the compensation performance. The retained RANSAC threshold values are listed in table 6.2.

Similarly, the optimal number of iterations in RANSAC was found. The values from 100 to 4900 with a step of 200 were examined. Figure 6.11 shows compensation results from each number of iterations. For most of the methods increasing the number of RANSAC iterations up to 1000 iterations leads to a significant reduction of the compensation errors. When passed this value, the improvement is not significant. Figure 6.12 shows how the execution time changes when the number of iterations is increased. The increase in the run time is not so prominent until 3800 iterations. Then the time complexity starts to grow faster. It takes around 20 ms to complete 5000 iterations which seemed acceptable in this work. The values that produce the best compensation results were selected and presented in table 6.2.

6.4 Evaluation

In the experiments the resolution of the reference images was 367×283, and the size of the acquired projections was 630×490. All the methods were executed on RGB images with cropped projection area in order to reduce the execution time and to facilitate the feature matching. In terms of color processing, the images did not undergo any color correction.

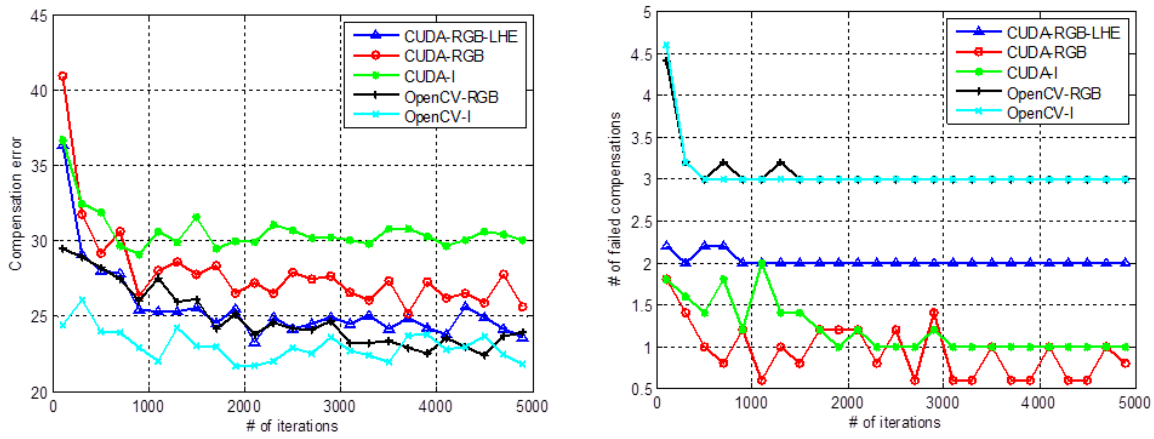


Fig. 6.11 Compensation errors and the number of failures obtained by the 5 evaluated methods with various numbers of iterations used by RANSAC.

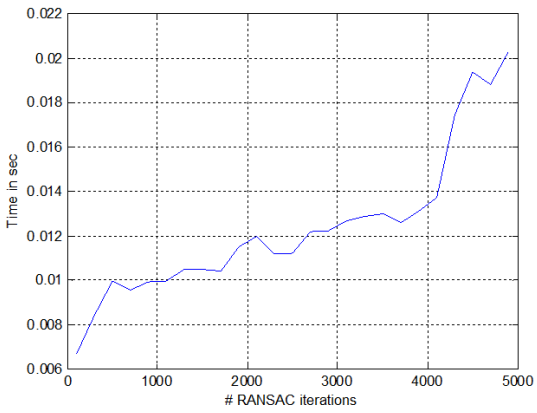


Fig. 6.12 RANSAC execution time dependence of the number of iterations.

Hence, the unprocessed acquired projections with their photometric distortions were used.

6.4.1 Evaluation criteria

Now, when all the parameters have been selected for each implementation, several evaluations are discussed. The algorithms were compared both in terms of time complexity and compensation performance. In the first evaluation the CPU OpenCV implementations of I-SURF and RGB-SURF were respectively compared with the corresponding GPU CUDA implementations. Total time spent on computing compensations, as well as more precise profiling are presented below. In the second evaluation, the algorithms were run on tested videos. Similarly to the previous evaluation described in section 5.2.1, the mean

SSD distance between computed and ground-truth compensations was exploited to measure the quality. On the captures through a dynamic projector-camera system, an approximate ground-truth was computed (refer to 4.3.5 to see more details on ground-truth computation for the test video acquisitions).

6.4.2 Evaluation results

This section presents the results obtained in three experiments. First, the compensation quality was evaluated on a set of video sequences projected by the static ProCam system. Second, videoprojections acquired by the dynamic ProCam system were used to compare the compensation quality of the implementations. Finally, the time complexity of the CUDA and OpenCV implementations were measured and compared.

Compensation errors obtained by the static ProCam system

Figure 6.13 illustrates the plotted error curves obtained on the training set of images. In this graph the abscissa corresponds to the number of images that are compensated, and the ordinate represents the Cumulative Compensation Error (CCE) that can be interpreted as the sum of the compensation errors for the current and all preceding images. This metrics was chosen to show the performance of a method on the whole set of images and to compare it with another method. The obtained results are quite similar on most of the image sequence. The difference is not significant with respect to the number of test images, even though for the last images CUDA-RGB-LHE error curve is slightly below the compared methods. It means that when all parameters are tuned individually for each implementation, CUDA SURF-RGB-LHE yields the best performance among all compared methods.

Next, several evaluations were made on prepared test videos (refer to chapter 4.3.1 for more details about video acquisition). For the experiments, the “Street” and “News” video sequences were selected. In the first scenario, videos were projected by the static projector-camera system. The second scenario implied that one static frame from each video sequence was projected while the system was in motion.

The resolution of the reference images was 367×283, and the size of the acquired projections was 630×490. The projection area was cropped in the acquired images in order to reduce the execution time and to facilitate the feature matching process. In terms of color processing, the images did not undergo any color correction. Hence, the unprocessed acquired projections with their photometric distortions were used.

