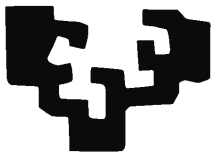

Contributions to Local Feature Extraction, Description and Matching in 2D Images

By

Iñigo Barandiaran Martirena

Dissertation submitted to the department of Computer Science and Artificial Intelligence in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

PhD Advisors:

Prof. Dr. Manuel Graña Romay At The University of the Basque Country

and

Dr. Marcos Nieto Doncel At Vicomtech-Ik4

Universidad del País Vasco
Euskal Herriko Unibertsitatea
Donostia - San Sebastian

2013

Universidad del País Vasco Euskal Herriko Unibertsitatea

AUTORIZACION DEL/LA DIRECTOR/A DE TESIS

PARA SU PRESENTACION

Dr/a. _____con N.I.F._____

como Director/a de la Tesis Doctoral: _____

_____

_____

realizada en el Departamento _____

_____

por el Doctorando Don/ña. _____ ,

autorizo la presentación de la citada Tesis Doctoral, dado que reúne las condiciones necesarias para su defensa.

En _____a _____de _____de _____

EL/LA DIRECTOR/A DE LA TESIS

Fdo.: _____

Universidad del País Vasco Euskal Herriko Unibertsitatea

CONFORMIDAD DEL DEPARTAMENTO

El Consejo del Departamento de _____

_____

en reunión celebrada el día ____ de _____ de _____ ha acordado dar la conformidad a la admisión a trámite de presentación de la Tesis Doctoral titulada: _____

_____

_____

dirigida por el/la Dr/a. _____

_____

y presentada por Don/ña. _____

ante este Departamento.

En _____a _____de _____de _____

Vº Bº DIRECTOR/A DEL DEPARTAMENTO SECRETARIO/A DEL DEPARTAMENTO

Fdo.: _____ Fdo.: _____

Universidad del País Vasco Euskal Herriko Unibertsitatea

ACTA DE GRADO DE DOCTOR

ACTA DE DEFENSA DE TESIS DOCTORAL

DOCTORANDO DON/ÑA. _____

TITULO DE LA TESIS: _____

_____

_____

El Tribunal designado por la Subcomisión de Doctorado de la UPV/EHU para calificar la Tesis Doctoral arriba indicada y reunido en el día de la fecha, una vez efectuada la defensa por el doctorando y contestadas las objeciones y/o sugerencias que se le han formulado, ha otorgado por_____la calificación de:

unanimidad ó mayoría

Idioma/s defensa: _____

En _____a _____de _____de _____

EL/LA PRESIDENTE/A, EL/LA SECRETARIO/A,

Fdo.: Fdo.:

Dr/a: _____ Dr/a: _____

VOCAL 1º, VOCAL 2º, VOCAL 3º,

Fdo.: Fdo.: Fdo.:

Dr/a: _____Dr/a:_____Dr/a:_____

EL/LA DOCTORANDO/A,

Fdo.: _____

Contributions to Local Feature Extraction, Description and Matching in 2D Images

by

Iñigo Barandiaran Martirena

Submitted to the Department of Computer Science and Artificial Intelligence in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

Nowadays, Computer Vision is becoming a very important research topic because of its great applicability and usefulness in many and heterogeneous areas such as medical or bio-medical, astronomy, industrial, or educational sectors, as well as in entertainment industry or even in our every day life. Despite this variety of application areas, a great number of Computer Vision based applications integrates at some point of their processing pipeline, the identification, extraction and matching of some type local features across images. Local features are well suited to image recognition and matching because of robustness against noise and geometric or photometric transformations, providing concise representations of objects in the image.Several interest point detectors and local feature descriptors, as well as strategies and algorithms for matching them, have been presented since in the last decade. Though a lot of progress has been done in this field, the problem of matching points across different images is far to be fully solved. This Thesis aims to contribute to the field of local image feature extraction and matching by giving useful insight of state-of-the-art, serving as a supplement to existing comparative studies about interest point extraction, feature description and matching, as well as by contributing with some new approaches regarding this technologies, such as a new local image descriptor based on the Trace transform. We also contribute to the field by providing the scientific community with a verified and well designed tool and image data sets, that allow comparing results obtained from different approaches regarding interest point extraction, feature descriptor or descriptor matching.

Keywords: *Interest Point Extraction, Feature Descriptors, Image Feature Matching, Robust Homography Estimation, Random Forest, Evaluation Framework.*

Agradecimientos

Me gustaría dar las gracias a todas las personas que, de una forma u otra, han participado en la consecución de esta Tesis. Me gustaría agradecer de manera especial a mi director de Tesis, el profesor Manuel Graña por su asesoramiento, experiencia e insistencia en que realice y defienda la Tesis. Sin sus collejas, seguro que no estaría escribiendo estas líneas. Quiero agradecer también a los directores de Vicomtech-IK4 y del departamento Edurne, Julián, Jorge, Shabs y Céline. Sin su confianza en mi y sin su apoyo no habría podido andar todo este camino. Agradecer también a todos los compañeros de Vicomtech-IK4 que me han ayudado durante este proceso, y de los que he aprendido tanto, especiálmente a Igor García con el que he pasado tantos buenos momentos, tratando de entender lo que DITEC quería decirnos. Tenemos una cerveza pendiente cuando todo esto acabe. Gracias a mis amigos Asier, David, Joserra y Luisma. Finalmente, y de manera muy especial, dar las gracias a mi familia y a Mai por su apoyo y cariño. Gracias.

Iñigo Barandiaran Martirena

Acknowledgements

I want to thank everybody involved in any way with my Thesis and my life. My special thanks to my advisor, Prof. Manuel Graña. Without him this Thesis is impossible. Thanks to Vicomtech-IK4 Directors Edurne, Julián, Jorge, Shabs and Céline. They had confidence in me and gave me the necessary time and resources for developing this work. Also I want to thank all my workmates and collegues at Vicomtech-IK4 Research Center who helped me in this task, specially to Igor García with whom I spent so many great moments trying to understand what DITEC was telling us. We have a pending beer when all this finishes. Thanks to my friends Asier, David, Joserra and Luisma. Also, and most importantly, I want to thank my family and Mai for their support in my life. Thanks.

Iñigo Barandiaran Martirena

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides the general introduction to the Thesis intended to allow a quick appraisal of its contents, contributions, supporting publications and structure. Its structure is as follows: Section 1.1 provides some general guidelines and the main motivations behind this Thesis. Section 1.2 enumerates the main technical and methodological contributions of the Thesis. Section 1.3 enumerates the publications obtained in the process of realization of the Thesis. Finally, Section 1.4 details the structure of the Thesis.

1.1 Motivation

Computer Vision is a relatively novel field of research and application closely related to numerous other areas such as Machine Learning, Physics, Control Systems, or Computer Science. Today, Computer Vision is becoming a very important research topic because of its great applicability and usefulness in many and heterogeneous areas such as medical or bio-medical, astronomy, industrial, or educational sectors, as well as in entertainment industry or even in our every day life. Despite this variety of application areas, a great number of Computer Vision based applications including registration, 3D reconstruction, motion estimation, image matching and retrieval, object and action recognition or image stitching integrates at some point of their processing pipeline, the identification, extraction and matching of some type of feature points or local features across images. Local features are well suited to image recognition and matching because they are robust to partial occlusion, clutter, and geometric or photometric transformations, providing concise

representations of objects in the image.

Local feature extraction, description and matching are low level information extraction processes. Following some kind of bottom-up approach, these low level processes are followed up by some other higher level processes where this basic information is converted, translated or interpreted until a conclusion at a semantic level is reached, either some meaning is recovered, some recognition is performed, or some high-level structure is estimated. In this way, feature extraction and matching processes often form the basis of many image analysis mechanisms, hence its of critical importance for the success of many Computer Vision applications.

Several interest point detectors and local feature descriptors, as well as strategies and algorithms for matching them have been presented in the last decades. Though a lot of progress has been made in this field, the problem of matching points across different images is far from being fully solved. The performance of the algorithms, both in terms of computational efficiency and robustness, are closely related to the complexity and type of the scenes, as well as the transformations between the images.

This Thesis aims to contribute to the field of image local feature extraction and matching providing useful insight on the state-of-the-art, serving as a supplement to existing comparative studies about interest point extraction, feature description and matching, as well as by contributing with some new approaches, such as a new local image descriptor based on the Trace transform. The Thesis also contribute to the field by providing the scientific community with a verified and well designed validation/evaluation tool and image data sets, that allow comparing results obtained from different approaches to interest point extraction, feature descriptor or descriptor matching. Moreover, results in this Thesis can be used to facilitate the selection of appropriate point detectors, region descriptors and matching strategies for specific target applications, as well as to facilitate the development of future related research lines.

1.2 Technical and Methodological Contributions

The following technical and methodological contributions in the field of the image local feature detection have been generated within this Thesis:

- A detailed analysis and additional insight into state-of-the-art interest point extractors. We evaluated the repeatability of several detectors by using an

in-house benchmark data set with known ground truth information for validation, under different geometric and photometric conditions.

- We provide a review of the most relevant state-of-the-art interest point description approaches. In addition, we propose a new contribution to local description based on the Trace transform. Finally, we performed a detailed evaluation of state-of-the-art approaches, along with our approach, on how they perform against several geometric and photometric transformations.

- A contribution to local interest point description matching based on machine learning techniques. Classifier ensembles like Random Forest were evaluated as an interest point matching strategy, solving the problem of invariance to scale, rotation and affine geometric transformations. This approach was integrated in an imaging pipeline for marker-less tracking applied to augmented reality applications.

- A detailed design and implementation of a new testing framework for the validation and evaluation of different interest point extraction, feature description and matching algorithms. This framework is intended to contribute to the development of these techniques by distributing it as Open Source software, hence every experiment described in this Thesis can be reproduced in the same conditions by any researcher in the field, as well as compared them with ongoing new research developments.

- Along with the evaluation framework, we designed and implemented a new image data set covering different geometric and photometric transformation for evaluating interest point detectors and local image descriptors. Proposed image data set was acquired in well-controlled laboratory conditions, ensuring precise Ground-Truth data generation.

- An overview and an evaluation of the most common approaches for robust estimation based on random sampling. We measured several aspects of such approaches like computation time or accuracy applied on the robust estimation of planar 2D homographies.

1.2.1 Related Projects

The research and scientific contributions generated during this Thesis are related with the development inside the IK4-Vicomtech research center of several indus-

trial and applied research local and European projects such as:

IMPROVE

The aim of this project was to improve lightweight near-to-the-eye displays and tiled stereoscopic large size displays. The improvements on the hardware level consisted in developing a unique stereoscopic head mounted display (HMD) using emerging display technology such as OLEDs. For tiled stereoscopic large screen displays improved calibration techniques were developed to ease and accelerate their use. On the software level improvements comprise the fidelity of the content to be displayed (rendering quality), the interfacing between the user and the displays through innovative 2D/3D interaction techniques for mixed realities and advanced tracking systems.

Figure 1.1: Object tracking with see-through HMD.

The achievements of IMPROVE were integrated into a collaborative mixed reality product development environment, showcased and evaluated in two application scenarios: collaborative product design in the car industry and architectural design.

VISION

During the last few years, videoconferencing has become a widely extended application. Companies and individuals use this technology to reduce transportation

Figure 1.2: Augmented reality scene.

costs and to improve communication between distant parties. However, traditional 2D videoconferencing still fails to produce natural impressions to the remote attendees. 3D videoconferencing is a logical evolution: a better feeling of presence is provided to conferees by leading them to believe that they are closer to each others. Currently, the emerging 3D displays and the increase of mainstream hardware computation capabilities make telepresence feasible.

Many important topics related to videoconferencing must be addressed to enhance user experience. One such issue is eye contact or gaze. Research like [9, 10] show that gaze is one of the most important non-verbal cues, responsible for providing feedback or expressing feelings and attitudes. Another problem related with video-conference is the lack of perception of depth. In a 2D video-conference, traditional displays are used. The absence of depth information in these type of displays in addition to the problem of gaze results in an unnatural experience. An accurate and efficient depth map estimation mechanism is needed for both problems to be addressed. For example, depth information is used by auto-stereoscopic displays to render multiple novel views. These generated views can be used to tackle the problem of eye-contact, by generating views that agreed with the user eye-sight. Moreover, auto-estereoscopic displays typically use depth maps information to generate 3D perception, as shown in Figure 1.3(right). Depth estimation process is done by identifying point correspondences between pixels in two images, hence computing disparity. Given the location of corresponding pixels in the images, their 3D coordinates can be retrieved by means of triangulation.

Disparity is commonly used to describe inverse depth in Computer Vision and to measure the perceived spatial shift of a feature observed from close camera viewpoints. Stereo correspondence techniques often calculate a disparity function

Figure 1.3: Image acquisition (left). Estimated depth map with our system (right).

$d(x;y)$ relating target and reference images, so that the $(x;y)$ coordinates of the disparity space match the pixel coordinates of the reference image. Stereo methods commonly use a pair of images taken with a known camera geometry to generate a dense disparity map with estimates at each pixel. This dense output is useful for applications requiring depth values even in difficult regions like occlusions and textureless areas.

The aim of this project was the development of a system capable of estimate accurate and efficient disparity maps by fusing the information stereo camera rigs. Disparity maps were integrated in auto-estereoscopic displays for applications like 3D real-time Videoconferencing.

eVirtual

This project aims to develop different methodologies and techniques regarding audiovisual effects for inmersive experiences. Our main objective was to develop image analysis techniques for generating 360º video panoramas like in Figure 1.4, similarly to Google Street View [11]. This video panorama are then mapped to a sphere that can be visualized interactively by the user through a Web browser.

In order to accomplish this task, we focused on the evaluation of different approaches of interest point extraction and feature description for the estimation of spatial transformation between cameras capturing the same scene. Estimation of such geometric relationship between sensors allow to stitch live video streams, conforming video panorama. For this purpose, we used Lady Bug device, depicted in Figure 1.4(right), consisting of 6 different cameras, of 2 Megapixels each. One of the most important challenges is dealing with the high image geometric distortion due to the short focal length of each sensor, needed for covering high field-of-view.

Figure 1.4: (left)360º Panorama, (right) Ladybug camera.

In addition to the development of local feature extraction for stitching video streams, we developed mechanism for optimizing user experience and connection bandwidth usage by generating scale-space analysis and computing different levels of details by tessellating original video panorama. This tessellation allows to optimize the loading of video panorama during playback in the client side (web browser) while also optimize user experience by reducing pause times when loading new content.

RETINA

The strong development of image acquisition systems oriented to medical sector is favoring the emergence of new clinical solutions that advance and improve traditional diagnostic procedures. Mechanisms such as X-ray angiography, magnetic resonance angiography or computed tomography, along with other image acquisition mechanisms, currently provide vital information to carry out certain processes of evaluation and diagnosis, allowing to reduce costs thanks to early detection of diseases, and the consequent reduction in hospitalization time.

Currently, there is an increasing scientific evidence regarding the role played by micro-vascular diseases in relation to the pathologies associated with macro-vascular structures. Studies such as [12] have shown how a condition in coronary microvascular structure, may cause serious heart failure with risk of heart attack and death, without any existence of pathology in coronary macrovascular structures, so that periodic checks of such structures may not reveal the existence of pathology. Moreover, certain dysfunctions in skin microvascularity, which is estimated to be representative of the entire micro-human circulatory system, have been associated with increased risks of heart attack. However, studies related to such microvascularization are small respect to its population because of the needs of

laborious and very invasive techniques. For this reason, researchers are now looking for alternatives and mechanisms that allow to accurately analyze microvascular structures in a non invasive way. For example, the retinal microcirculation has great potential for in-vivo analysis with minimally invasive techniques such structures intended to convey or represent other vascular structures, and to infer from them a wide range of pathological situations.