Figure 6.14 shows the compensation results for two videos and two acquisition scenar-

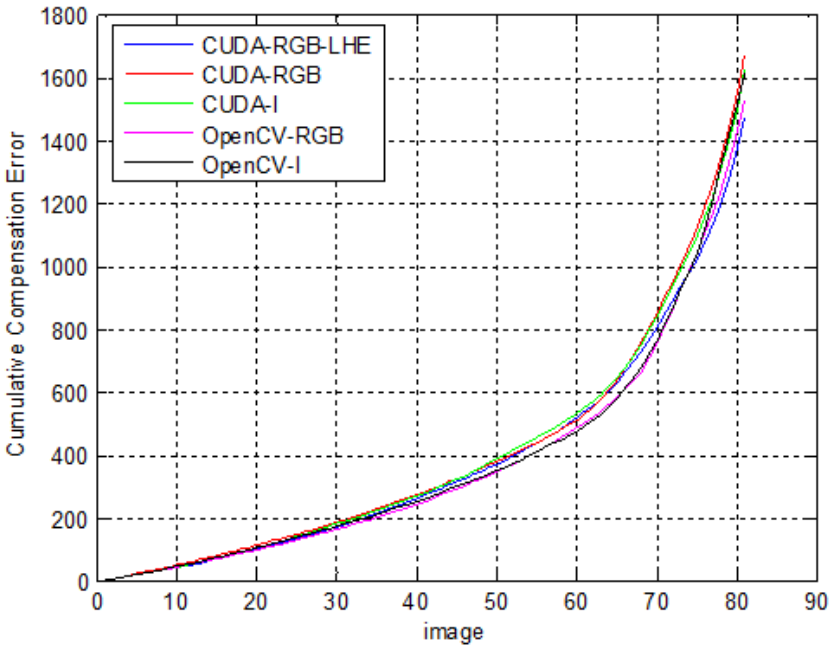


Fig. 6.13 Compensation results obtained on the train set.

ios. First of all, it can be noticed that, for the"News" video sequence, the best results are achieved using RGB or RGB-LHE descriptors, while, for the "Street" video, Intensity-based methods perform better. Different colorfulness of the videos can explain this fact. Figure 6.15.(a) and (b) show the chroma histograms of the reference and projected frames from both videos (Figure 6.15.(c) and (d)). It can be seen that the "News" video sequence is more colorful and thus using RGB color space in feature matching can yield better results when compared with the Intensity-based descriptors. The compensations of the projected "News" video are significantly worse than the ones that were obtained on the "Street" video. It can be explained by the low contrast of the acquired projections which complicated the feature extraction process. As a result, only a small amount of feature points could be extracted. In this case, using color descriptors increases the feature matching recall.

Compensation errors obtained by the dynamic ProCam system

Passing to the experimental scenario in which static frames were projected while the ProCam system was in motion, it can be observed that the nature of the results is similar. For the projected frame from the "News" video, color descriptors are preferred, whereas I-SURF is slightly better than its RGB analog.

The results when only one frame from the "News" video was projected in the dynamic

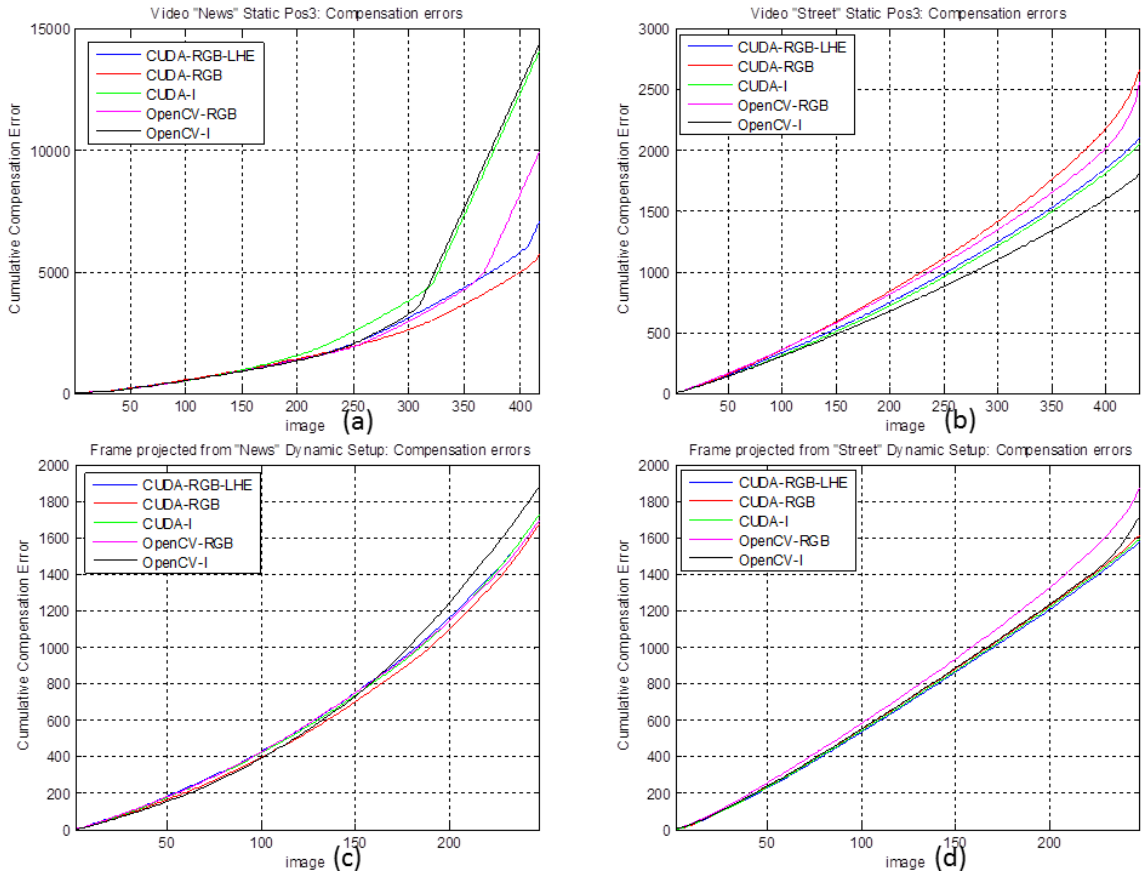


Fig. 6.14 Compensation results obtained for two videos in two projection scenarios. (a) and (b) - "News" and "Street" videos projected by the static setup; (c) and (d) One frame from "News" and "Street" videos were projected by the dynamic setup

setup are far better compared to the scenario in which the whole video was projected. The chosen projected frames have a higher contrast than the average contrast of the whole video and, therefore, more features can be extracted from the projected selected frames.

CUDA SURF LHE-RGB is not the best method in most of the cases, however, it is the one that gets the best average score. It has the best average rank of 2.25. It is slightly better than CUDA SURF-RGB (Avg. rank 2.5) and CUDA-I (2.75). OpenCV implementation shows worse performance in average (3.5 for the Intensity SURF and 4 for the OpenCV RGB-SURF). The differences in the results can also be explained by non-optimal parameter choice on the training images because they differ from the test data in terms of color, intensity, and distortions.