Recent studies like [12] which point to the importance and attention it is currently receiving the fundus, as a substantial part of a large number of diagnostic procedures for a wide variety of pathologies. As discussed, little is known regarding microvascular dysfunction in coronary heart disease because of difficulties in studying the coronary microcirculation directly. However, the retina is an anatomical region where images can be obtained directly from the capillaries, which provides great opportunities for invivo study of the structure and the pathology of the human circulation, as well as the ability to detect changes related microvascular with the development of cardiovascular diseases, among others. Currently many clinical studies link retinal vascular signs with coronary heart disease, and highlights the abundant scientific evidence found that retinal vascular signs may reflect the state of the coronary microvasculature. The retinal photographs offer us lasting records that control the longitudinal changes of these manifestations and vascular health in general. A clear example is diabetic retinopathy. This condition is the leading cause of blindness in the population. This pathology is characterized by a set of retinal damage caused by complications of diabetes mellitus, a disease that causes an abnormal elevation of glucose concentrations in blood. Early symptoms of diabetic retinopathy appear as small changes in the micro-retinal vascularization, as micro-aneurysms, accompanied by the appearance of exudates. Changes in blood pressure, together with other mechanisms, modify the blood supply to critical structures such as the optic nerve, causing the loss of vision. Early detection of these conditions can favor more effective and early treatments, thus slowing down the progressive loss of vision. Currently, the most advanced technique for the diagnosis of this type of disease is fundus imaging. This technique allows to obtain high resolution images of the internal structures of the retina, such as the microvascular tree or the optic disc, as shown in Figure 1.5.

Nowadays various clinical studies are being conducted trying to evaluate the retinal microvascular analysis as an indicator or bio-marker for the early detection of cerebrovascular strokes (acv). These strokes can be almost negligible for both patients (transient stroke) as well as generators of deficiencies and physical scars

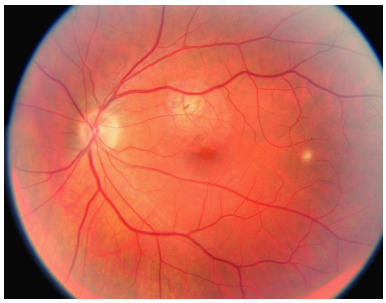

Figure 1.5: Image of the retina.

and / or motor, or even cause death. Regardless of the degree of severity of the stroke, the damage caused by them can be identified through magnetic resonance imaging or computed tomography. Some of these strokes are also known as silent strokes that caused by vascular occlusion are discovered incidentally on MRI or CT scan without any detectable neurological signs in apparently healthy patients. Studies like [13] indicates that the risk of cardiac stroke is greater when there are evidences of retinal microvascular abnormalities, in people who have had silent strokes.

This project aims to develop a tool image analysis software capable of identifying, extracting and quantifying anatomical features from fundus images, which can be exploited in clinical research processes. More precisely, these research processes are focused on the identification and correlation of different detectable abnormalities in the retinal microvascular system with the appearance of silent cerebral infarction.

1.3 Publications

- Barandiaran, I.; Cortes, C; Nieto, M.; Graña, M. and Ruiz, O. A New Image Dataset and Evaluation Framework for Keypoint Extraction and Feature Descriptor Matching. Proceedings of International Conference on Computer Vision Theory and Application, pp. 252-257, (2013).

- Barandiaran, I., Maiz, O., Macia, I., Ugarte, J., Toram, P., and Vazquez, X.: Towards a Tool for Automatic Retinal Vessel Quantification in Early Detection of Silent Brain Infarction. To appear in Proceedings of Imaging and Applied Optics conference, (2013).

- Barandiaran, I.; García, I. and Graña, M. Evaluation of interest point detectors. **International Journal of Cybernetics and System**, 44(2):98-117, (2013).

- Olaizola, I.; Barandiaran, I.; Sierra, B. and Graña, M. DITEC:Experimental Analysis of an Image Characterization Method based on the Trace Transform. Proceedings of International Conference on Computer Vision Theory and Application, pp. 344-352, (2013).

- Acosta, D.; Barandiaran, I.; Congote, J. ;Ruiz, O.; Hoyos, A.; Graña, M. Tuning of Adaptive Weight Depth Map Generation Algorithms. **Journal of Mathematical Imaging and Vision**, (online 2012) `http://link.springer.com/article/10.1007%2Fs10851-012-0366-7`.

- Barandiaran, I.; Maclair, G.;Goienetxea, I.; Jauquicoa,C. and Graña, M. A Comparative Study of Classier Ensembles for Karyotyping. Proceedings of International Conference on Knowledge Based and Intelligent Information and Engineering Systems, pp.1400-1407, (2012).

- Barandiaran, I.; Congote, J.; Goenetxea, J.; Graña, M. and Ruiz, O. Evaluation of interest point detectors for image information extraction. Proceedings of International Conference on Knowledge Based and Intelligent Information and Engineering Systems, pp.2170-2179, (2012).

- Goienetxea, I.; Barandiaran, I.; Jauquicoa, C.; Maclair, G. and Graña, M. Image Analysis Pipeline for Automatic Karyotyping Hybrid Artificial Intelligent Systems. Proceedings of HAIS, Springer, pp.392-403, (2012).

- Hoyos, A.; Congote, J.; Barandiaran, I.; Acosta, D. and Ruiz, O. Statistical tuning of adaptive-weight depth map algorithm. Computer Analysis of Images and Patterns, Springer, pp. 563-572, (2011).

- Barandiaran, I.; Paloc, C. & Graña, M. Real-time optical markerless tracking for augmented reality applications. **Journal of Real-Time Image Processing**, Springer, Vol(5), pp. 129-138, (2010).

- Congote, J.; Barandiaran, I.; Barandiaran, J.; Montserrat, T.; Quelen, J.; Ferran, C.; Mindan, P.; Mur, O.; Tarres, F. and Ruiz, O. Real-time depth map

generation architecture for 3d videoconferencing. Proceedings of 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video, pp.1-4, (2010).

- Congote, J.; Barandiaran, I.; Barandiaran, J.; Nieto, M. and Ruiz, O. Face Reconstruction with structured light. Proceedings of International Conference on Computer Vision Theory and Application, pp.149-155, (2010).
- Barandiaran, I.; Macia, I.; Berckmann, E.; Wald, D.; Dupillier, M.; Paloc, C. and Graña, M. An automatic segmentation and reconstruction of mandibular structures from CT-data. Proceedings of IDEAL, Springer, pp. 649-655, (2009).
- Congote, J.; Moreno, A.; Barandiaran, I.; Barandiaran, J. and Ruiz, O. Adaptative cubical grid for isosurface extraction. Proceedings of International Conference on Computer Graphics Theory and Applications, pp. 21-26, (2009).
- Congote, J.; Barandiaran, J.; Barandiaran, I. and Ruiz, O. Realtime dense stereo matching with dynamic programming in CUDA. Proceedings of the 19th Spanish Congress of Graphical Informatics, pp. 231-234, (2009).
- Barandiaran, I.; Cottez, C.; Paloc, C. and Graña, M. Comparative evaluation of Random Forest and Fern classifiers for real-time feature matching. Proceedings of International Conference in Central Europe on Computer Graphics, visualization and Computer Vision, pp. 59-166, (2008).
- Barandiaran, I.; Cottez, C.; Paloc, C. and Grana, M. Random forest classifier for real-time optical markerless tracking. Proceedings of International Conference on Computer Vision Theory and Application, Vol(2), pp. 559-564, (2008).
- Santos, P.; Stork, A.; Gierlinger, T.; Pagani, A.; Paloc, C.; Barandarian, I.; Conti, G.; de Amicis, R.; Witzel, M.; Machui, O. and others. Improve: An innovative application for collaborative mobile mixed reality design review International Journal on Interactive Design and Manufacturing, Springer, Vol(1), pp. 115-126, (2007).
- Stork, A.; Santos, P.; Gierlinger, T.; Pagani, A.; Paloc, C.; Barandarian, I.; Conti, G.; Amicis, R.; Witzel, M.; Machui, O. and others. IMPROVE: An

innovative application for collaborative mobile mixed reality design review Proceedings of Virtual Concept, (2006).

- Paloc, C.; Barandiaran, I.; Carrasco, E. and Macia, I. Computer simulation of multi-leaf collimated fields for radiotherapy treatment planning verification. Proceedings of CARS, Elsevier, pp. 1281-1298, (2005).

1.4 Structure of the Thesis

The content of the Thesis is the following:

- Chapter 2: Introduces the general image processing problem related with interest point extraction, recognition and matching. Also this chapter gives an overview of the main technologies, techniques and background related with the main contributions of this Thesis.
- Chapter 3: Reviews the main state-of-the-art interest point detectors, reporting a detailed evaluation of their performance on an extensive inhouse developed benchmark.
- Chapter 4: A proposal of a new local image descriptor based on the Trace transform is given. Also, a review of the main state-of-the-art local feature descriptors is given. An exhaustive comparative evaluation of both the state-of-the-art and our innovative descriptor is reported.
- Chapter 5: Describes our contribution to interest point description matching based on Machine Learning techniques.
- Chapter 6: Concludes by giving the main contributions and conclusions extracted from this Thesis work. .

Complementarily, two Appendix are included in the Thesis.

- Appendix A: This appendix describes the experimental framework implemented during the Thesis, that was used for carry out the evaluations described in Chapters 3 and Chapter 4.
- Appendix B: Introduces state-of-the-art approaches for robust estimation based on random sampling, the RANSAC and related algorithms. Also, results of an evaluation about RANSAC algorithm and some of its variations and extensions are described.

Chapter 2

Background

This chapter is structured as follows: Section 2.1 gives a brief introduction about the context of interest point extraction, feature description and matching. Section 2.2 introduces interest point extraction mechanisms, and the scale-space framework. Section 2.3 describes local region description approaches. Section 2.4 describes the projective geometry fundamentals that are extensively used along the Thesis. Finally, Section 2.5 describes the methodology used along this Thesis for the evaluation of several interest point extraction and feature description approaches.

2.1 Introduction

Computer vision applications deal with information extraction from the images acquired by a camera sensor. Often, this information is summarized by a collection of local features composed of relevant pixels or regions having discriminant characteristics, i.e. retaining information about the structures in the imaged scene. Local features are image patterns which differ from their immediate neighborhoods. They are associated with the change of one or several image properties such as intensity, color or texture. Local features can be points, segments, lines, regions or blobs. Global features, on the other hand, describe the image as a whole, regardless of the content of isolated pixels. Color histogram [14] or dictionaries (bag-of-words) [15] are global features. Global features are affected by clutter and partial occlusion and are not suitable for spatial localization of objects.

Local features are well suited to image recognition and matching because they

are robust to partial occlusion, clutter, and geometric or photometric transformations, providing concise representations of objects in the image. Many different terms in the literature refer to the same structures, such as feature points, interest points, key points, or corner points. There has been a lot of work done in the last decade on local features for still image and video data leading to substantial improvements in many computer vision areas including registration, 3D reconstruction, motion estimation, image registration, matching and retrieval, object and action recognition, 3D object reconstruction, camera pose estimation, and image stitching. Local feature extraction techniques are usually found in the first image analysis stages of many computer vision applications and are considered as low level image information extractors.

Some computer vision applications need to identify a set of points to be matched setting correspondences between images. These applications share a common image processing pipeline, similat to the one depicted in Figure 2.1. This pipeline can be split into several processes:

- The first step is interest point (aka keypoint) extraction. This process selects a group of pixels (regions) where their surrounding or neighboring pixels retain enough information, that allow the regions to be identified afterwards. An in deep review of region detectors, and a measure for computing point repeatability are shown in [2, 16].

- Extracted keypoints are subjected to the next process converting the neighborhood of each point into a vector of values, known as descriptor. These descriptors act as identifiers of their corresponding interest points. The simplest descriptor consists in rearranging pixel values of a regular image patch surrounding a keypoint into a one-dimensional vector.

- Once every interest point is characterized by its corresponding descriptor, a matching process identifies cooresponding points between images, looking for the most similar descriptors by using some distance function such as Euclidean, Mahalanobis or Hamming, among others.

- The filtering process at the end of the pipeline is used for removing wrong keypoint matches. This process is carried out by applying temporal, spatial or geometric restrictions, allowing the identification of outliers with respect to an specific function, or model, such as homography estimation [17]. These

processes allow to identify or to separate true correct matches from the set of matches.

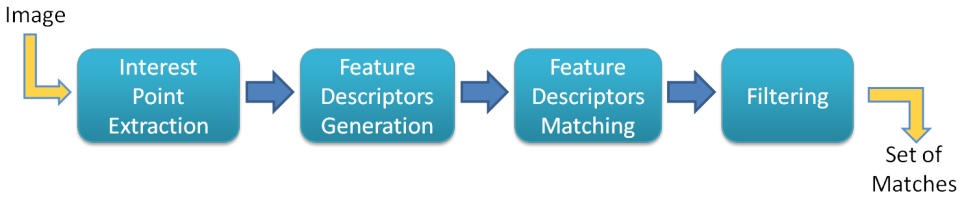

Figure 2.1: Image Pipeline for local feature extraction and matching.

The number of correct correspondences found by the algorithm depends on the nature of the descriptors, the texture content of the images, the transformation, either photometric or geometric, between images to be matched. Mainly, the percentage of correct correspondences found depends on the unequivocal identification of image regions by the region descriptor, and its robustness against image transformations, i.e. the ability to identify the same image region after an image transformation. Robust region descriptors are desirable because they provide more correspondences, which in their turn improve the robustness of transformation parameter estimation algorithms. For example, in the case of homography estimation, four keypoint correspondences are the minimum required to compute an estimation of homography parameters. However, the linear model solved for the estimation may be degenerate, or the correspondences very biased by chance. Providing more than four point correspondences allows to solve by least squares the linear model increasing its robustness.

The output of the image processing pipeline depicted in Figure 2.1 is a set of point correspondences between image a and b $(x_{ai} \leftrightarrow \widetilde{x}_{bj})$ that will be delivered to posterior processes until some high level conclusion is reached, such as object recognition, object 3D reconstruction for medical diagnosis [18], or augmented reality [19].

2.2 Interest point extraction

As explained before, local feature detection mechanisms are used to extract or detect interest points from source images. They usually provide information not only about the spatial image coordinates (x,y) of the points, but also about the

shape of their support regions. Feature detectors can be classified according to the extracted regions as corner detectors [20] or blob detectors [21]. A corner detector extracts local features defined as regions of the image with strong intensity variations along any direction. On the contrary, a blob detector detects blob-like structures, i.e. regions with locally uniform intensity values. However, these two classes of detectors are not crisply separated. Let us consider the Hessian matrix, it was first used as a corner detector since it finds corner points [20], but these are also usually localized at the boundaries of uniform regions, thus it could also be considered as a blob detector [22].

Interest point detectors should have the following properties [2],[23]:

- *Repeatability*: Given two images of the same scene taken under different viewing conditions, a high percentage of the features detected in both images should be found in both images. Repeatability explicitly compares the geometrical stability of the detected interest points between different images. An interest point is "repeated" if the projection of the same 3D real world point detected in the first image is also accurately detected in the second one. The repeatability rate is the percentage of total observed points that are detected in both images. Repeatability can be achieved in two different ways: either by invariance or by robustness. Repeatability is also called *stability* in some contexts, such as tracking [24].

- *Invariance*: When large deformations are expected between images, the preferred approach is to model them mathematically, developing feature detection methods that are unaffected by these mathematical transformations.

- *Robustness*: In case of small deformations, it often suffices to make feature detection methods less sensitive to such deformations, i.e., the accuracy of the detection may decrease, but still be effective.

- *Distinctiveness/informativeness*: The support regions of detected local features should be discriminant, so that they can be distinguished and matched. Distinctiveness [25] is based on the likelihood of a local gray value descriptor computed at the point within the population of all observed interest point descriptors. Descriptors characterize the local shape of the image at the interest points.

- *Locality*: The features should be local reducing the probability of occlusion

and allowing simple model approximations such as affine approximation to local projective distortion.

- *Quantity*: The number of detected features should be sufficiently large, meaning that a reasonable number of features are detected even on small objects. In any case, the optimal number of features depends on the application. Ideally, the number of detected features should be controllable by a simple and intuitive to set threshold. The density of features should reflect the information content of the image to provide a compact image representation.

- *Accuracy*: Detected features should be accurately localized, both in image domain position and scale.

- *Efficiency*: Detection of interest points should be computationally efficient in terms of both CPU usage and memory consumption, in order for ensuing processes in the pipeline depicted in Figure 2.1 to have enough resources and adequate response times.

It is worth noticing that some of the above properties are difficult to fulfill simultaneously.

- For example, locality and informativeness. Clearly, as the features become more local, i.e. represented by smaller image patches, it is less informative. This reduction of discriminant information makes the feature much more difficult to match.