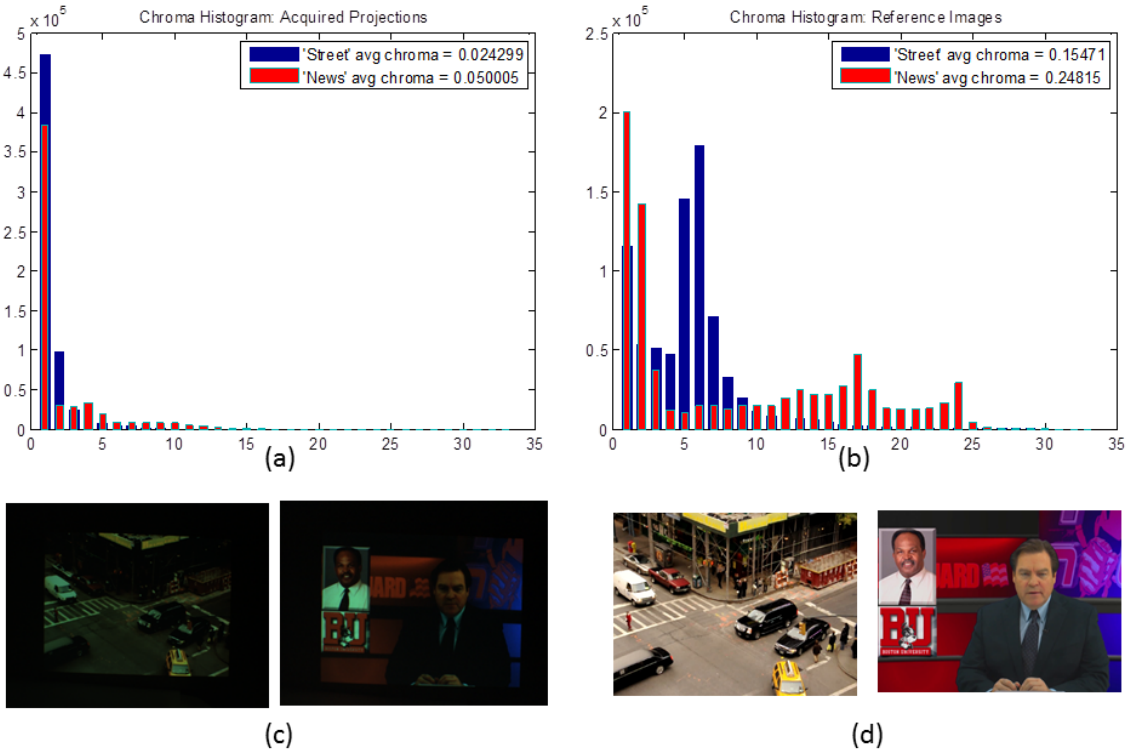


Fig. 6.15 (a) and (b) depict chroma histograms of two acquired (c) and two reference (d) images.

Execution time

Figure 6.16 illustrates the execution time curves for each of the tested methods. As it was expected, time complexity grows linearly with the number of extracted features. The time complexity of CUDA-RGB-LHE is more sensitive to the increase in feature points than CUDA-RGB, because local histogram equalization is performed three times for each feature point. This procedure is the most time-consuming in the pipeline (see time profiling of CUDA SURF RGB-LHE in Figure 6.17), thus it can be a subject for further optimizations. Even though CUDA implementation of SURF RGB-LHE is now capable of compensating images with 300 features each, at the speed of almost 10fps, it outperforms OpenCV CUDA-RGB implementation that uses only CPU in terms of time complexity. Regarding the CUDA profiling, a considerable execution time is spent on overhead expenses such as memory allocation, transfer between CPU and GPU, device synchronization. This time is very similar both for CUDA-RGB and CUDA-RGB-LHE, which means that LHE implementation does not harm the time complexity of the whole algorithm.

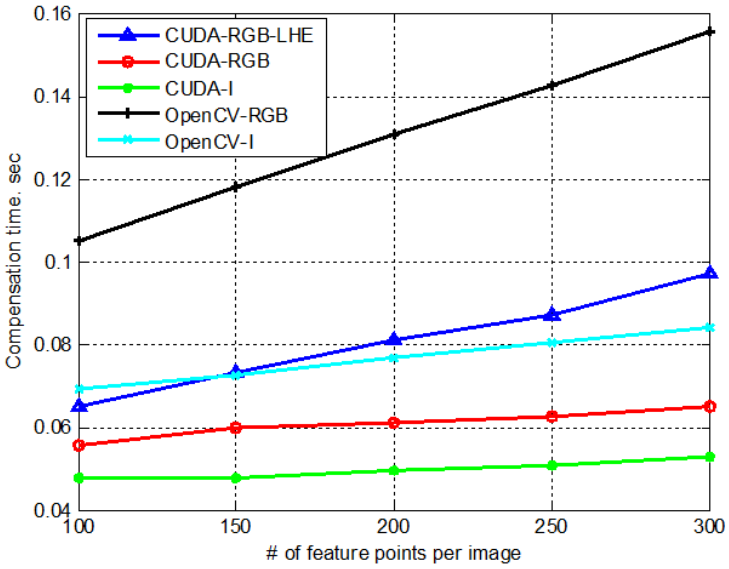


Fig. 6.16 Computation time required to process a pair of images and to obtain a compensation.

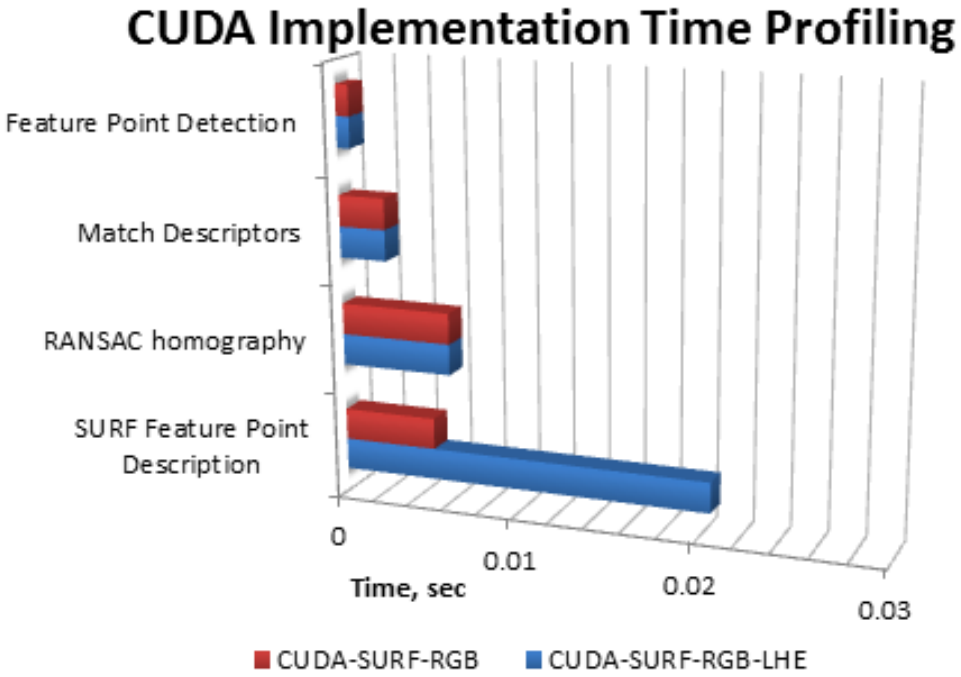


Fig. 6.17 CUDA implementation time profiling.