- Similarly, *distinctiveness* and *robustness* are competing properties. In order to increase robustness, some information must be singled out as noise. This loss of information reduces feature *distinctiveness.*

It is, therefore, clear that a tradeoff between several properties need to be made in order to have an effective feature detector. The final application is a key factor to decide which properties are more relevant. For example, in applications such as camera calibration [26] or wide baseline image mosaicing, *accuracy* is mandatory. In real time contexts, such as camera pose estimation or SLAM [27], *quantity* and *efficiency* should play a most important role. *Quantity* is particularly useful in scenarios where the number of miss-matches (outliers) can be very high. Anyway, described properties can be reduced to two: quality and efficiency. Quality represents the ability of a feature detector to provide accurate, precise, dense and robust

set of points to the next process in the image analysis pipeline. Efficiency represents how fast and economical in computation resources is the feature detector carrying out the task. Depending on these factors subsequent tasks in the pipeline should or should not apply different mechanisms, filters, estimators, or heuristics in order for the application to succeed, running efficiently and obtaining accurate results.

2.2.1 Multi-Scale Methods

About the concept of scale: When faced with the problem of image analysis, one important consideration is that an image is a *physical observable* that represents the reality as measured by a camera that isable to register some physical measure in a regular discrete finite grid and with a certain dynamic range. Both, the discrete sampling grid and the available dynamic range, implies that there exist a finite scale range at which observations are made. The lower bound of this scale range, often referred to as the *inner scale*, is determined by the sampling characteristics of the acquisition device and it refers to the size of the finest possible feature that can be detected. The *upper scale* bound is limited by the scope of the field of view and refers to the coarsest features that can be observed or captured on the image. Moreover, every imaged structure, represented in digital images, have different sizes and some of them are only meaningful at a certain range of scales. For example, the concept of "tree" is only meaningful at the human vision scale, while speak about the molecules that form the tree are meaningful at a much more finer scale. This finite scale range and multi-scale image nature must be taken into account when performing any image analysis task, with no a priori information about the scale, in order to fix the proper scale(s) at which calculations are meaningful.

Sometimes, calculations performed at a single scale may miss some information. Conversely, image analysis performed using operators tuned to a non optimal scale, may generate false responses, false positive or spurious detections. As [28] pointed out, no single operator can be optimal simultaneously at all scales and a multi-scale approach is necessary, which deals with every relevant scale separately. This happens, for example, when trying to detect some objects or entities whose coarse and fine details span a variable range of scales and all the information is relevant. This is the case of images of the human vasculature [29]. A complex vascular network is comprised of multiple vessels of varying length and diameter and multi-scale approaches are required in most cases for the detection and extraction

of the whole vascular tree. See the work of [29] for a review of multi-scale tubular structure like detection and extraction mechanisms.

Effect of scale Early interest point extraction methods relied on simple Harris' corner detector [20]. However, this corner detector is not able to deal with changes in scale. If there is a change in scale between two different images, for example due to the camera moving away from the object or viceversa, Harris corner extraction would not be able to extract the same set of interest points in both images, hence the feature matching process between those images would be impossible. In contexts where a change in viewpoint can significantly change the relative distance between the camera and the scene being captured, invariant or covariant scale transformation approaches are needed.

Many approaches [30, 31, 32] have been developed in order to tackle the scale sensitivity of Haris' corner detector. Most of such approaches were focused on mechanism to extract points over a range of different scales from an input image and using all these points together to represent the image [33]. This form of image representation is known as multi-scale approach or as scale-space representation. Some other detectors, such as FAST [4], perform interest point extraction very efficiently, however they have poor stability under changes in scale as will be described in Chapter 3, because no scale-space analysis is carried out. Scale-space theory [34, 35, 36] has been instrumental in the success of current applications of local features [21, 30, 37].

Scale-space framework

In order to tackle the problem of scale selection or multi-scale analysis for digital imaging, it is necessary to convert the images into a multi-scale representation and deal with each scale separately, requiring:

- To determine the optimal smoothing filter in order to obtain image representations at different scales.
- Detect the intensity changes at each scale.
- Integrate the information obtained at different scales.

There are two physical considerations to be taken into account to determine an appropriate smoothing filter [28]:

- Filtering should reduce the range of scales over which intensity changes take place. This implies that the frequency spectrum of the filter must be smooth and roughly band-limited with a small variance $\Delta\varpi$.

- Features at each scale should be spatially localized. This implies that the contributions to each point in the filtered image should be obtained from a smoothed average of nearby points, so the filter must be smooth and localized in the spatial domain with a small variance Δx.

The problem is that these two localization requirements, one in the spatial and the other in the frequency domain, are conflicting: It is impossible to concentrate a function both in the spatial and frequency domain. The more concentrated it is in the spatial domain, the more spread it is in the frequency domain. Lindenberg shown in [30] that under some general assumptions on scale invariance, the Gaussian kernel and its derivatives are the only possible smoothing kernel for scale-space analysis. Gaussian convolution satisfies several interesting properties [38], such as n-dimensional separability and, most important, it preserves the existence of any local minima or maxima (zero-crossing point) along the scale-space, while not generating spurious local maxima nor minima in coarser scales that do not to exist in a finer scale of the original signal [31].

Lindenberg [37] introduced the concept of automatic scale selection using a scale-invariant detector which finds maxima in a normalized Laplacian scale-space. Scale-space theory is focused on the basic property that image structures exist at different scales, and the fact that there is no *a priori* knowledge about the scales of relevant image structures, for a given image analysis. Therefore, successful image analysis works at all scales simultaneously and as uniform as possible.

The scale-space representation [21] of a 2D image f can be defined as the solution of a diffusion equation 2.1:

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}(\partial_{xx} + \partial_{yy})\, L, \tag{2.1}$$

given that $L(.;0)$ is equivalent to the original signal f.

This scale-space representation $L(x,y,t)$ of an image represented as $f(x,y)$ can be carried out convolving f with a Gaussian function g [30], as in Equation 2.3, where parameter t denotes the scale being computed. The scale value t represents the variance of the Gaussian function (aka aperture).

Figure 2.2: Original Image (t=0).

$$g(x,y;t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \tag{2.2}$$

$$L(x,y;t) = g(x,y;t) * f(x,y) \tag{2.3}$$

When variance parameter t is equal to 0, Gaussian function g becomes an impulse function, thus the result of convolving f with such a function, is the original f. As the value of parameter t increases, Gaussian function values spread out more along the image space domain. When convolving image f with such different Gaussian functions, the signal f becomes smoother. The smoothness of Gaussian convolved images is directly related with $\sqrt{t}$ where details, with frecuencies similar to $\sqrt{t}$ are removed after convolution, acting as a low-pass filtering. This fact is illustrated in the following images. In Figure 2.2 original function f is shown, where all details are perfectly distingible and sharp. Images depicted in Figure 2.3 are the results of convolving the original function f with different Gaussian functions g, where parameter t is increasing, in discrete values, from $t = 1$ to $t = 16$. As a results, details are becoming increasingly blurred and displaced from their original spatial locations. The set of pictures in Figure 2.3 can be set as the scale-space representation of original input signal f.

Scale-space representation also provides a consistent way of calculating scaled, smoothed image derivatives.

$$L_{x^i y^j}(x,y;t) = (\partial_{x^i y^j} L)(x,y;t), \tag{2.4}$$

Figure 2.3: Smoothed versions of original image for Scale-space representation at t=1,t=4,t=8 and t=16.

Due to commutative property between the derivative operator and the Gaussian function, the scale-space derivatives can be obtained by convolving input signal f with Gaussian derivarive operators, as in Equation 2.5.

$$L_{x^i y^j}(x,y;t) = (\partial_{x^i y^j} L) g(x,y;t) * f(x,y). \tag{2.5}$$

One important property related with the scale-space framework is that the amplitude of spatial derivatives decreases inversely with scale. If a signal is subject to scale-space smoothing, the numerical values of spatial derivatives computed from these smoothed data are expected to decrease [37]. This fact is a direct consequence of one the axioms in [39] that states that convolution with Gaussian kernels does not generate new local-maxima, hence the values of derivatives can not be higher as scale increases. Hence, in order to compute any measure regarding derivatives over several scales, some type of normalization or weighting is needed [21].- For example, many feature detectors such as Harris-Affine [22] computes the Harris cornerness measure (see Chapter 3) as a criteria for finding maxima over scales. Thanks to the use of normalized derivatives, a comparable strength of the cornerness measure is obtained for points detected at different scales, such that a single threshold can be used to reject less significant corners over all scales. This scale adapted detector significantly improves the repeatability under scale transformation changes.

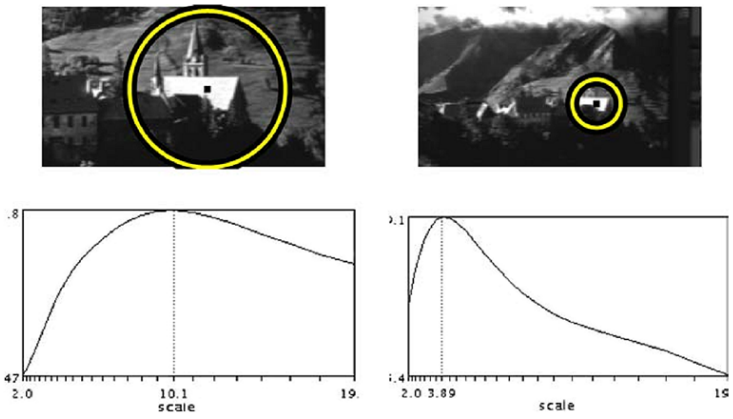

Figure 2.4: Example of an object at different apparent scales (image extracted from [1]).

Scale selection

Scale-selection refers to the identification of the most characteristic scale of a given interest point. Once the multi-scale analysis is computed, a search over scales is carried out in order to find the maxima or minima, setting this value as the characteristic scale of the point. As stated in [21] "*In the absence of other evidence, assume that a scale level, at which some (possible non-linear) combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data.*". In this way, the characteristic scale σ assigned to a given interest point x_i can be interpreted as the apparent scale of the corresponding structure in the real world. Figure 2.4 shows an example where the same scene were captured with two different values of focal length (zoom). The bottom plots are the response of the Laplacian function over the same point (church) over the two images. As can be seen the maximum of both responses correspond with the apparent scale of the structure in the images.

2.3 Feature Descriptors

Feature descriptors mechanisms are intended to represent numerically small image regions surrounding interest points. The shape and size of such regions depends on the nature of the description mechanisms. Usually, those regions are small sets of pixels of regular shape such as rectangles, ellipses or circles. A feature de-

scriptor for an interest point i is a numerical vector f_i which embodies feature data information $f_i = \{f_{i1}, f_{i2}, ..., f_{is}\} \in \mathbb{R}^n$, where s denotes the dimensionality of the descriptor, as shown in Figure 2.5. The numerical vector f_i represents, identify or compress image information surrounding the interest point i. Descriptor vectors are employed to compare interest points by a similarity/dissimilarity function $D(f_1, f_2)$. A good feature vector should be compact, discriminant and robust to geometric and photometric transformations. An evaluation about how state-of-the-art descriptors perform against several transformations is reported in Chapter 4.

The simplest feature detector could be a vector formed by the value, intensity or color, of each pixel surrounding a given interest point i forming an image patch. The computation time needed for the extraction of such descriptors is minimal, however the use of the patch itself as a descriptor is not efficient due to several aspects such as its high dimensionality. Given an image patch of m by n pixels will result in a feature descriptor of mxn dimensions. Performing a task such as comparing that descriptor with all points in a set, by using a distance function d that accounts for all mxn dimensions, would result in a very high time computing task. Such scenarios are typical in a context of 3D reconstruction applications where dense depth maps estimations are needed ([40]). Dense depth maps means estimating relative depth for each pixel in the image, instead of reconstructing a discrete or sparse set of points, such those resulting after an interest point extraction process. This approach is therefore suitable only for short base-line applications where the difference between images to be compared is small. Moreover, using only the image patch by itself in wide base-line scenarios is inefficient due to instability and sensitivity to small changes, reduced distinctiveness capabilities and lack of robustness against geometric or photometric transformations. Therefore, in order to identify, distinguish or match each point from a set of potential matches, better strategies need to be tackled.

According to [41], state-of-the-art feature descriptors can be divided mainly into three different categories: differential descriptors, spatial-frecuency descriptors and distribution based descriptors. Differential descriptors [42] are based on the Taylor series approximation of a function representing input image I (Equation 2.6). The feature descriptor is then formed by several partial derivatives in a local neighborhood of interest point i up to order N. The set of partial derivatives for a scale factor σ up to order N is known as *local jet* [43].

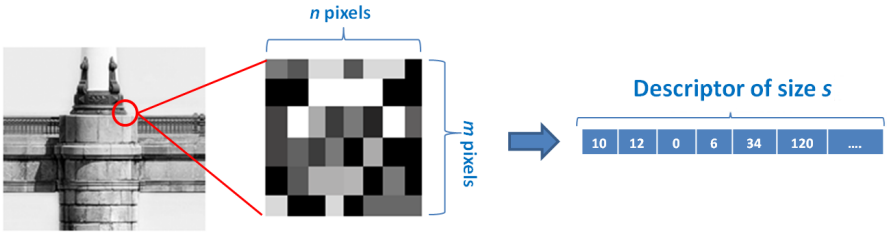

Figure 2.5: Example of Interest Point detection and image patch extraction (Left), and feature descriptor generation (Right).

$$I(x_0+x,y_0+y) = I(x_0,y_0)+x\frac{\partial}{\partial x}I(x_0,y_0)+y\frac{\partial}{\partial y}I(x_0,y_0)+ \\ +\ldots+\sum_{n=1}^{N}x^p y^{N-p}\frac{\partial^N}{\partial x^p \partial y^{N-p}}I(x_0,y_0)+\mathcal{O}(x^N,y^N) \quad (2.6)$$

Spatial-frequency are filters-based descriptors such as Gabor [44], steerable [45] or complex filters. However, more common descriptors are those based on intensity distribution like SIFT [46], SURF [3] or CHOG [47]. These approaches compute statistics about pixel intensity or color values in the neighborhood of interest points, generally in form of histograms. Hence, these descriptors discretize or quantize local image patches in several discrete bins, where pixels sampling for computing such bins is unique of every approach. Some approaches like Local Binary Pattern (LBP) [48] use pixels intensity values, while some others like SIFT or CHOG uses histograms of gradients. By computing image gradient, descriptors are less sensitive to illumination changes because image derivatives are less sensitive to intensity changes. Additionally, by computing local image gradients, some approaches such as FREAK [6] or SIFT estimates a dominant orientation for patch rectification, increasing robustness to in-plane geometric transformation.

2.4 Feature Matching and Homography Estimation

In feature matching, a set potential matches $C = C_a \cup C_b$ over sets of descriptors $C_a = \{f_{1a}, f_{2a}, \ldots, f_{na}\}$, $C_b = \{f_{1b}, f_{2b}, \ldots, f_{mb}\}$ extracted from image a and image b, respectively, are evaluated in order to find correspondences between C_a and C_b.

Correspondences involve measuring the similarity between feature descriptors d_i and d_j associated with their corresponding interest points. For computing such similarities a function or metric $D(f_i, f_j) = \varepsilon$ typically using computationally efficient element-wise measures such as the Minkowsy, Euclidean, chi-squared or Hamming distances, are commonly employed ([22, 49, 50]. These measures assume a linear relationship between descriptors, and it is therefore typical to normalize them in the presence of imaging non-linearities, including illumination changes ([46]).

Nowadays, most state-of-the-art descriptor approaches [50, 51, 52, 6] use binary string descriptors, i.e. a descriptor vector of dimension s is composed as a sequence of s 1's and 0's. These descriptors show great computational performance in several studies [6, 53] as they benefit from using Hamming distance that can be performed very efficiently by using low-level CPU instructions.

Given the set C of potential correspondences and the value of dissimilarity function D, the most common strategy for finding closest matches is by applying brute-force search between all descriptors in C and select the k-nearest neighbors. Some authors [31, 46] propose to use simple heuristics for retaining only "strong" matches, after distances among all descriptors were computed. Lowe proposes to discard correspondences where the distance ratio between the first and second nearest neighbors is higher than t times the minimum distance in the data. These heuristics may improve the set of matches by limiting the influence of potential outliers, hence following processes such as robust model estimation (see Appendix B) can converge faster. Approaches such as [54] uses structures like randomized kd-trees in order to significantly improve performance during computation of k-nearest neighbors, at an additional overhead of computation and memory for preprocessing the data.

It worth noticing that mentioned strategies and distances assume dimensional independence, i.e. non-correlation between descriptor dimensions. In real conditions this assumption does not hold, due to noise or lack of information contained in local small patches, hence dimensions can be somehow redundant. Data analysis such as LDA of PCA are also proposed in some approaches such as PCA-SIFT [55], however, due to time and resources constraints in many application contexts, simple distance functions are preferable.

2.4.1 Projective Geometry

This subsection describes an algorithm to estimate an homography that relates two images, given a set of point matches extracted from respective images. Firstly, we review projective geometry fundamentals to understand the homography estimation process, extensively used along the Thesis. As described in [56] projective transformations may be divided into different "levels" depending on their number of parameters or degrees of freedom.

Isometry

Isometries are R^2 plane transformations that preserve Euclidean distance. An isometry is represented by the following matrix:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha cos\theta & -sin\theta & t_x \\ \alpha sin\theta & cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2.7}$$

The most important isometries are those where $\alpha = 1$, known as Euclidean transformations. They can be rewritten as:

$$\widetilde{x}' = H_e\widetilde{x} = \begin{bmatrix} R & t \\ 0^t & 1 \end{bmatrix}\widetilde{x}, \tag{2.8}$$

where R is a $2x2$ rotation matrix and t represents a translation vector. An Euclidean transformation between planes have three degrees of freedom, one for the rotation, and two for the translation. Given that each match rises two constraints, one for x and one y directions respectively, only two corresponding points are needed in order to estimate the Euclidean transformation between planes.

Similarity

A similarity is an Isometry with an isotropic scaling. If the Isometry is an Euclidean transformation, can be decomposed as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} scos\theta & -ssin\theta & t_x \\ ssin\theta & scos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.9}$$

or:

$$\widetilde{x}' = H_s\widetilde{x} = \begin{bmatrix} sR & t \\ 0^t & 1 \end{bmatrix}\widetilde{x} \tag{2.10}$$

where s represents a scale factor. They have four degrees of freedom, one for rotation, two for translation, and one for scaling, thus only two matches are needed to compute a similarity transformation between two planes.

Affinity

An Affine transformation or affinity is defined as a non-singular linear transformation followed by a translation. An Affinity can be represented as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.11}$$

By rearranging the Equation 2.11:

$$\widetilde{x}' = H_a\widetilde{x} = \begin{bmatrix} A & t \\ 0^t & 1 \end{bmatrix}\widetilde{x} \tag{2.12}$$

Where A is a 2x2 non-singular matrix that defines the affine transformation. These transformations in projective space p^2 have six degrees of freedom, two for the translation, and 4 for the non-singular sub-matrix A. Computing the affinity between two planes requires at least three matches.

A very important implication about affinities is that by the first order Taylor formula, any planar smooth deformation can be approximated around each point by an affine map [49]. As will be described later, the perspective deformation of a plane surface induced by a camera motion is a 2D homography transform, which being smooth, can be locally approximated with an affine transformation. In this way, given a locally smooth surface of a solid object, the apparent deformation in that surface arising from a change in the viewpoint, i.e. camera motion, can be locally modeled by an affinity.

Projectivity

Projectivities are the most general form of transformation in a projective space p^2. They can be expressed as follows:

$$\widetilde{x'} = H_p \widetilde{x} = \begin{bmatrix} A & t \\ v^t & u \end{bmatrix} \widetilde{x} \tag{2.13}$$

Where A is a 2x2 non-singular matrix, t represents the translation vector and v^t represents the vector responsable of the projectivity distortion effect. Therefore, a projectivity accounts for 8 degrees of fredom, thus for computing a projectivity between two planes requires at least four matches.

2.4.2 Homography estimation

A 2D homography is a linear invertible projective transformation that maps points from one plane into another plane. Figure 2.6 illustrates the geometry involved in this process. This transformation is extensively used in computer vision due to its many applications in tracking and 3D reconstruction [57], motion estimation [58], image rectification [59] or camera calibration [26]. Mapping points of a planar surface in the world to its projection in the image plane, or mapping points from a planar surface from one image to another can be modeled with an homography transformation. An homography is also known as a collineation because maps straight lines to straight lines [56]. This transformation does not preserve sizes nor angles but do preserve incidence and cross-ratio [60]. As a projective transformation, an homography H has 8 degrees of freedom up to a scale factor, thus it can be scaled by any factor $k \neq 0$, representing the same transformation.

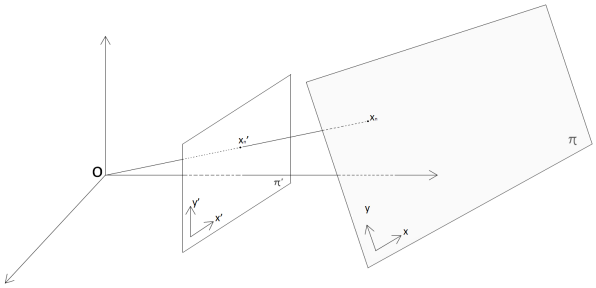

Figure 2.6: Plane to Plane Homography.

Under an homography, we can write the transformation or mapping from points x_1 in plane π_1 to points $\tilde{x}_2$ in plane π_2 as:

$$\widetilde{x_2} = Hx_1\, x_1, \widetilde{x_2} \varepsilon P^2 \tag{2.14}$$

Written element by element, in inhomogeneous coordinates we get:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{2.15}$$

In inhomogeneous coordinates $x'_2 = x_2/z_2$ and $y'_2 = y_2/z_2$:

$$x'_2 = \frac{H_{11}x_1 + H_{12}y_1 + H_{13}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1} \tag{2.16}$$

$$y'_2 = \frac{H_{21}x_1 + H_{22}y_1 + H_{23}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1} \tag{2.17}$$

To estimate an homography between two planes only four correspondences, i.e. four point matches, are needed because each correspondence provides two constraints (x_i, y_i). Without loss of generality we can set $z_1 = 1$, meaning that all points from image 1 come from a real plane in the world located at $z_1 = 1$. We can rearrange equations 2.16 and 2.17:

$$x'_2(H_{31}x_1 + H_{32}y_1 + H_{33}) = H_{11}x_1 + H_{12}y_1 + H_{13} \tag{2.18}$$

$$y'_2(H_{31}x_1 + H_{32}y_1 + H_{33}) = H_{21}x_1 + H_{22}y_1 + H_{23} \tag{2.19}$$

Equations 2.18 and 2.19 can be rewritten as $a_x^T h = 0$ and $a_y^T h = 0$, respectively, where:

$$h = (H_{11}, H_{12}, H_{13}, H_{21}, H_{22}, H_{23}, H_{31}, H_{32}, H_{33})^T \tag{2.20}$$

$$a_x = (-x_1, -y_1, -1, 0, 0, 0, x'_2x_1, x'_2y_1, x'_2)^T \tag{2.21}$$

$$a_y = (0, 0, 0, -x_1, -y_1, -1, y'_2x_1, y'_2y_1, y'_2)^T \tag{2.22}$$

Given more than a minimal set of four corresponding points in both planes, we can solve the coefficients of H, as the solution of a linear system of equations of the form:

$$Ah = 0 \tag{2.23}$$

where:

$$A = \begin{pmatrix} a_{x1}^T \\ a_{y1}^T \\ \vdots \\ a_{xn}^T \\ a_{xn}^t \end{pmatrix} \tag{2.24}$$

Every corresponding point gives two equations to the system, one for each dimension (a_{xn}, a_{yn}). A system of the form of equation 2.23 can be solved by using Singular Value Decomposition (SVD) of A:

$$A = U\Sigma V^T = \sum_{i=1}^{9} \sigma_i u_i v_i^t \tag{2.25}$$

The solution of A in equation 2.25 is given by the column vector v_i corresponding with the smallest singular value σ_i. The column vector v_i correspond with the coefficients of the homography matrix H.

It is worth noticing that, in general, corresponding points are contaminated by noise. This noise can have different sources such as localization error or outliers. Localization error means that a true spatial location of a corresponding point (x_{i1}, y_{i1}) were distorted by an amount of noiseσ_j, thus detected point is $(\tilde{x}_{i1} \pm \sigma_j, \tilde{y}_{i1} \pm \sigma_j)$. Outliers source of error is represented by estimated corresponding points that are not true correspondences. Depending on the amount of these wrong correspondences or outliers, homography transformation estimated directly with Equation 2.25 may be inaccurate. Therefore, a mechanism for robustly estimate homography, even in the presence of noise is desirable. In Appendix Ba review of some of the most extended mechanisms for robust homography estimation are shown.

2.4.3 Camera Model

In this section we give an overview of the mathematical representation of the most common camera model used in computer vision, known as pinhole camera or pinhole model [61]. The pinhole camera is modeled as a closed box with only an small hole in one of the sides, and the projection plane or the photographic sensor, at the opposite side of the box, as shown in 2.7. The very first real cameras worked exactly like this model, without using any lens, only the box and the plate with

photo sensible material and a hole.

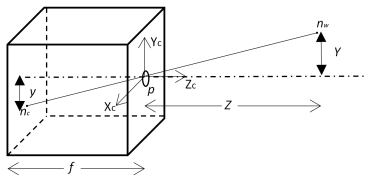

Figure 2.7: Central projection.

In computer graphics, in general, pinhole camera model is adopted as a common standard, except in the case of omnidirectional sensors [57]. In computer vision community, an extended version of the pinhole camera model is employed for modeling the process of projecting world points into the images captured by a camera. Figure 2.8 depicts a general schema of a pinhole camera where world points $P_w(X,Y,Z)$ in world reference frame W are projected to points $P_i(u,v)$ in image plane, located at distance f of the center of projection, given the camera reference frame C.

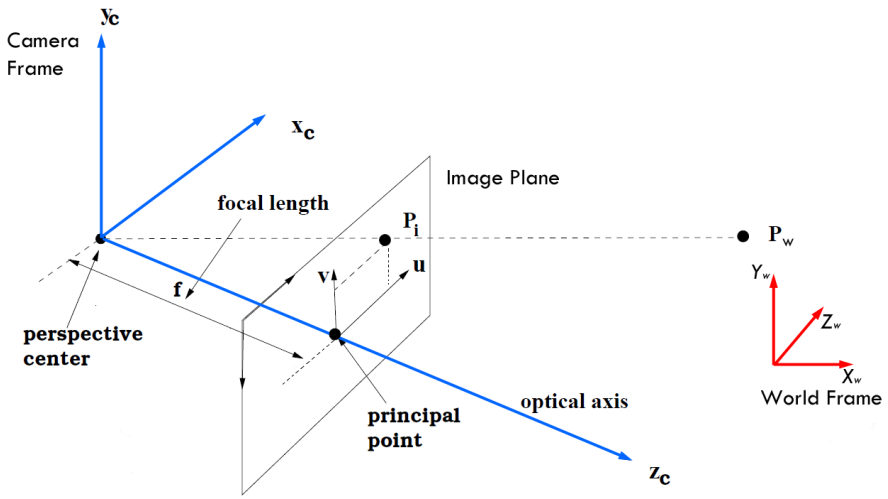

Figure 2.8: Camera Model.

Image formation process is usually represented by equation 2.26 where X_w represents world point, x_i represent world points projected in the image, P is the composition of a translation and a rotation transformation between world and cam-

era coordinate systems, and$K$ describes final transformation from camera reference frame to image reference frame (image sampling).

$$x_i = KPX_w \tag{2.26}$$

$$P = [R|t] \tag{2.27}$$

Matrix K is also known as intrinsic camera parameters and can be represented as:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

where c_x, c_y are the coordinates of the principal point, i.e. the projection of the center camera projection in image, s represents the skew for non rectangular pixels and f_x, f_y are camera focal length represented in pixels. Graphically, projection Equation 2.26 can be represented as follows:

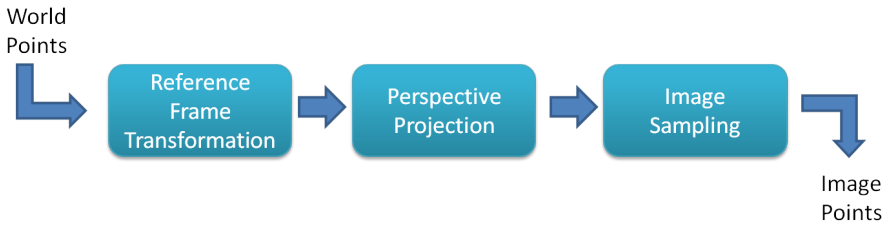

Figure 2.9: Projection pipeline.

2.5 Matching Evaluation Methodology

Chapters 3 and 4 report studies about interest point extraction and feature descriptors mechanisms. Both studies use an evaluation framework implemented during this Thesis, available at www.vicomtech.tv/keypoints, described in detail in Appendix A. The proposed framework has the following I/O specification:

INPUTS:

1. A set of images $I = \{I_1, I_2, \ldots, I_z\}$ captured from a particular scene.

2. A set of bijective functions $S_0 = \{f_{1,2}, f_{1,3}, \ldots, f_{i,j}, \ldots\}$, such that $f_{i,j} : I_i \rightarrow I_j$ establishes the real correspondence between pixels of I_i and I_j, so that mapped pixels are actual projections of the same world point in 3D coordinates.

3. A set of matching algorithms $A = \{A_1, A_2, \ldots, A_w\}$. Algorithm A_m is composed of an interest point extraction algorithm and a feature descriptor technique. A_m produces a set of functions S_m when applied on I, matching keypoints of the images from I. The set of functions S_0 denotes the ground-truth data of I, i.e. the set of mappings corresponding images to world real structures. The set of functions S_m is an approximation to S_0.

OUTPUTS: A set of performance measures of the matching algorithms A_m ($1 \leq m \leq w$). These performance measures grade the quality of A_m against the ground-truth data. Performance measures are repeatability, accuracy and invariance to geometric or photometric transformations.

Many of the test image data sets used for algorithm evaluation in the community working on local feature image characterizaiton are collections of images and the groun truth 2D homography transformations between them [22, 16], allowing to know *a priori* where a point x_i extracted from image a shall be projected in image b, by using equation 2.29,

$$x_{jb} = H_{ab} x_{ia}, \tag{2.29}$$

where H_{ab} is the homography transformation between images a and b. Conversely, points extracted from image b can be projected back to image a applying H_{ab} inverse. Let $\tilde{x}_{jb}$ be the estimated match of x_{ia} provided by a given A_m. Then, H_{ab} can be used to measure the accuracy and repeatability of a point detector algorithm computing the error measure d_{ij} beween the estimated and the ground truth keypoints of a pair of images specified in Equation 2.30:

$$d_{ij} = d(\tilde{x}_{jb}, H_{ab} x_{ia})^2 + d(x_{ia}, H_{ab}^{-1} \tilde{x}_{jb})^2. \tag{2.30}$$

In order to estimate correct matches m_{ab} among all potential matches or correspondences, i.e. pairs of points x_{ia} and x_{jb} extracted from images a and b respectively, we used the overlap error, defined in Equation 2.31 as proposed in [22]. This error measures the correspondence between supporting regions R_a and R_b of

key points x_{ia} and x_{jb}, respectively, under the known geometric transformation. In our case, this transformation is an homography.

$$\varepsilon_s = 1 - \left(\frac{R_a \cap H^T R_b H}{R_a \cup H^T R_b H} \right). \tag{2.31}$$

Figure 2.10 depicts examples of elliptic region overlapping error [2]. In this case, ground truth data is represented by red ellipses and estimated regions around corresponding interest points are blue colored. It is worth noticing that the overlap error comes from differences in region size, position or orientations.

The pair of points x_{ia} and x_{jb} that has lowest error measure d_{ij}, given by equation 2.30, and the lowest overlap error, given by 2.31, is considered as a true match. The overlap error reduces the probability of false positive matches. We calculate the overlap of the ellipses by the software proposed in [62].

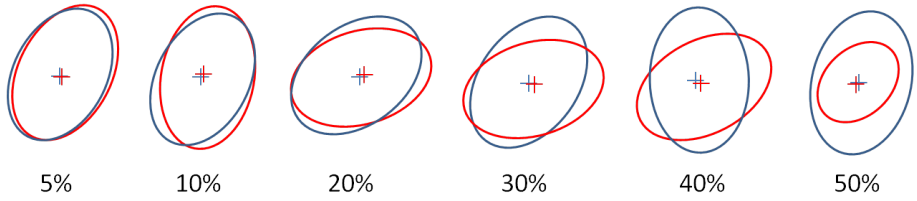

Figure 2.10: Examples of regions overlap error.