6.5 Conclusions

In this chapter two main contributions were presented. First, the implementation of LHE for CUDA SURF was described. Second, this implementation was compared with color and intensity SURF descriptors implemented in both CUDA and OpenCV in terms of execution time and compensation quality. The evaluation showed that CUDA LHE-RGB SURF is capable of compensating 10 images per second which is faster than OpenCV RGB SURF on CPU. The compensation quality differed for each method and for each experimental scenario. Nevertheless, LHE-based descriptor appeared to be the method with the best average compensation quality, followed by CUDA RGB-SURF. OpenCV implementations showed slightly worse performance, which can be explained by less optimal parameter selection and implementation particularities.

Chapter 7

Conclusion

7.1 Conclusions and contributions

The objective of my PhD thesis was to investigate the problem of geometric correction in the presence of complex photometric distortions typical for a projector-camera system. In the context of this work three questions, formulated in Section 1.2, were addressed. The main contributions and conclusions are described below.

Color invariance feature matching. Color invariance effect on feature matching between reference and acquired projected images was studied through the example of Local Histogram Transform and some other color invariant spaces. The properties of the local descriptors were studied and several evaluations were made. Moreover, a new LHE-based color invariant descriptor was proposed which was shown to be invariant to the color changes produced by a ProCam system. In the evaluation from section 5.1, LHE-RGB SURF was the most stable descriptor with the best average compensation quality. It failed in less scenarios than its competitors. In case color distortions are uniform in the image, the use of intensity global histogram equalization is more advantageous when compared with the other descriptors.

ProCam database. A part of my work was dedicated to preparation of a large database of acquired projections, called ProCam database (Sections 4.3 and 4.2). It covered various scenarios when a ProCam system was used, including image and video projections from static and dynamic positions. Reference projected content consisted of images representing different categories (natural images, faces, corporate presentations) and videoprojections. Different geometric distortions ranged from single- to quadruple-homography transforma-

tions and were either static or changed dynamically in each capture. Photometric distortions were represented by four illumination sources, colored backgrounds and eventually by the uncalibrated projection and acquisition devices. The database was provided with additional data that allow computing ground-truth geometric transformations and performing different color corrections. The importance of the presented database was shown by three evaluations (Sections 5.2, 5.3, and 6.4) and by one application to object recognition in the ProCam system context (Section 5.2.2).

Evaluations and applications on Static ProCam acquisitions. One of the evaluations was performed on static projected images from the database (5.2.1). This experiment aimed to study to which degree the quality of geometric compensation for visualization quality enhancement can be improved when color invariance is enabled. The results revealed that, in general, color invariance improves the homography compensation quality. Particularly, it was demonstrated by the example of homography estimation methods for image stitching improved by HE color invariance.

Another application of the database was proposed in this work which consisted in performing object recognition through the Bag-Of-Words method from a set of projected images (Section 5.2.2). This application could be interesting for both research communities and industry that deal with home or corporate projections and that are interested in object recognition. Indeed, such an application could recognize actor/movies in the case of home projector cinema, and different objects (buildings, cities, or products) from presentation images. The experimental results showed that this problem is far more complicated than the classical object recognition problems because the BoW classification model was trained on original (projected) images whereas recognition was performed from acquired projections that had strong photometric distortions due to the uncalibrated camera and projector responses, and illumination in the scene. Hence, solving this problem requires some preprocessing to improve the recognition quality. Namely, two steps were carried out. The first one, a simple color correction, minimized the color distortions, whereas the second one, projection area selection, reduced the effect of the background and, thus, the number of noisy feature points involved in object recognition.

Evaluations and applications on Dynamic ProCam acquisitions. To address the problem of geometric compensation for video projections, in Section 3.3.4 I proposed to use a combination of Feature Matching and Optical Flow algorithms. It allows exploiting temporal coherence of the projected video content and applying it to optimize geometric transfor-

mations. The results obtained in Section 5.3.2 showed that if Feature Matching is performed from time to time (for example, every 30 frames) and the rest of the time low-weight Optical Flow is applied, the compensation results do not degrade much, while yielding a significant gain in the time complexity (by around 5 times in the experimental framework).

GPU parallel implementation of the color invariant Feature Matching compensation In order to demonstrate that color invariant feature matching can potentially be applied for real-time geometric compensation in a ProCam system, in this work a compensation framework, described in Chapter 6, was implemented with the use of GPU parallel processing tools. I began with an open-source CUDA SURF implementation that was then modified by embedding LHE computation and multiple-channel descriptor computation and matching. The implementation was completed with an CUDA RANSAC implementation that had been also publicly available.

In the evaluation I compared the performance of the CUDA and OpenCV SURF implementations. Obtained on a series of experiments described in Section 6.4, the results allow making several conclusions. First, it was shown that CUDA RGB-LHE implementation had in average the best compensation quality compared with Intensity- and RGB-SURF implemented in CUDA and OpenCV. It proved the first evaluation performed on a synthetic dataset in which the LHE-RGB descriptor had in average the best performance.

Second, CUDA implementations of several SURF-based feature matching techniques generally give better image compensation quality than the corresponding OpenCV implementations. Finally, an analysis of the time complexity of the methods was presented. Although LHE-based version is slower than its Intensity- and RGB-CUDA analogs, it is still faster than OpenCV RGB-SURF compensation with an average observed speed of 10 fps. For comparison, C++ OpenCV RGB-SURF implementation could only process around 6 frames per second.

7.2 Future work perspectives

My PhD thesis research work opens several research avenues in the field of geometric compensation for SAR. This section summarizes several perspectives of unintrusive geometric compensation based on the outcomes of my research work:

Binary descriptors. More investigation can be performed to examine the performance of binary descriptors, such as BRISK and FREAK, in the context of homography compensation

in a projector-camera system. The main advantage of these descriptors is a significantly reduced time complexity in comparison with more conventional SIFT and SURF, which could be an important factor in such real-time applications as geometric compensation for ProCam systems. Moreover, binary descriptors should be suited for ProCam systems because they are independent on the absolute values of luminance but only depend on the color ranking, as for LHE descriptors.

Adaptive method for most suitable color descriptor selection. Since in my evaluation several descriptors showed good performance in different color distortion scenarios, the choice of the most suitable descriptor can be made adaptively depending on the type of distortions. In the simple case of homogeneous color changes, I-HE can be applied. When complex inhomogeneous color distortions occur, the best would be to use RGB and LHE-RGB based descriptors.