The repeatability measure of an interest point detector is defined by Equation 2.32,

$$repeatabilityScore = \frac{numberOfTrueMatches}{numberOfDetectedPoints}, \tag{2.32}$$

where $numberOfTrueMatches$ denotes the number of true matches according to d_{ij} and ε_s, $numberOfDetectedPoints$ denotes the number of detected points in one image such that their projections by the given ground truth homography are inside the other image. Hence, before the computation of the repeatability score, we filter out in both images interest points that do not have correspondence in the other image, taking into account only parts of the scene present in both images. This filter is important because those regions would always give way to false correspondences, thus would degenerate the repeatability score. Figure 2.11 shows a matching example where interest point extracted from image a (left) were filtered out by selecting only those that are inside the image region represented by image b (right).

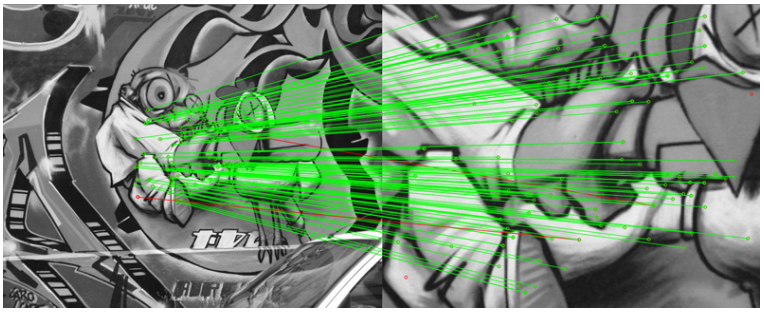

Figure 2.11: Correct (green lines) and wrong (red lines) matches between image *a* (left) and image *b* (right).

Chapter 3

Interest Point Extraction

This Chapter gives an overview of the current state-of-the art of image interest point extraction mechanisms, as well as an evaluation of how these approaches perform against different image transformations. The Chapter is structured as follows: Section 3.1 gives an introduction about the first and most relevant approaches for interest point extraction. Section 3.2 gives an overview about mechanisms for getting feature extractors invariant or robust to some geometric transformations. Section 3.3 briefly describes some of the most relevant approaches of the state-of-the-art interest point extractors. Section 3.4 shows the results of the evaluation about the behavior of several interest point extraction approaches regarding their robustness against geometric and photometric transformations. Finally, Section 3.5 gives a discussion about the results obtained during the evaluation and depicts some conclusions.

3.1 Introduction

The Moravec approach is one of earliest corner detector algorithms [63]. This approach was based on point self-similarity. The author proposes that a corner point should satisfy the condition of being sufficiently disimilar of its surroundings. The author proposes a measure of similarity based on the sum of squared differences (SSD), defined in Equation 3.1:

$$S(x,y) = \sum_{u}\sum_{v}(I(u+x,v+y) - I(u,v)))^2 \tag{3.1}$$

In order to determine if a pixel is a corner point, SSD of several overlapping

patches around the pixels must be computed. If the value of the maximum of these SSDs is over a threshold t and locally maximal, then it is considered as a corner point. In an uniform image region, i.e. a non textured region, every SSD around each pixel will result in low values, meaning all the pixels are very similar. On the other hand, if an edge structure is present in an image region, the patches nearby edge pixels will generate high values of dissimilarity. In addition to the computational cost of calculating several sums of square differences for many patches around every pixel in the image, as the author pointed out, this approach is not isotropic([20]). This anisotropy means that, for example, if an edge is present in an image region, then patches shifted along the orientation of the edge will result in small dissimilarity values, while patches shifted perpendicularly to the edge will generate maximum values. Therefore, depending on the direction of patch shifting some interest points may not be detected.

One of the most important and relevant interest point detectors to date is the Harris corner detector [20] and the mostly used by computer vision community since its publication. Harris approach can be seen as an evolution Moravec's detector [63]. Harris approach improves Moravec's detector by taking into consideration different orientations around the candidate pixel, instead of shifting patches at every 45 degrees.

Suppose we want to compute a weighted Sum of Squared Differences (SSD) between two pacthes, one located in pixel(x,y) in spatial coordinates, and the other one displaced (u,v) pixels:

$$S(x,y) = \sum_u \sum_v w_{u,v}(I(u+x,v+y) - I(u,v)))^2 \tag{3.2}$$

where w represents a smooth circular Gaussian window as defined in Equation 3.3. This weighting window improves Moravec's corner detector, not being so sensitive to image noise, due to local derivative calculations, thus not generating spurious or false corners.

$$w_{u,v} = exp - (u^2 + v^2)/2\sigma^2 \tag{3.3}$$

$I(u+x,v+y)$ can be approximated by Taylor series expansion, using partial derivatives:

$$I(u+x,v+y) \simeq I(u,v) + I_x(u,v)x + I_y(u,v)y \tag{3.4}$$

By substituting 3.4 in 3.2 we have the weighted SSD as:

$$S(x,y) \simeq \sum_u \sum_v w_{u,v}(I_x(u,v)x + I_y(u,v)y)^2 \tag{3.5}$$

$S(x,y)$ can now be re-written in matricial form as Equation 3.7:

$$S(x,y) \simeq (x,y)M(x,y)^t \tag{3.6}$$

$$M = \sum_u \sum_v w_{u,v} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{3.7}$$

The author in [20] suggests that second-moment matrix M approximates the autocorrelation matrix proposed by Moravec [63]. In fact, matrix M, compound of partial derivatives of the image, describes its shape locally. The matrix M is also known as the Harris matrix, and has been extensively used in many practical computer vision applications where discrete features need to be extracted from images. From matrix M the author of [20] proposed to compute an eigen analysis. The two eigenvectors v_1 and v_2 extracted from M defines two main directions, representing the main local changes in the pixels intensity values. As described in [64], the second moment matrix describes the gradient distribution in a region or a neighborhood of a point, hence describes local image curvatures. The corresponding eigenvalues α and β of M represents principal curvatures. Depending on the values of eigenvalues different structures can be described:

- If both curvatures, i.e. α and β values are low, means that there is no much change in the curvature, so the matrix M was computed on a flat image region, or a region without texture. These flat or homogenous image regions show constant pixel intensities and therefore no interest point can be extracted from there.

- If one cuvature is high and the other one is low it can be interpreted that the curvature changes strongly in only one direction. This type of regions is usually referred to as containing an edge structure. Any patch shift in the direction of the edge will cause small changes in the values of Harris matrix M. The corresponding eigenvector of highest eigenvalue represents the dominant orientation of the edge.

- If both α and β eigenvalues are high we can understand that the support-

ing region from where Harris matrix M was computed present two strong changes in curvature. These changes in curvature can be interpreted as a presence of a corner point. Moreover, these regions where both eigenvalues are significantly high can be interpreted as being highly textured. These regions are the most appropriate for extracting interest points because they retain more information about their respective neighborhoods, hence interest points extracted from those areas can be discriminant.

Given the three different combinations of eigenvalues, we can interpret matrix M as a region or local image descriptor by itself. In addition to the local image description through eigen decomposition, the author proposes a measurement C 3.8 that allow to quantify how an image region can contain a corner point:

$$C = \alpha\beta - k(\alpha+\beta)^2 = det(M) - k(Tr(M))^2 \tag{3.8}$$

Where $det(M)$ represents the determinant of Harris matrix M, $Tr(M)$ represent the trace of M and k is a user selectable parameter representing the filter sensitivity. The tuning of this parameter allows the detector to act as a line or corner detector. The smaller the value of k, the more likely the algorithm is to detect strong or sharp corners. For corner detection this parameter is usually set empirically in a range of $[0.04, 0.06]$. As we defined with the eigen analysis of M matrix, regions can also be described by their corresponding cornerness measure C as:

- Flat regions without intensity changed or absent of texture, will have lower responses of C.
- Positive responses of C means the presence of a corner point.
- Negative responses of C means the presence of an edge region.

The computation of C also avoids the explicit computation of eigenvalues, hence is much lighter computationally speaking. Several authors [65, 22] use the Harris cornerness measure as a previous or pre-filter corner detection, as well as post-filtering step for non-maxima local corner suppression.

Figure 3.1 shows 4 different results of computing the Harris cornerness measure over the same image but varying scale parameter σ. As can be seen, as scale parameter increases the resulting cornerness measure image becomes more and more blurry, because of the Gaussian smoothing window w of equation 3.7.

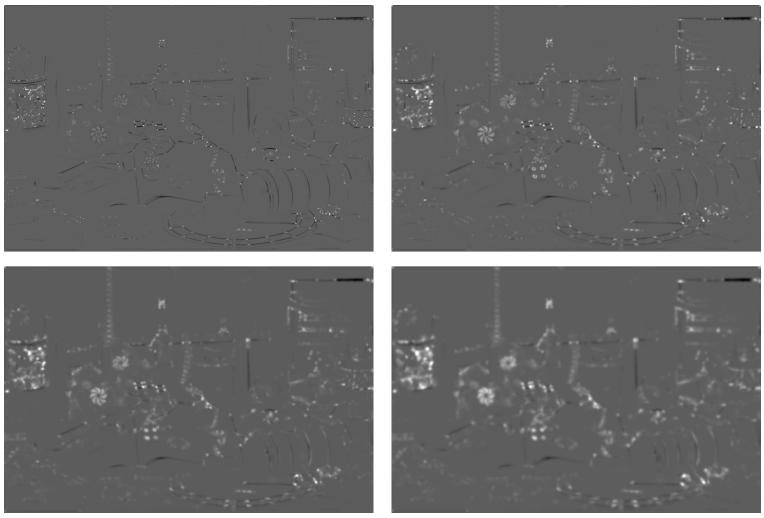

Figure 3.1: Images of Harris Cornerness Measure by varying scale parameter σ

3.2 Towards Viewpoint Invariant Methods

This section gives an overview of different mechanisms adopted by several interest point detectors for getting invariance or robustness to geometric transformations. As described in Chapter 2 there are several geometric transformation involved in image formation, such as isometries, affinities or projectivities.

3.2.1 Scale and affine invariant detectors

The main drawback of Harris detector is that it does not perform multi-scale analysis, thus robustness against changes in scale is poor. Approaches like Harris-Laplace [66] overcomes scale transformation sensibility by extending Harris criteria to multi-scale analysis, resulting in multi-scale Harris cornerness measure 3.9:

$$C(I;\sigma_d,\sigma_i) = det(\Gamma(I;\sigma_d,\sigma_i)) - kTr^2(\Gamma(I;\sigma_d,\sigma_i)) \tag{3.9}$$

where $\Gamma(I;\sigma_d,\sigma_i)$ represents the multi-scale second moment matrix 3.10

$$\Gamma(I;\sigma_d,\sigma_i) = \begin{bmatrix} I_x^2(x_{\sigma_d}) & I_xI_y(x_{\sigma_d}) \\ I_xI_y(x_{\sigma_d}) & I_y^2(x_{\sigma_d}) \end{bmatrix} \tag{3.10}$$

and σ_d,σ_i represents differentiation and integration scales respectively. As

Figure 3.2: Detected Harris corners with different scale parameter σ.

stated in [67], performing scale-space analysis for detecting interest points leads to scale invariance in the sense that interest points are preserved under scale transformations, and the selected scales are transformed covariantly with the amount of scaling ([37]). Hence, the apparent scale values obtained from these interest points can be used afterwards for normalizing local neighborhoods with respect to scaling variations ([68]) which is essential for the scale invariant properties, and also for computing normalized local gradient distributions ([46]).

Some approaches like Harris-Affine [22] and Hessian-Affine [66] extract an elliptical region around each point, in order to be robust against affine geometric transformations. The shape of the elliptical region is defined by the directions of the computed eigen vector from the autocorrelation matrix. The shape of these regions can be modeled by an affine transformation, and thus can be rectified back to their canonical circular shape ([2]), as shown in Figure 3.3:

Hessian-Affine operates very similarly to Harris-Affine but using Hessian matrix, i.e. second order derivatives, as defined by Equation 3.11, for detection points in the scale-space.

$$H(x,\sigma_d) = \begin{bmatrix} I_{xx}(x_{\sigma_d}) & I_{xy}(x_{\sigma_d}) \\ I_{xy}(x_{\sigma_d}) & I_{yy}(x_{\sigma_d}) \end{bmatrix} \tag{3.11}$$

As described in [2] the second order derivatives give strong responses on blobs

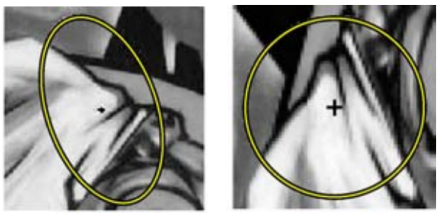

Figure 3.3: (Left) Elliptical region extraction, (Right) Region rectified to canonical shape (Image extracted from [2]).

and ridges. Regions extracted using 3.11 are very similar to those detected by Laplacian operator used in [37, 31], but using a function based on the determinant of the Hessian matrix. This function penalizes long structures for which the second order derivatives in one particular orientation is small, i.e. penalizing line-like structures as defined by [69], hence avoiding the generation of unstable, non-informative interest points, extracted from those structures.

In [2] a detailed evaluation of affine covariant region detectors such as Harris-affine, Hessian-affine, EBR [70] or MSER [71] can be found. As a general conclusion from that study is that MSER performs better than the rest of detector, in most of the cases except for image blurring.

In addition to robustness against affine and scale transformations, interest point extractors deal with in-plane rotation transformation. Usually ([46, 3, 6]), the orientation invariance is obtained by rotating the patch towards the direction of the dominant gradient orientation. The most common approach was proposed by Lowe [31]. Lowe proposes to compute a discretized histogram of gradient orientations, normalized with estimated apparent scale. The peak of that histogram, i.e. the most frequent orientation in the local neighborhood of the interest point is selected as dominant orientation.

3.3 Interest Point Detectors

This section reviews some of the most relevant state-of-art interest point extraction approaches.

SIFT

(Scale Invariant Feature Transformation) descriptor [31] is one of the most successful approaches for feature or interest point extractor and description. Interest point detection is based on the convolution of images with difference of Gaussians (DoG) operator $\eta = (g_\sigma - g_{\sigma'})$. Difference of Gaussians can be seen as an approximation to the Laplacian of Gaussian, as stated in [72]. Difference of Gaussian operator is computed by smoothing each image of a given octave with Gaussian kernels of different size σ, and then subtracting them, as depicted in Figure 3.4, where the first two octaves of input image are shown.

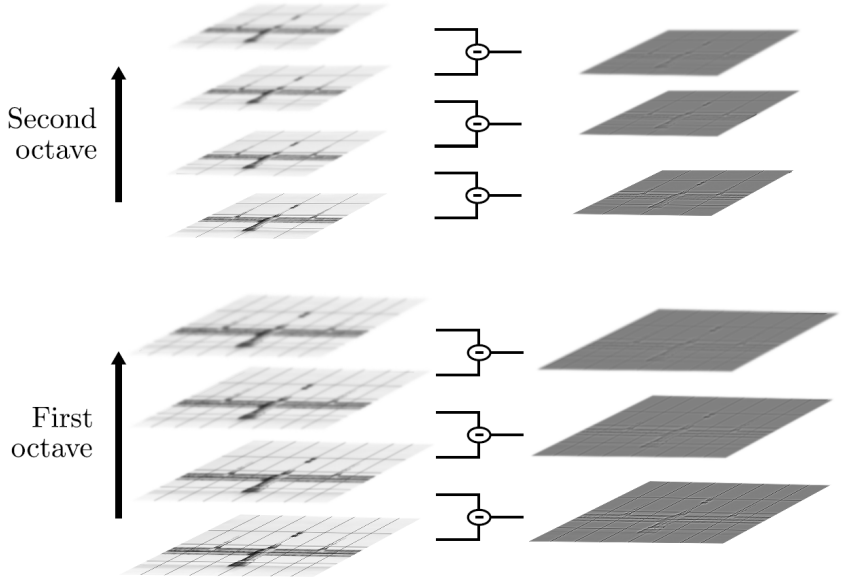

Figure 3.4: Scale-space computation by using DoG.