A generic Multiple-homography compensation. Even though the case of multiple-homography transformations was addressed in the prepared ProCam database, more work can be done generalize the compensation approach, described in Section 3.3.3, to simultaneously compensate 3 and more homography projections. This work presented only double-homography transformation compensation in which the heuristic algorithm performed well. This algorithm should be validated on more complex projection transformations.

Another improvement, that needs to be performed, is automatic estimation of the number of projection surfaces. It is useful in the case of a dynamic ProCam when projection is performed dynamically on an arbitrary number of planar surfaces (for example, one or two surfaces in the video projections described in Section 4.3.1). This problem could be approached by analyzing the sets of inliers obtained by RANSAC.

Spatial image information extraction for efficient feature matching. The developed geometric compensation algorithms were based only on extracted local features. Such an approach provide good results if the matched images contain high frequency information. If the images contain large uniform regions, then feature matching would probably fail because there are not enough feature points extracted from those regions. To cope with this problem image segmentation could help. Matching in this case could be performed at the level of the whole uniform image regions, rather then local features. However, large color variations in the matched images complicate segment matching. It is difficult to achieve similar segmentations so as to have similar segments in the both images.

An alternative approach to improve the matching quality can be represented by Graph Matching algorithms [23, 65, 136]. Exploiting spatial coherence between the detected features would improve the robustness to false matches that do not fit in the graph structure. However, time complexity of the graph matching techniques could become an issue when targeting real-time performance.

Improved Feature Matching-Optical Flow (FM-OF) compensation method. Several ways can be taken to improve the compensation quality of the FM-OF method. First, the thresholds can be adaptively optimized along a video stream to make the system more generic. Second, the quality of OF can be ameliorated by better exploiting previously estimated transformations, for example a feature point motion history can be used to make the tracking more robust. Finally, statistical filtering, such as Kalman, can employ a smoothness constraint on the resulting homography to improve its detection on a sequence of spatially coherent images.

Geometric compensation-based color correction. Because the results of geometric compensation provide a per-pixel correspondences between the reference and the acquired projected images, this information can be leveraged to estimate color transformation on the fly. By gathering the corresponding pixel color values during several frame, it is possible to get an approximated model that maps original projected color values to the acquired ones. The advantage of such an approach lies in the fact that both color and geometric corrections are working together in an unintrusive ProCam system. However, several challenges can be mentioned. First, in such a compensation system, the color compensation quality would strongly depend on the precision of the estimated geometric transformation. Second, the problem of choosing the color mapping function would occur if there is no information *a priori* available on the device responses and the illumination in the scene. Next, the problem of inhomogeneous color variations, for instance when the projection surface is nonuniformly colored or when ambient illumination lights only a part of the projection, should be additionally addressed. Finally, the compensation results as well as the number of frames, needed to adapt the color compensation model, would depend on the color distribution in the projected content. It is not guaranteed that the sufficient number of color will be projected to cover the whole color gamut.

Human evaluation of developed compensation methods. All the evaluations, presented in my PhD work, aimed at quantitative quality assessment of different algorithms. Since er-

ror measures sometimes differ from the human perception, it could be interesting to perform a human evaluation of the compensation quality of the developed methods.

Passive-active hybrid compensation approach. To make a robust geometric compensation system suitable for SAR applications, Structured Light techniques could be added to unintrusive content-based compensation. The former method is executed only when the later fails, which could be the case for homogenous and textureless images or very complex color, or geometric distortions that are difficult to deal with local feature matching.

References

[1] Abdel-Hakim, A. E. and Farag, A. A. (2006). CSIFT: A SIFT Descriptor with Color Invariant Characteristics. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 1978–1983, Washington, DC, USA. IEEE Computer Society.

[2] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR)*, pages 510–517. IEEE Computer Society.

[3] Albitar, C., Graebling, P., and Doignon, C. (2007). Design of a monochromatic pattern for a robust structured light coding. In *Proceedings of the International Conference on Image Processing, EEE ICIP 2007, September 16-19, 2007, San Antonio, Texas, USA*, pages 529–532.

[4] Audet, S. (2012). *Markerless interactive augmented reality on moving planar surfaces with video projection and a color camera*. PhD thesis, Tokyo Institute of Technology.

[5] Battle, J., Mouaddib, E., and Salvi, J. (1998). Recent progress in coded structured light as a technique to solve the correspondence problem : A survey. *Pattern Recognition*, 31(7).

[6] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. J. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

[7] Benko, H., Jota, R., and Wilson, A. (2012). Miragetable: Freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 199–208, New York, NY, USA. ACM.

[8] Benko, H., Wilson, A. D., and Zannier, F. (2014). Dyadic projected spatial augmented reality. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 645–655, New York, NY, USA. ACM.

[9] Bentolila, J. and Francos, J. M. (2014). Homography and fundamental matrix estimation from region matches using an affine error metric. *Journal of Mathematical Imaging and Vision*, 49(2):481–491.

[10] Bérard, F. (2003). The magic table: Computer-vision based augmentation of a whiteboard for creative meetings. In *IEEE Workshop on Projector-Camera Systems (PROCAM)*.

[11] Bimber, O., Coriand, F., Kleppe, A., Bruns, E., Zollmann, S., and Langlotz, T. (2005). Superimposing pictorial artwork with projected imagery. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA. ACM.

[12] Bosch, A., Zisserman, A., and Muñoz, X. (2007). Image classification using random forests and ferns. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pages 1–8.

[13] Bosch, A., Zisserman, A., and Muñoz, X. (2008). Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):712–727.

[14] Brown, M. and Lowe, D. G. (2003). Recognising panoramas. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1218–1227.

[15] Buchsbaum, G. (1980). A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1 – 26.

[16] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg. Springer-Verlag.

[17] Canclini, A., Cesana, M., Redondi, A., Tagliasacchi, M., Ascenso, J., and Cilla, R. (2013). Evaluation of low-complexity visual feature detectors and descriptors. In *18th International Conference on Digital Signal Processing, DSP 2013, Fira, Santorini, Greece, July 1-3, 2013*, pages 1–7.

[18] Carr, P., Sheikh, Y., and Matthews, I. (2012). Point-less calibration: Camera parameters from gradient-based alignment to edge images. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 377–384.

[19] Carrihill, B. and Hummel, R. A. (1985). Experiments with the intensity ratio depth sensor. *Computer Vision, Graphics, and Image Processing*, 32(3):337–358.

[20] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[21] Chang, C.-H., Sato, Y., and Chuang, Y.-Y. (2014). Shape-preserving half-projective warps for image stitching. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, pages 3254–3261.

[22] Cheok, D. A., Haller, M., Fernando, O. N. N., and Wijesena, J. P. (2009). Mixed reality entertainment and art. *International Journal of Virtual Reality*, 8(2):83–90. in press.

[23] Cho, M., Lee, J., and Lee, K. M. (2010). Reweighted random walks for graph matching. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision (ECCV 2005)*, volume 6315 of *Lecture Notes in Computer Science*, pages 492–505. Springer.

[24] Chojnacki, W., Szpak, Z. L., Brooks, M. J., and van den Hengel, A. (2010). Multiple homography estimation with full consistency constraints. In *International Conference on Digital Image Computing: Techniques and Applications, DICTA 2010, Sydney, Australia, 1-3 December, 2010*, pages 480–485.

[25] Cotting, D., Fuchs, H., Ziegler, R., and Gross, M. H. (2005). Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *Computer Graphics Forum*, 24(3):705–714.

[26] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In Schmid, C., Soatto, S., and Tomasi, C., editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334.

[27] Dehos, J., Zeghers, E., Renaud, C., Rousselle, F., and Sarry, L. (2008). Radiometric compensation for a low-cost immersive projection system. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pages 130–133, New York, NY, USA. ACM.

[28] Demirkus, M., Clark, J. J., and Arbel, T. (2013). Robust semi-automatic head pose labeling for real-world face video sequences. *Multimedia Tools and Applications*, pages 1–29.

[29] Deng, H., Zhang, W., Mortensen, E. N., Dietterich, T. G., and Shapiro, L. G. (2007). Principal curvature-based region detector for object recognition. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*.

[30] Donoser, M. and Bischof, H. (2006). Efficient maximally stable extremal region (MSER) tracking. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 553–560.

[31] Drouin, M.-A., Jodoin, P., and Prémont, J. (2010). Camera-projector matching using an unstructured video stream. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 33–40.

[32] Elfiky, N. M., Khan, F. S., van de Weijer, J., and Gonzàlez, J. (2012). Discriminative compact pyramids for object and scene recognition. *Pattern Recognition*, 45(4):1627–1636.

[33] Evans, C. (2009). Notes on the opensurf library. Technical report, University of Bristol.

[34] Fang, J., Varbanescu, A. L., and Sips, H. (2011). A Comprehensive Performance Comparison of CUDA and OpenCL. In *Proceedings of the 2011 International Conference on Parallel Processing*, ICPP '11, pages 216–225, Washington, DC, USA. IEEE Computer Society.

[35] Favre-Brun, A., Jacquemin, C., and Caye, V. (2012). Revealing the "spirit of the place": Genius Loci, a spatial augmented reality performance based on 3D data and historical hypotheses. In *18th International Conference on Virtual Systems and Multimedia, VSMM 2012, Milan, Italy, September 2-5, 2012*, pages 103–108.

[36] Fechteler, P. and Eisert, P. (2008). Adaptive color classification for structured light systems. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–7.

[37] Finlayson, G. D., Drew, M. S., and Funt, B. V. (1994). Spectral sharpening: sensor transformations for improved color constancy. *Journal of the Optical Society of America A*, 11(5):1553–1563.

[38] Finlayson, G. D., Hordley, S. D., and Xu, R. (2005). Convex programming colour constancy with a diagonal-offset model. In *Proceedings of the 2005 International Conference on Image Processing, ICIP 2005, Genoa, Italy, September 11-14, 2005*, pages 948–951.

[39] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[40] Forster, F. (2006). A high-resolution and high accuracy real-time 3d sensor based on structured light. In *International Symposium 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 208–215. IEEE Computer Society.

[41] Fujii, K., Grossberg, M. D., and Nayar, S. K. (2005). A projectorcamera system with real-time photometric adaptation for dynamic environments. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 814–821.

[42] Funt, B., Ciurea, F., and Mccann, J. (2000). Retinex in Matlab. In *Journal of Electronic Imaging*, pages 112–121.

[43] Gavaghan, K., Peterhans, M., Oliveira-Santos, T., and Weber, S. (2011). A portable image overlay projection device for computer-aided open liver surgery. *IEEE Transactions on Biomedical Engineering*, 58(6):1855–1864.

[44] Geusebroek, J., van den Boomgaard, R., Smeulders, A. W. M., and Dev, A. (2000). Color and scale: The spatial structure of color images. In *Computer Vision - ECCV 2000, 6th European Conference on Computer Vision, Dublin, Ireland, June 26 - July 1, 2000, Proceedings, Part I*, pages 331–341.

[45] Gibson, J. J. (1950). *The perception of the visual world*. Houghton Mifflin, Boston.

[46] Griffin, P. M., Narasimhan, L. S., and Yee, S. R. (1992). Generation of uniquely encoded light patterns for range data acquisition. *Pattern Recognition*, 25(6):609–616.

[47] Grossberg, M. and Nayar, S. (2003). What is the Space of Camera Response Functions? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 602–609.

[48] H. Hoppe and C. Kuebler and J. Raczkowsky and H. Woern and and S. Hassfeld (2002). A Clinical Prototype System for Projector-Based Augmented Reality: Calibration and Projection Methods. *Proceedings of the 16th International Congress and Exhibiton on Computer Assisted Radiology and Surgery (CARS) 2002, Springer*, page 1079.

[49] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[50] Heinly, J., Dunn, E., and Frahm, J.-M. (2012). Comparative evaluation of binary features. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, ECCV'12, pages 759–773, Berlin, Heidelberg. Springer-Verlag.

[51] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203.

[52] Igarashi, T., Moscovich, T., and Hughes, J. F. (2005). As-rigid-as-possible shape manipulation. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 1134–1141, New York, NY, USA. ACM.

[53] Ito, M. and Ishii, A. (1995). A three-level checkerboard pattern (tcp) projection method for curved surface measurement. *Pattern Recognition*, 28(1):27–40.

[54] Jacquemin, C., Chan, W. K., and Courgeon, M. (2010). Bateau ivre: an artistic markerless outdoor mobile augmented reality installation on a riverboat. In Bimbo, A. D., Chang, S.-F., and Smeulders, A. W. M., editors, *ACM Multimedia*, pages 1353–1362. ACM.

[55] Jones, B. R., Benko, H., Ofek, E., and Wilson, A. D. (2013). Illumiroom: Peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 869–878, New York, NY, USA. ACM.

[56] Kanan, C., Flores, A., and Cottrell, G. W. (2010). Color constancy algorithms for object and face recognition. In Bebis, G., Boyle, R. D., Parvin, B., Koracin, D., Chung, R., Hammoud, R. I., Hussain, M., Tan, K.-H., Crawfis, R., Thalmann, D., Kao, D., and Avila, L., editors, *ISVC (1)*, volume 6453 of *Lecture Notes in Computer Science*, pages 199–210. Springer.