In this way, images are arranged in a pyramidal representation, where every level (octave) of the pyramid represents a down sampled and smoothed version of the image in the previous level. As seen in the description of scale-space Framework([33]) in Chapter 2, a 2D Gaussian function is separable, so it is able to apply a 1D Gaussian kernel (Equation 3.12) in every dimension separately. This convolution separation notably reduces computational costs, thus improves the overall performance of the detector.

$$g(x,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/2\sigma^2} \tag{3.12}$$

Interest point detection is thus detected by performing a search for scale-space extrema within the difference of Gaussians computed image pyramid. In order to remove or alleviate the strong responses of DoG operator along the edges, SIFT performs local suppression by removing those scale-space extrema that respond lower to a given threshold τ, using a ratio between the eigenvalues of the Hessian matrix as defined in Equation 3.13:

$$\frac{det(H)}{Trace(H)^2} = \frac{L_{xx}L_{yy} - (L_{xy})^2}{(L_{xx} + L_{yy})^2} \geq \tau \tag{3.13}$$

Characteristic scale σ of a given SIFT interest point is set as the maxima or minima found over the difference of Gaussians pyramids.

SURF

(Speed Up Robust Feature) [3]extractor follows a similar approach to SIFT, addressing explicitly the problem of reducing computation cost. SURF searches for local maxima of the Hessian determinant in the scale-space. SURF calculates Hessian determinants by using a discrete approximation of the Gaussian second order partial derivatives$D_{xx}, D_{yy,}, D_{xy}$, as defined in Equation:

$$det(ApproximatedHessian) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{3.14}$$

where w is a normalization factor set to 0.9 according with the authors [3]. D_{xx}, D_{yy} and D_{xy} approximations are obtained by using box-filtering, as depicted in Figure 3.5.

SURF generates box-filter kernels very efficiently by employing integral image representation ([73]). In addition, SURF contrary to SIFT, scale-space computation is carried out by increasing the size of box-filter kernels while input image remains at full resolution instead of sub-sampling Gaussian smoothed versions of the input image. The dilation of box-filters are again very efficiently computed by using integral images. The speed-up of SURF compared with SIFT comes from the intensive and optimal use of integral images for computing box-filters. In fact, the computational cost of applying the box filter is independent of the size of the filter because of the integral image representation. For characteristic scale selec-

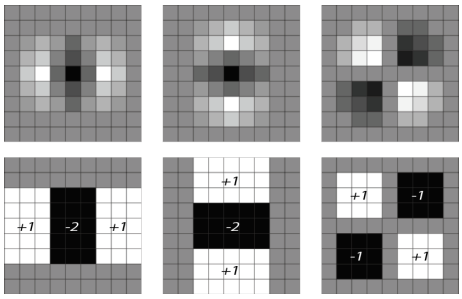

Figure 3.5: (Top) L_{xx}, L_{yy}, L_{xy} Second order Gaussian Derivatives, (Bottom) D_{xx}, D_{yy}, D_{xy} box-filter Gaussian approximation (image adapted from [3]).

tion, the authors propose to do a search for a maximum in scale-space computed using box-filtering, where box-filter of size $9x9$ is considered as the initial scale layer, corresponding with Gaussian derivatives computed at scale $\sigma = 1.2$, and interpolate between adjacent octaves using [74].

STAR(Censure)

This point extractor is also known as Censure (Center Surround Extrema) [24]. This approach approximates the Laplacian not using DoG operator as SIFT does, but using bi-level center-surround filters of different shapes such as boxes, octagons, or hexagons. The computation of these filters in combination with integral images allows the detection of interest points in scale-space much faster than SIFT. In our evaluations we used bi-level star shaped filter as proposed and implemented in [75].

Censure detector is very similar in essence than SIFT or SURF, but the authors propose to improve location accuracy by performing scale-space search at full resolution. As described in 2, when computing image scale-space, achieve accuracy in both frequency and spatial domain is difficult. Minimum or maximum extrema localized at coarser scales are poorly localized due to abroad variance of Gaussian kernels. Instead, Censure interest points are extracted as the extrema of the center-surround filters computed over multiples scales, using original image resolution at every scale. In this way, spatial localization are much more accurate and hence applications such as camera calibration or visual odometry can improve their performance. Censure detector computational efficiency is obtained by implementing bi-level filtering using integral images ([73]).

FAST

FAST (Features from Accelerated Segment Test) originally proposed in [76] follows a different approach than SIFT or SURF detectors. FAST uses supervised classification to label pixels as members of the class "interest point" or the class "background", by examining the values of pixels surrounding a candidate point in a circular path, as illustrated in Figure 3.6. A feature is detected at pixel p if the intensities of at least n contiguous pixel of a surrounding circle of j pixels are all below or above the intensity of p by some threshold t. The final set of feature points is determined after applying a non-maximum suppression ([77]) step to previously computed potential interest points. This detector follows a similar previous approach proposed in [65] where a corner response function (CRF) is proposed by evaluating the intensity of opposite pixels disposed in a circle around a pixel p. Original FAST approach does not perform scale-space representation. Moreover, FAST by itself does not produce a measure of cornerness, hence the authors propose to apply Harris cornerness measure (see Eq. 3.8) for selecting the N higher points.

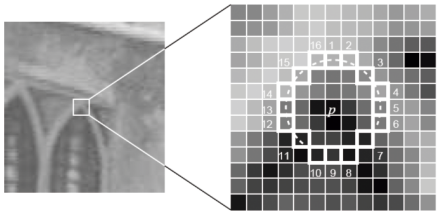

Figure 3.6: FAST Local Detector ([4])

FAST approach can be seen as a morphological feature detector, meaning that it is based on morphological operations rather than convolution-based operations. In this way, it does not require second order derivative computations and thus no prior denoising. This difference accounts for a large part in its efficiency gain. Another version of FAST was also proposed by the same authors in [4] where original performance was again improved by optimizing computation using better low-level CPU instructions and assembly code.

ORB

ORB is the acronym of Oriented FAST and rotated BRIEF. This algorithm proposed in [52] is a modified version of FAST detector for computing orientation during detection step, and an efficient computation of BRIEF based approach for generating descriptors. This approach tries to merge the rotation and scale invariance of SIFT and the computational efficiency of FAST detector. Both FAST([4]) and BRIEF ([78]) were designed with performance being the most important factor. However, as it will be shown during evaluation in Section 3.4, and later in Chapter 4, both are sensitive to scale and rotation geometric transformation. ORB tries to overcome both limitations. ORB improves rotation invariance by computing an orientation vector based on intensity centroid method defined in [79]. This method proposes form a vector from the center of the patch O to the centroid point C computed by using central moments ([80]), defined by Equation 3.15:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{3.15}$$

where $I(x,y)$ represents intensity of pixels at position (x,y) of image I. The centroid C is determined by:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{3.16}$$

Join points O and C we have the vector $\overrightarrow{OC}$, representing dominant orientation, and angle θ can be computed as $\theta = atan2(m_{10}, m_{01})$. Moment calculations are performed within a circular region of radius r centered on point (x,y). Finally, ORB performs a gaussian smoothing of image patch before BRIEF descriptor computation, in order to increase robustness against digital noise. BRIEF descriptor is described later in Chapter 4.

MSER

(Maximally Stable Extremal Regions) proposed in [71] is an approach based on the detection of blob like structures, similarly to SIFT but instead of using differential computations MSER uses intensity segmentation. MSER detects blobs by using local luminance extrema, obtained by iteratively applying watershed based segmentation. A region R_i is considered stable and therefore considered a potential interest region, i.e. a feature, if for all of its n joined connected components

$R_1, ..., R_n$,obtained after n watershed segmentation steps, reaches a local minimum in the function $q_i = \frac{\|R_{i+\alpha} - R_{i-\alpha}\|}{\|R_i\|}$, where α is a user defined parameter and the operator $\|.\|$ represents the cardinality of the blob measured in pixels. MSER detector is by definition covariant to affine transformations ([71]). This detector estimates elliptic regions by estimating the most similar elliptic regions enclosing a given arbitrary shaped region. MSER uses ellipse estimation for setting a dominant orientation to every interest point. More precisely, MSER sets a dominant orientation as the longest axis of the estimated ellipse, corresponding with the eigenvector associated with the highest eigenvalue.

AGAST(Brisk)

Proposed in [50], this detector implements a modification of the FAST detector proposed in [81] which improves the original FAST score computation by changing the original classifier by binary decision tree. AGAST detector tries to overcome the limitations of FAST detector regarding scale robustness by computing FAST detection over several octaves in a scale-space representation. This scale-space representation consists of n octaves(levels) C_i and $n-1$ intra-octaves d_i, located between octaves C_i and C_{i+1}. As SIFT detector, octaves are formed progressively by half-sampling the original input image corresponding with octave C_0. The first intra-octave is obtained by down sampling original image by a fixed scale of 1.5, thus subsequent intra-octaves are half-sampled according with $d_i = 2^i * 1.5$. Once scale-space is computed, FAST score is computed in every octave and intra-octave separately by using the same threshold T to identify potential regions of interest. Then a non-maxima suppression to every potential interest region is computed by comparing it with its 8 local-neighbors in the same octave, as well as with the octave above and below. For selecting the apparent or characteristic scale for a given interest detected in octave C_i, the authors propose to use interpolation between the scales corresponding with the intra-octave d_{i-1}, octave C_i and intra-octave d_{i+1}. This interpolation is carried out by fitting these three scales in a 1D parabola along the scale axis.

KAZE

Introduced in [5]this detector proposes a novel multi-scale interest point detection, where common linear scale decomposition with Gaussian filtering, used in several approaches such as SIFT, SURF, BRISK, or pyramidal Harris, is replaced by a

non-linear diffusion filtering. This type of filtering smooths images similarly to Gaussians, but better preserving region boundaries, as shown in Figure 3.7.

Figure 3.7: (Top) Linear Gaussian scale-space, (bottom) non-linear difussion scale-space (image extracted from [5]).

As stated in [82], the well known Gaussian scale-space is just one instance of the linear diffusion and other linear scale-space do exist, such as [83]. As stated in Chapter 2 obtaining proper localization accuracy in coarser levels is difficult due to broad values of scales σ that widely smooths the image. The authors propose to improve localization accuracy by performing scale-space computation using non-linear diffusion filtering efficiently by using Additive Operator Splitting ([84]).

The authors of KAZE propose a scheme of scale-space levels very similar to SIFT, consisting of several octaves but where each sub-level of corresponding octave is computed using non-linear diffusion technique. For interest point detection, the authors propose to do a search for maximum within the scale-space of normalized Hessian ([37]).

3.4 Evaluation

This section shows the results obtained in different tests we carried out following the experimental framework described in Appendix A. This framework allows estimating several performance measures of interest point detectors such as repeatability score, detection accuracy, and computation time. We evaluated the behavior of interest point detectors described in previous section as implemented using OpenCV Library version 2.4 ([75]), running entirely in CPU (not using the computer's GPU). We set all specific detectors' parameters to their default values, as suggested by their authors.

In the current detectors evaluation we use several sets of images showing both geometric and photometric transformations. First, we use a data set proposed in [2], composed of three different sets of 6 images each, showing rotation, scaling and perspective transformations as displayed in Figure 3.8.

Figure 3.8: Sample images from Graffiti (left), Boat (centre), and Brick (right) data sets from [2]

In addition to these datasets, we use the set of images proposed in [16] and the synthetic image generator described in Appendix A.

3.4.1 Detection density evaluation

Detection density test compares the number of interest points that every detector is able to extract. Depending on the specificities of each algorithm, the number of extracted points may vary significantly, even if they are applied on the same image. Furthermore, depending on the image spatial frequencies, the number of detected points can be different. We have used three different set of images with different contents and therefore different textures and spatial frequencies. For example, images of the Graffiti data set exhibit well defined smooth and homogeneous regions, while images of the Brick data set show highly frequent repeatable patterns. All tests were carried out limiting the maximum number of detections to 6000.

	SIFT	SURF	AGAST	ORB	FAST	HARRIS	MSER	STAR	KAZE
Graffiti	1108	2505	1080	6000	5759	699	555	874	5209
Boat	1451	4088	3691	6000	4850	3426	192	1923	5209
Brick	1821	5371	1511	6000	5458	5571	1447	1461	5209

Table 3.1: Density results.

Table 3.1 contains the density detection results of all tested detectors over the Graffiti, Boat and Brick data sets. ORB and FAST detectors are the ones that get more dense clouds of interest points, followed by KAZE. ORB detector seems to reach always the maximum number of detections allowed, in this case 6000, independently on the image content. This tends to generate very close detection of points or clusters, what may have a negative impact in some applications such as camera tracking. Similarly high number of detections is obtained with KAZE detector, however KAZE detected points are more uniformly distributed over the image domain than ORB or FAST detected points. It's worth mentioning that MSER approach generates the lower number of detections. A very small number of detections can limit the usefulness of the detectors in some applications such as SLAM or 3D reconstruction, where dense detections are preferable. Finally, we remind the reader that the number of points detected means is not the only measure for a successful detector, but also how discriminative and repeatable they are against some transformations such as geometric or photometric ones.

3.4.2 Evaluation of robustness against geometric Transformations

In this section we describe the results evaluating the robustness against rotation and scale similarity transformations, affinity transformation, and finally perspective transformations.

Rotation similarity transformation: In this test we evaluated how different approaches are robust against image rotation. We used the first image of Graffiti data set along with the tool described in [16] to generate rotated images by applying different angles of in-plane rotation similarity, starting from 0º (same image), to 360º degrees, in steps of 7.2 degrees.

Results depicted in Figure 3.9 show that some detectors such as SIFT or MSER are almost insensitive to in-plane rotation transformation obtaining almost constant value 70% of repeatability along the whole transformation range. SIFT operator uses DoG operator for approximation to Laplacian of Gaussian operator that is rotationally covariant ([30]), hence SIFT DoG operator is rotationally less sensitive to other approaches like box-filtering([3]). ORB is also insensitive to transformation but its repeatability values are lower than SIFT and MSER, around 55%. Some detectors such as KAZE and specially SURF shows high sensitivity to specific in-plane rotation values like 45º, 90, 135º, 180º, 225º, or 270º degrees. In the case of SURF we attribute this sensitivity to discretization effects induced by the use of

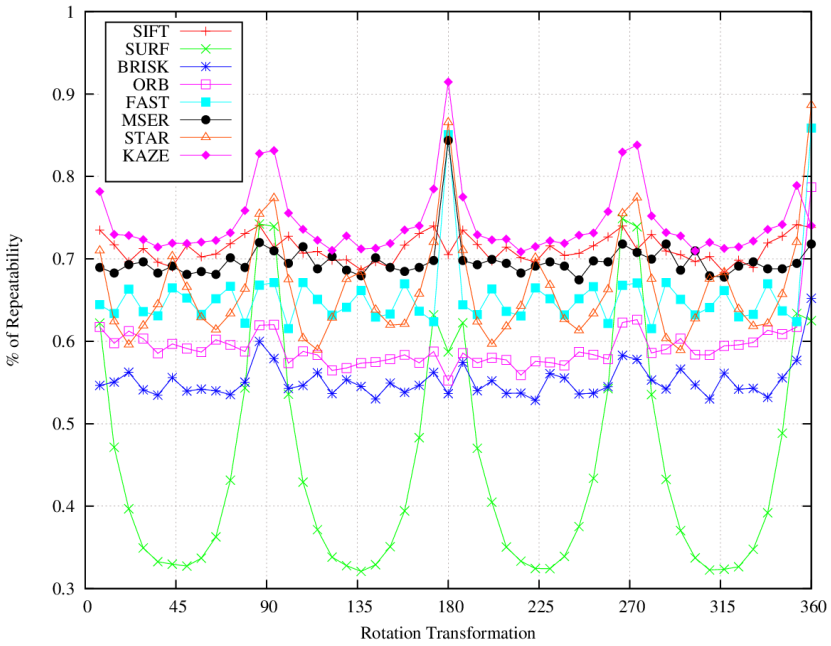

Figure 3.9: Repeatability results of rotation transformation.

box filters as approximations of the LoG operators. It's worth noticing that FAST detector, despite its simplicity, obtains good results along the transformation range, generating the best results, together with STAR and KAZE detectors, when image rotation is exactly 180º (upside down image). Apart from FAST, the remaining detectors estimate a dominant orientation in the supporting region around every interest point, allowing to rotate it back or to rectify it, in order to obtain robustness to rotation transformation when computing their respective descriptors. FAST detector only evaluates some pixels (from 9 to 16) around an interest point without the need of dominant rotation estimation and correction, thus being computationally optimal.

Scale similarity transformation: We use again the first image of graffiti data set to generate new iso-tropically scaled views of that image. More precisely, we generated 50 images with a range of scale factors from 0.04 to 2.4. Scale values below 1 mean augmentation of image structures, while values above 1 mean reduction. Figure 3.10 plots the repeatability results in this experiment. Clearly the SIFT detector shows superior results when the value of scale factor transformation is extreme, being robust even with scale factors higher than two. Also, it is

worth mentioning that MSER and BRISK obtain good results, performing better than SIFT for scale factors lower than 1, i.e. when images are augmented versions of the original one, or the camera is moving closer to the scene, so the objects seem to increase their apparent size. Finally, it's worth mentioning that FAST detector is not invariant to scale transformations, given results of repeatability close to 0 when scale factors are out of the range 0.65-1.25.

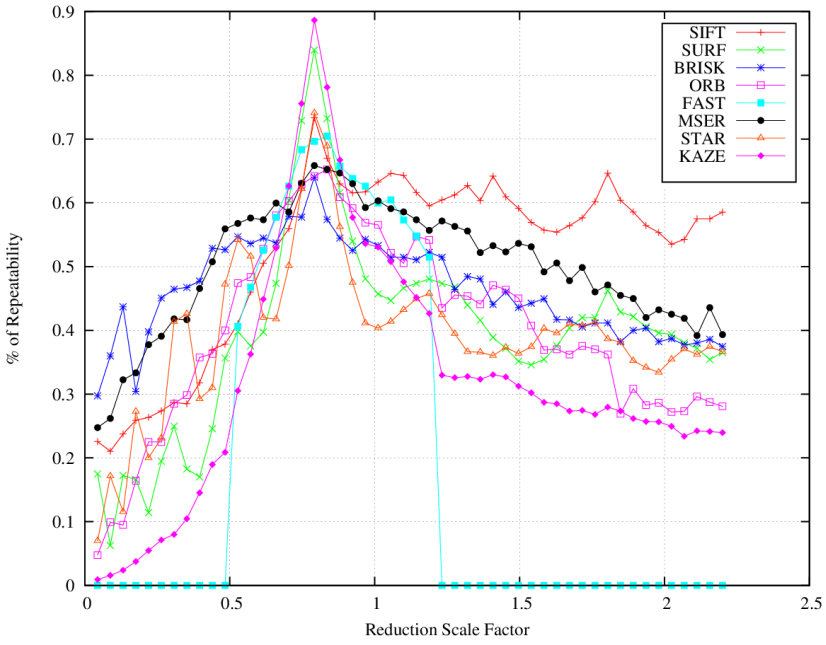

Figure 3.10: Repeatability results of scale transformation.

Affine transformation: Besides the most general transformations (projectivities), affine transformations are the most interesting transformations modeling camera viewpoint change. A perspective transformation of a smooth surface can be locally approximated by an affine transformation. They are very useful in several contexts such as SLAM or camera tracking. In this test, we use 50 generated images by applying affine deformation (non-uniform scaling and skew) in x direction from image 0 to image 25 and then in y direction from image 26 (most distorted image) to 50 (original image).

Results shown in Figure 3.11 demonstrate that none of the detectors but MSER is fully invariant to affine transformation but all of them perform robustly. However, the number of MSER regions in an image is, in general, very limited and

very dependent on the content. KAZE detector obtains the best results, getting 85% on average along the transformation range. Anisotropic diffusion employed by KAZE seems to be more appropriate than usual Gaussian smoothing employed by many of scale invariant approaches, when dealing with affine deformation due to its non-linearity.

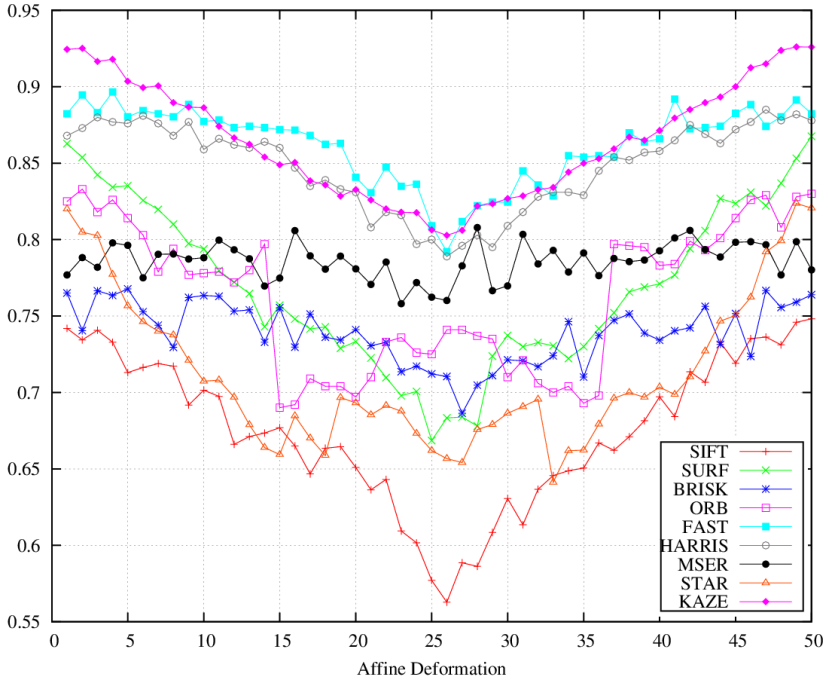

Figure 3.11: Repeatability results of affine transformation.

Perspective transformation: In the following test we evaluate robustness against homography projective transformation. The projective transformation between any two images of the same planar structure in space can be described by a homography transformation. Homography transformation estimation is widely used by computer vision community in many applications such as image rectification, image registration, camera calibration or camera pose estimation. In the current test, we use the first four of the Graffiti data set for measuring the repeatability score between image 1 (reference) and the other three images. We left out the last two images of the data set because perspective distortion between the reference image and these images is too severe. This distortion limits the applicability of every

detector because they are unable to extract a significantly high number of stable interest points, thus generating that repeatability score were not very reliable.

Results in Figure 3.12 show that none of the tested detectors are truly invariant to perspective transformation. BRISK and ORB got the best results, followed by KAZE. In general, the repeatability scores in this test are lower than in the rest of tests, meaning that detectors are very sensitive to perspective transformations and distortions. All current approaches propose to extract interest point to be invariant to scale and rotation transformations. In addition some other approaches such as MSER[71] or Harris-Affine[22] are proposed to be also affine invariant. However, non of them is truly invariant to perspective or projective transformation because projectivities are too general thus having too many degrees of freedom. Moreover, when distortion generated by perspective transformation is not very high, this transformation can be locally approximated by an affinity. Therefore, affine invariant detectors such as MSER can be robust against small perspective transformations.

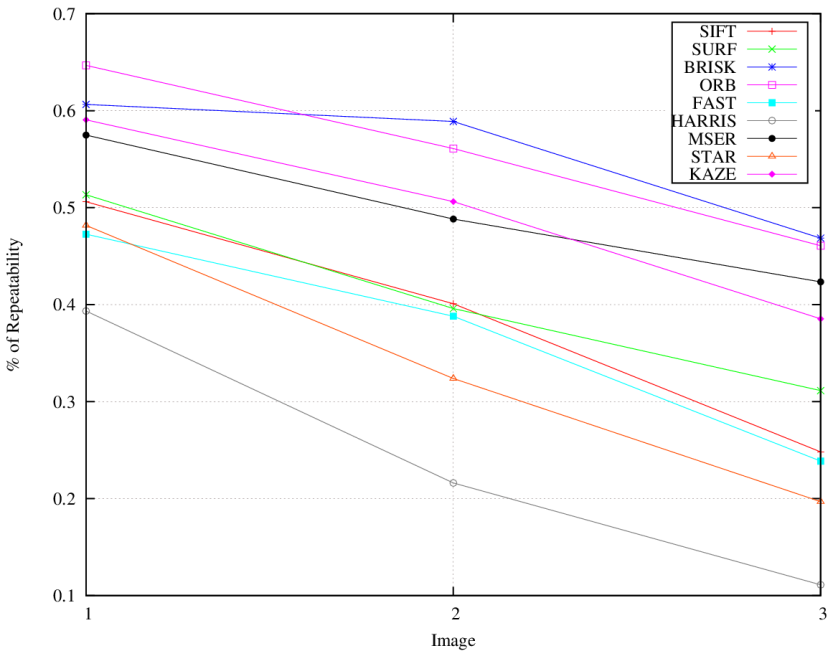

Figure 3.12: Repeatability results of projective transformation.

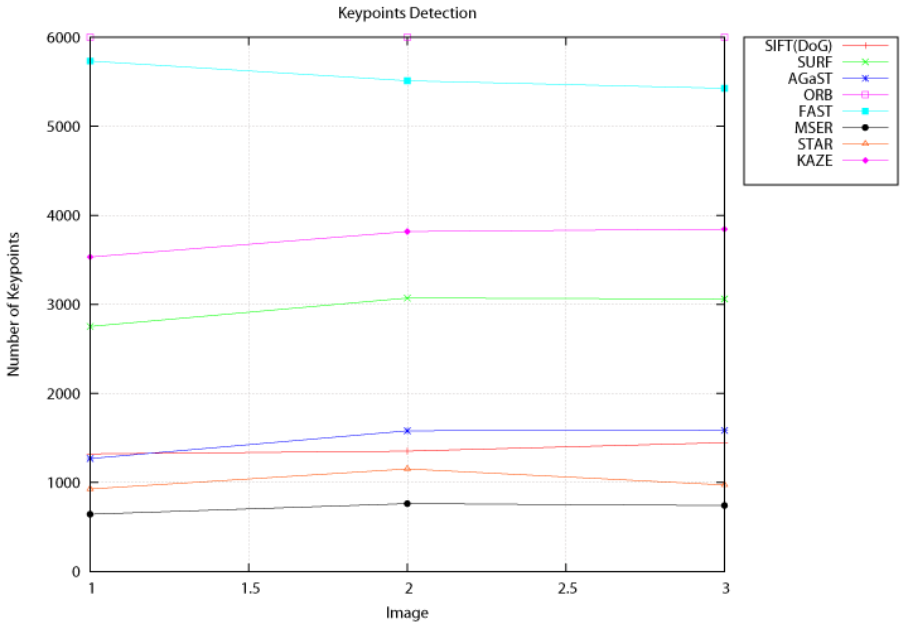

Figure 3.13: Number of interest points detected in Graffiti data set.

3.4.3 Evaluation of robustness against photometric Transformations

In addition to geometric transformations, we carry out an evaluation of robustness of described interest point detectors against photometric transformations.

Exposure photometric transformation: This test evaluates the robustness of the detectors against variations of light intensity. We use the data set proposed in [16] consisting of 15 images captured under controlled light conditions. The light is modified from a correct scene exposition to around 4.5 f-stops less of exposure, in steps of 1/3 f-stop, Sample images are shown in Figure 3.14 .

Figure 3.14: Sample images of photometric exposure transformation dataset.

Results obtained with this data set suggest that light intensity variations affect

all detector. As light decreases, the repeatability scores of every detector also decreases. The most stable results are obtained by FAST, and SURF, followed by MSER. As shown in Figure 3.15, as light intensity decreases the number of detections of every detector also decreases but in the case of FAST. When light intensity is reduced to around 3 f-stops the number of detections of every detector is reduced to a number lower than 50% of the total number of detections with correct exposure. As described previously, FAST detector is based on the computation of relative pixel intensity differences. Clearly, this approach is robust and invariant to linear intensity light variations.

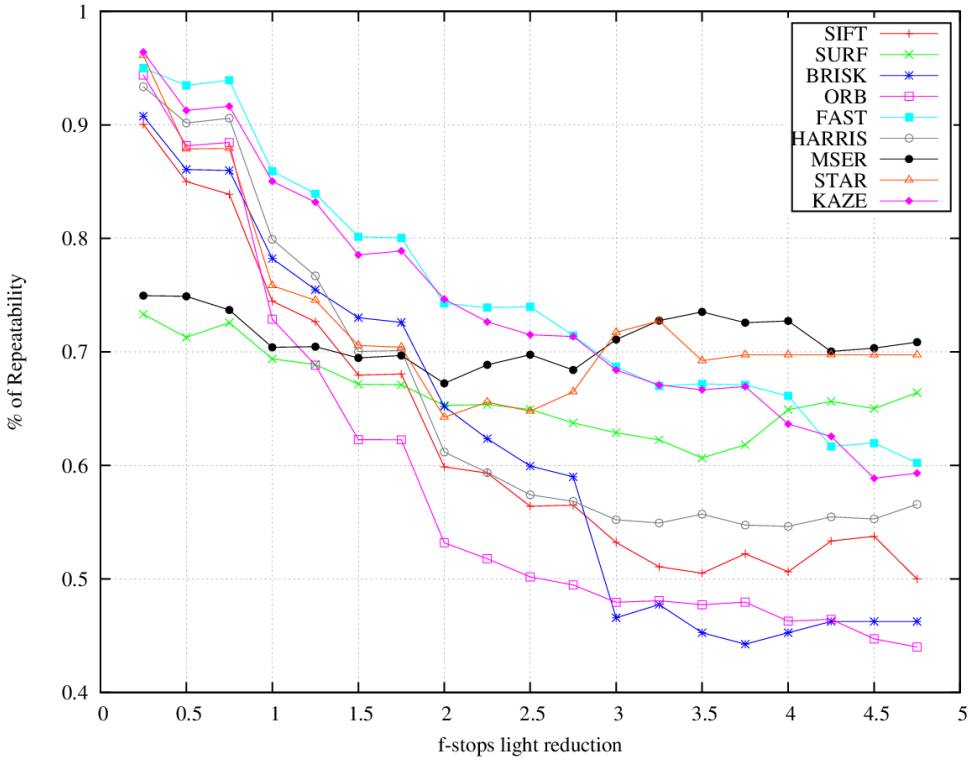

Figure 3.15: Exposure data set Repeatability score.

Blurring photometric transformation: This test measures the robustness against image blurring. This photometric transformation may occur due to fast camera movements or by a change on the lens focus point. We used a data set consisting of 15 real images where lens focus point was modified from a perfectly in focus image to a completely out of focus image, as shown in Figure 3.17.

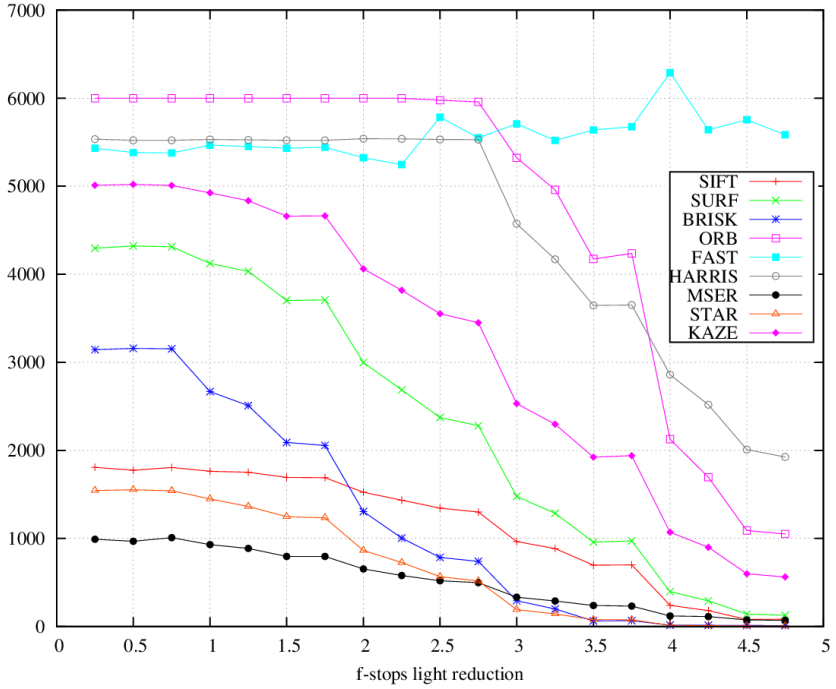

Figure 3.16: Number of interest points detected.

Results of repeatability and detection density are depicted in Figure 13 showing that: as image blurring increases the number of detection decreases, in some cases like in BRISK this reduction is very severe. Every detector uses some type of image blurring, usually through Gaussian functions, prior to interest point detection in either single or multi-scale approach. The most stable detectors are BRISK, ORB and SURF. It is worth noticing that some approaches such as FAST, SIFT and Harris are very sensitive to this type of transformations, obtaining the worst results of the evaluation.