[57] Kanazawa, Y. and Kawakami, H. (2004). Detection of planar regions with uncalibrated stereo using distribution of feature points. In *British Machine Vision Conference (Kingston upon)*, pages 247–256.

[58] Kawasaki, H., Furukawa, R., Sagawa, R., and Yagi, Y. (2008). Demo: Dynamic scene shape reconstruction using a single structured light pattern. In *IEEE Computer Vision and Pattern Recognition (CVPR08) Demo session*.

[59] Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 506–513, Washington, DC, USA. IEEE Computer Society.

[60] Khan, F. S., Anwer, R. M., van de Weijer, J., Bagdanov, A. D., López, A. M., and Felsberg, M. (2013). Coloring action recognition in still images. *International Journal of Computer Vision*, 105(3):205–221.

[61] Khan, F. S., de Weijer, J. V., Bagdanov, A. D., and Felsberg, M. (2014). Scale coding bag-of-words for action recognition. In *Proceedings of International Conference on Pattern Recognition (ICPR)*.

[62] Koninckx, T. P. and Gool, L. V. (2006). Real-time range acquisition by adaptive structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):432–445.

[63] Kooi, T., de Sorbier, F., and Saito, H. (2012a). Colour descriptors for tracking in spatial augmented reality. In Park, J.-I. and Kim, J., editors, *ACCV Workshops (2)*, volume 7729 of *Lecture Notes in Computer Science*, pages 387–399. Springer.

[64] Kooi, T., de Sorbier, F., and Saito, H. (2012b). Colour descriptors for tracking in spatial augmented reality. In Park, J.-I. and Kim, J., editors, *ACCV Workshops (2)*, volume 7729 of *Lecture Notes in Computer Science*, pages 387–399. Springer.

[65] Leordeanu, M. and Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision (ICCV 2005)*, pages 1482–1489. IEEE Computer Society.

[66] Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Washington, DC, USA. IEEE Computer Society.

[67] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168.

[68] Lin, J.-F. and Su, X. (1995). Two-dimensional fourier transform profilometry for the automatic measurement of three-dimensional object shapes. *Optical Engineering*, 34(11):3297–3302.

[69] Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA.

[70] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA. IEEE Computer Society.

[71] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[72] 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 - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[73] Luo Juan, O. G. (2009). A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3:143–152.

[74] Mao, X., Chen, W., Su, X., Xu, G., and Bian, X. (2007). Fourier transform profilometry based on a projecting-imaging model. *Journal of the Optical Society of America A*, 24(12):3735–3740.

[75] Maruyama, M. and Abe, S. (1993). Range sensing by projecting multiple slits with random cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):647–651.

[76] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Journal of Image and Vision Computing*, 22(10):761–767.

[77] Mazin, B., Delon, J., and Gousseau, Y. (2012). Combining color and geometry for local image matching. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 2667–2680.

[78] Menk, C., Jundt, E., and Koch, R. (2010). Evaluation of Geometric Registration Methods for Using Spatial Augmented Reality in the Automotive Industry. In Koch, R., Kolb, A., and Rezk-Salama, C., editors, *Vision, Modeling, and Visualization (2010)*. The Eurographics Association.

[79] Michelsen, J. and Björk1, S. (2014). The rooms creating immersive experiences through projected augmented reality. In *Proceedings of the 9th Conference on the Foundations of Digital Games*.

[80] Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 128–142, London, UK, UK. Springer-Verlag.

[81] Mikolajczyk, K. and Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[82] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 27(10):1615–1630.

[83] Miksik, O. and Mikolajczyk, K. (2012). Evaluation of local detectors and descriptors for fast feature matching. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 2681–2684.

[84] Mine, M. R., van Baar, J., Grundhofer, A., Rose, D., and Yang, B. (2012). Projection-Based Augmented Reality in Disney Theme Parks. *IEEE Computer*, 45(7):32–40.

[85] Morano, R. A., Ozturk, C., Conn, R., Dubin, S., Zietz, S., and Nissanov, J. (1998). Structured light using pseudorandom codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 20(3):322–327.

[86] Morita, H., Yajima, K., and Sakata, S. (1988). Reconstruction of surfaces of 3-d objects by m-array pattern projection method. In *Second International Conference on Computer Vision, ICCV 1988. Tampa, Florida, USA, 5-8 December, 1988, Proceedings*, pages 468–473.

[87] Mortensen, E. N., Deng, H., and Shapiro, L. (2005). A SIFT descriptor with global context. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 184–190, Washington, DC, USA. IEEE Computer Society.

[88] Nannen, V. and Oliver, G. (2012). Optimal number of image keypoints for real time visual odometry. In *IFAC Workshop on Navigation, Guidance & Control of Underwater Vehicles, NGCUV 2012. Porto, Portugal*, pages 331–336. IFAC.

[89] Nayar, S., Peri, H., Grossberg, M., and Belhumeur, P. (2003). A Projection System with Radiometric Compensation for Screen Imperfections. In *ICCV Workshop on Projector-Camera Systems (PROCAMS)*.

[90] Ortiz, R. (2012). Freak: Fast retina keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517, Washington, DC, USA. IEEE Computer Society.

[91] Oyallon, E. and Rabin, J. (2015). An Analysis of the SURF Method. *Image Processing On Line*, 5:176–218.

[92] p. Tardif, J., Roy, S., and Meunier, J. (2003). Projector-based augmented reality in surgery without calibration.

[93] Pagès, J., Collewet, C., Chaumette, F., and Salvi, J. (2006). An approach to visual servoing based on coded light. In *IEEE Int. Conf. on Robotics and Automation, ICRA'2006*, pages 4118–4123, Orlando, Florida.

[94] Pagès, J., Salvi, J., Collewet, C., and Forest, J. (2005). Optimised De Bruijn patterns for one-shot shape acquisition. *Journal of Image and Vision Computing*, 23(8):707–720.

[95] Pagès, J., Salvi, J., and Forest, J. (2004). A New Optimised De Bruijn Coding Strategy for Structured Light Patterns. In *ICPR (4)*, pages 284–287. IEEE Computer Society.

[96] Park, H., hyun Lee, M., jun Kim, S., and il Park, J. (2006a). Contrast enhancement in direct-projected augmented reality. *2012 IEEE International Conference on Multimedia and Expo*, 0:1313–1316.

[97] Park, H., Lee, M., Seo, B., Park, J., Jeong, M., Park, T., Lee, Y., and Kim, S. (2008). Simultaneous geometric and radiometric adaptation to dynamic surfaces with a mobile projector-camera system. 18(1):110–115.

[98] Park, H., Lee, M.-H., Seo, B.-K., and Park, J.-I. (2006b). Undistorted projection onto dynamic surface. In Chang, L.-W. and Lie, W.-N., editors, *PSIVT*, volume 4319 of *Lecture Notes in Computer Science*, pages 582–590. Springer.

[99] Porter, S., Marner, M. R., Smith, R. T., Zucco, J., and Thomas, B. H. (2010). Validating spatial augmented reality for interactive rapid prototyping. In *9th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2010, Seoul, Korea, 13-16 October 2010*, pages 265–266.

[100] Ridel, B., Reuter, P., Laviole, J., Mellado, N., Couture, N., and Granier, X. (2014). The Revealing Flashlight: Interactive spatial augmented reality for detail exploration of cultural heritage artifacts. *Journal on Computing and Cultural Heritage*, 7(2):1–18.

[101] Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1508–1515, Washington, DC, USA. IEEE Computer Society.

[102] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA. IEEE Computer Society.

[103] Salvi, J., Batlle, J., and Mouaddib, E. M. (1998). A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19(11):1055–1065.

[104] Salvi, J., Fernandez, S., Pribanic, T., and Llado, X. (2010). A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43(8):2666–2680.

[105] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172.

[106] Schulz, A., Jung, F., Hartte, S., Trick, D., Wojek, C., Schindler, K., Ackermann, J., and Goesele, M. (2010). CUDA SURF-A real-time implementation for SURF.

[107] Setkov, A., Carillo, F. M., Gouiffès, M., Jacquemin, C., Vanrell, M., and Baldrich, R. (2015). Dacimpro: A novel database of acquired image projections and its application to object recognition. In Bebis, G., editor, *ISVC (2)*, volume 9475 of *Lecture Notes in Computer Science*, pages 463–473. Springer.

[108] Setkov, A., Gouiffès, M., and Jacquemin, C. (2013). Color invariant feature matching for image geometric correction. In *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, MIRAGE '13, pages 7:1–7:8, New York, NY, USA. ACM.

[109] Setkov, A., Gouiff'es, M., and Jacquemin, C. (2016). Evaluation of color descriptors for projector-camera systems. *Journal of Visual Communication and Image Representation*.

[110] Shafer, S. A. (1992). Color. chapter Using color to separate reflection components, pages 43–51. Jones and Bartlett Publishers, Inc., USA.

[111] Singhal, P., Deshpande, A., Pandya, H., Reddy, N. D., and Krishna, K. M. (2014). Top down approach to detect multiple planes from pair of images. In *Proceedings of the 2014 Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP'14, Bangalore, India, December 14-18, 2014*, pages 53:1–53:8.

[112] Smith, S. M. and Brady, J. M. (1997). Susan—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

[113] Srinivasan, V., Liu, H. C., and Halioua, M. (1985). Automated phase-measuring profilometry: a phase mapping approach. *Journal of Applied Optics*, 24(2):185–188.

[114] Tajima, J. and Iwakawa, M. (1990). 3-d data acquisition by rainbow range finder. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, volume i, pages 309–313 vol.1.

[115] Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 32(5).

[116] Torr, P. H. S. and Zisserman, A. (1996). Robust parameterization and computation of the trifocal tensor. In *BMVC*. British Machine Vision Association.

[117] Torr, P. H. S. and Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156.

[118] van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 32(9):1582–1596.

[119] van de Weijer, J. and Schmid, C. (2006). Coloring local feature extraction. In Leonardis, A., Bischof, H., and Pinz, A., editors, *ECCV (2)*, volume 3952 of *Lecture Notes in Computer Science*, pages 334–348. Springer.

[120] Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/.

[121] Vincent, E. and Laganiere, R. (2001). Detecting planar homographies in an image pair. *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, pages 182–187.

[122] Wen, R., Nguyen, B. P., Chng, C.-B., and Chui, C.-K. (2013). In situ spatial ar surgical planning using projector-kinect system. In *Proceedings of the Fourth Symposium on Information and Communication Technology*, SoICT '13, pages 164–171, New York, NY, USA. ACM.

[123] Wen, R., Yang, L., Chui, C.-K., Lim, K.-B., and Chang, S. K. Y. (2010). Intraoperative visual guidance and control interface for augmented reality robotic surgery. In *ICCA*, pages 947–952. IEEE.

[124] West, G. and Brill, M. (1982). Necessary and sufficient conditions for von kries chromatic adaptation to give colour constancy. *Journal of Mathematical Biology*, 15:249–258.

[125] Willis, K. D. D., Poupyrev, I., and Shiratori, T. (2011). Motionbeam: a metaphor for character interaction with handheld projectors. In Tan, D. S., Amershi, S., Begole, B., Kellogg, W. A., and Tungare, M., editors, *CHI*, pages 1031–1040. ACM.

[126] Wust, C. and Capson, D. W. (1991). Surface profile measurement using color fringe projection. *Mach. Vis. Appl.*, 4(3):193–203.

[127] Yamanaka, T., Sakaue, F., and Sato, J. (2010). Adaptive image projection onto nonplanar screen using projector-camera systems. In *International Conference on Pattern Recognition*, pages 307–310. IEEE.

[128] Yamazaki, S., Mochimaru, M., and Kanade, T. (2011). Simultaneous self-calibration of a projector and a camera using structured light. In *Proc. Projector Camera Systems*, pages 67–74.

[129] Yang, R. and Welch, G. (2001). Automatic and continuous projector display surface estimation using everyday imagery. In *WSCG*, pages 328–335.

[130] Yu, G. and Morel, J.-M. (2011). ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 1.

[131] Yue, Hui-Min, S. X.-Y. L. Y.-Z. (2007). Fourier transform profilometry based on composite structured light pattern. *Optics and Laser Technology*, 39(6):1170–1175.

[132] yves Bouguet, J. (2000). Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*.

[133] Zaragoza, J., Chin, T.-J., Tran, Q.-H., Brown, M. S., and Suter, D. (2014). As-Projective-As-Possible Image Stitching with Moving DLT. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1–1.

[134] Zhang, D. and Lu, G. (2003). Evaluation of similarity measurement for image retrieval. In *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, volume 2, pages 928–931 Vol.2. IEEE.

[135] Zhang, Z. (1997). Parameter estimation techniques: a tutorial with application to conic fitting. *Journal of Image and Vision Computing*, 15(1):59–76.

[136] Zhou, F. and De la Torre, F. (2012). Factorized graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[137] Zhou, Z., Jin, H., and Ma, Y. (2012). Robust plane-based structure from motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 1482–1489.

[138] Zollmann, Langlotz, S., and Tobiasand Bimber, O. (2007). Passive-active geometric calibration for view-dependent projections onto arbitrary surfaces. *JVRB - Journal of Virtual Reality and Broadcasting*, 4(6):10.

[139] Zuliani, M., Kenney, C. S., and Manjunath, B. S. (2005). The multiransac algorithm and its application to detect planar homographies. In *ICIP (3)*, pages 153–156. IEEE.