MSER also suffer from image blurring and in addition it is the approach that extracts less interest points. Watershed segmentation approach used in MSER tends to extract wide, homogenous well delimited regions, hence as the amount of blur increases the region boundaries diminishes those regions are poorly estimated.

Noise photometric transformation: We also evaluate the robustness of interest point detector algorithms against image noise. In the current evaluation we are dealing with approaches working on image intensity only, thus ignoring color information. In this way, we use a data set composed of 15 images that progressively

Figure 3.17: Sample images of photometric focus transformation data set.

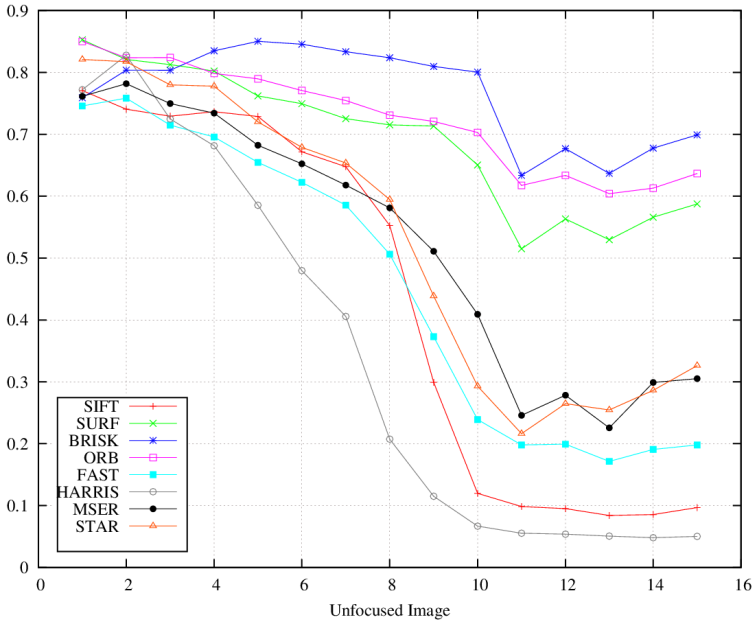

Figure 3.18: Repeatability results of blurring photometric transformation.

contaminates input image with luminance additive Gaussian distributed noise as shown in Figure 3.20.

Contrary to the previous experiment, the number of detections of every approach increases as image noise increases. This is due to the addition of spurious data that generates new responses while computing image derivatives. This spurious data cause new false responses (interest points) during the search of local maxima or minima over different scales. Despite of the number of detections, clearly these false interest points are not stable, thus repeatability scores are continuously decreasing as image noise quantity increases, as depicted in Figure 3.21. The most stable detector against image noise is BRISK followed by ORB, but all of them

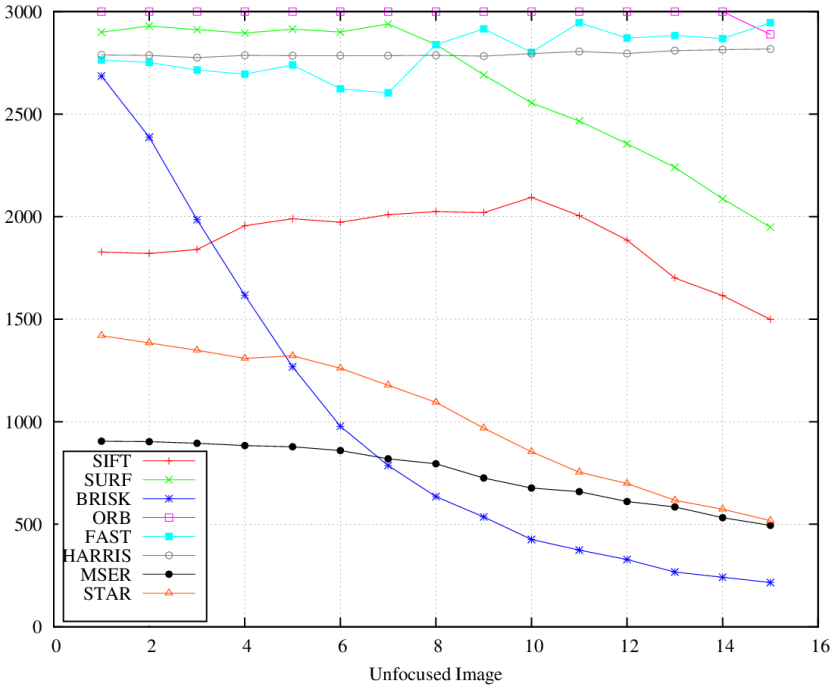

Figure 3.19: Number of interest points detected.

follow the same trend. None of the tested detectors is fully robust to luminance image noise.

3.5 Discussion and Conclusions

Results in the evaluation section confirm that there is not a single interest point detector that clearly outperforms the rest of approaches in all situations. There are some tests where a particular detector performs better than the others, but downgrades in other tests. In general, the best approach is the one that better fits into the specific application requirements. For example, SURF approach performs similar to SIFT, generating more dense interest points, being computationally faster, but suffering from rotation sensitivity, showing irregular results along rotation transformation range. If our particular application does not expect severe camera or object rotation, SURF can be a perfect alternative to SIFT. Otherwise, if rotations are expected, ORB would be a much better option.

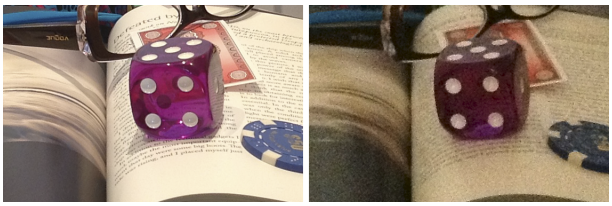

Figure 3.20: Detail of two images with different Signal-To-Noise-Ratio (SNR).

The ORB detector shows a good trade-off between repeatability in several tests and computational efficiency, however we have observed that spatial locations of their interest points are usually clustered in very close spatial locations. This spatial clustering may cause that descriptors extracted from such regions are not distinctive enough in order to effectively perform discriminative matches across images. Conversely, BRISK detector shows very similar robustness measure responses compared with ORB being computationally faster and, more importantly, generating much more uniform spatially distributed and stable interest points. The weakest aspect of BRISK in our results is its sensitivity to light intensity changes. Both, the number of detections and the repeatability scores decrease drastically as light intensity decreases. Fortunately, the number of computer vision scenarios with such a difference in light exposure is limited, mainly appearing in applications related with outdoor tracking or SLAM where light conditions are not controlled and may vary significantly, from image to image.

Affine transformation robustness is a very important measurement, because projective transformations can be locally approximated by an affine transformation. KAZE and MSER detectors obtain very good robustness results. Despite of MSER robustness to affine transformation, we have observed that this approach tends to generate a low number of detections because it needs extensive, well-defined homogeneous regions. This feature can be a serious limitation in many real practical applications. In addition to robustness to affine transformation, we think that invariance to scale geometric transformation is a critical aspect regarding many interest point matching scenarios, such as camera tracking or object recognition. In this aspect, SIFT is still the best performing algorithm, generating the most stable interest points along different scale factors. Another good performer regarding scale transformation is BRISK, being much faster than SIFT, thus more suitable for real-time operation. Finally, when real-time operation is a critical requirement, efficient approaches such as FAST, ORB, BRISK or STAR are the most appropri-

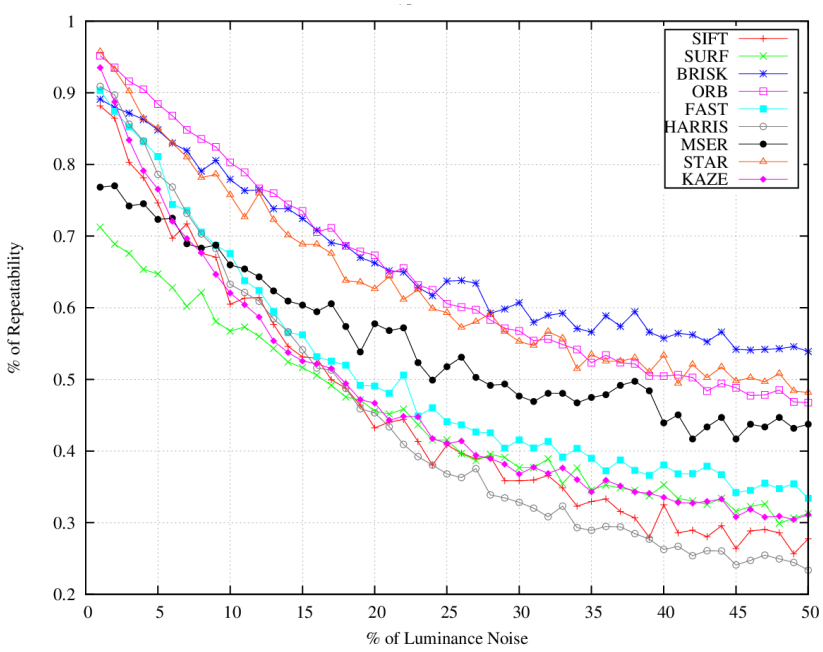

Figure 3.21: Repeatability results of noise photometric transformation.

ate. FAST is a very efficient approach, regarding CPU and memory consumption, but is very unstable regarding scale transformation.

Overall, we can conclude that recent BRISK detector obtains the best ratio between robustness and efficiency. ORB shows the best performance over rotation transformation, while BRISK shows great performance in scale, affine and projective transformation, being the fastest approach followed by FAST. ORB is a modification of FAST, which does not have an orientation component and does not produce multi-scale features. Therefore FAST is not as accurate as ORB dealing with rotation and scaling transformations.

Nowadays, efficiency is a very important aspect, as more and more applications are being migrated to mobile devices, such as iPad or iPhone. In that way, approaches similar to FAST or BRISK detector, requiring low computation and memory resources, are very useful and promising. The next step is to evaluate some of these algorithms on mobile devices, taking into account that some implementations must be rewritten and optimized for running on specific processor architectures using specific instructions and with several restrictions regarding parallel execution or memory management.

Chapter 4

DITEC Descriptor

This Chapter reviews the most relevant approaches to local feature description. We propose a new mechanism for global and local region description or representation based on the Trace transform. Additionally, we show the results of an evaluation of state-of-the-art approaches performance against different photometric and geometric transformations. This chapter is organized as follows: Section 4.1 gives some introductory remarks regarding local feature descriptors. Section 4.2 introduces image transformation that forms the basis of DITEC descriptor. Section 4.3 describes in detail our implementation of DITEC as local descriptor. In Section 4.4 a parameter sensitivity analysis of local DITEC approach is given. In Section 4.5 a review of most relevant approaches of state-of-the-art about local feature descriptors is shown. Finally, in Section 4.6 a detailed evaluation of several feature descriptors is given by studying their behavior against several geometric and photometric transformations.

4.1 Introduction

Decomposing or dividing an image into local regions of interest or features is a widely applied technique in many computer vision applications. As mentioned in Chapter 2 describing images of objects, structures or scenes by exploiting their local appearance properties instead of using the image as a whole can alleviate several problems such as occlusion, clutter or noise. Image representation, object recognition and matching, 3D scene reconstruction or motion tracking are some of many computer vision applications that rely on the extraction of stable, representa-

tive feature descriptors in the image. Different needs of these and other computer vision applications had motivated the development of many different approaches to local image representation of feature description.

As described in Chapter 3 an ideal interest point detector can extract interest points robustly, given both geometric and photometric variations or transformations between images being processed. Similarly, an ideal feature descriptor would capture the most important and distinctive information content surrounding interest point regions, such that the same underlying structure can be recognized or matched, even if it is captured in different conditions, i.e. under different geometric or photometric transformation. Local invariant image features are widely used for correspondence, as they can be efficiently extracted and matched between images without an explicit knowledge of the geometric or photometric transformation relating different images.

4.2 DITEC Descriptor

One of the tools that is of great benefit when dealing with information extraction from simple pixel intensities is the ability to transform an image from the spatial domain to an alternative domain, in which information can be more easily extracted or disposed. Some important examples of spatial domain image transformations, widely used by image processing community, can be the Fourier transformation, Laplace transformation, Mellin or Hough Transformation, among others. All these transformations share the same feature of transforming image pixel intensities into some other values representing different entities. In such a way, for example Fourier transformation 4.1 converts a time domain function $f(t)$ or spatial pixel intensities $f(x,y)$ in case of images, into a frequency domain function $f(\omega)$. In such domain, every point ωrepresents the amount (amplitude) of sinusoidal signal of frequency ωthat is present in original signal $f(t)$. The phase of this component is also represented and thus, the coefficient of the Fourier transform belong to the complex domain $\mathbb{C}$. After this transformation takes place, some operations such as canceling or removing some frequencies, or increasing and decreasing the amplitude of some others are easily carried out in frequency domain than performing them in the original time domain. In case of Fourier transformation applied on images, some operations such as edge detection or image smoothing, can be easily performed by canceling low frequencies or removing high frequencies, respectively.

$$f(\omega) = \int f(t)e^{-i\omega t}dt = \int f(t)[cos(\omega t) - sin(\omega t)]dt \tag{4.1}$$

One important property of the Fourier transform for signal processing is that the convolution in the temporal domain of two signals becomes into a pointwise product in the Fourier domain. This property is known as the convolution theorem.

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \mathscr{F}^{-1}\{\mathscr{F}\{f\} \cdot \mathscr{F}\{g\}\} \tag{4.2}$$

Our approach to local image description is also based on an image transformation. This approach is based on a global image descriptor or identifier proposed by Olaizola et al. [85]. We call this implementation DITEC.

Like many other function or image transformation, DITEC approach is based on the transformation of the image space into a parameter space, similar to Hough transformation for example. More precisely, we propose to use the trace transform [86] as a basic transformation for our local image descriptor.

4.2.1 Trace Transform

The trace transform, or tr-transform was originally proposed by Fedotov et. al in [86]. In this contribution Fedotov et. al proposed an approach based on image transformation as a solution of a pattern recognition problem, for the identification of different types of blood cells, such as erithrocytes. They proposed to convert the image space S in a parameter space, by intersecting several lines l' with S, represented in polar coordinates as 4.3.

$$l' = \{(x,y) : xcos\phi + ysin\phi = p\} \tag{4.3}$$

$$l' = l'(p,\phi) \tag{4.4}$$

In 4.4 l' is characterized by distance p from the origin(center) of S to l', and by angle ϕ(up to 2π). Points (x,y) represents the Cartesian coordinates of the plane or image. Figure 4.1 shows an example where image plane S is intersected with line l' in a circle or radius R.

A set of all straight lines intersecting a circle or radius R, centered in the image plane, can be described by 4.5. The set of all straight lines are topologically equivalent to a Möbius band [87].

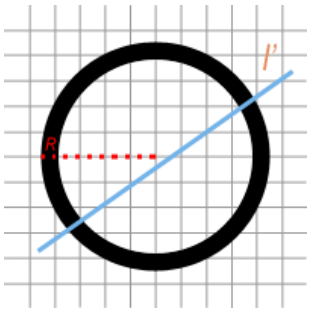

Figure 4.1: Intersection on lines l' with image plane S.

$$\Lambda = \{(\rho, \phi) : 0 \leq \phi \leq \pi, -R \leq \rho \leq R\} \tag{4.5}$$

While intersecting lines l',along the ranges of ρ and ϕ, with the plane S, a function T can be applied to the values, i.e. pixel intensities, along each line. Thus, the set of values than forms 4.6, i.e. all point in Möbius band Λ, was coined by the Fedotov et. al [87] as a Trace-transform. The function T is usually described in many contributions as the *trace functional* [7, 88].

$$g(\rho, \phi) = T(l'(\rho, \phi), S) \tag{4.6}$$

4.2.2 Radon Transform

Radon transform is a well known and widely applied image transformation used in many applications of signal or image processing. For example, this transformation is very important in current medical imaging research areas, as it forms a basic tool for computed tomography (CT) image acquisition. This type of imaging is obtained by a device such as the one depicted in Figure 4.2 (left).

This device is based on a setup, where a X-Ray emitter and a receptor(camera) are arranged one in front of the other in a circle of radius R. This setup rotates 2π radians around the patient lying in the bed, as shown in Figure 4.2 (right).

At each round of the setup at time t, the device obtains a Radon transformed image or sinogram, where the values of each point in these images represents the x-ray attenuation received by the detector, due to the X-ray absorption in the different tissues where the X-ray passed through. By combining many sinograms along the patient axial direction and reconstructing the original signal, i.e. obtaining the

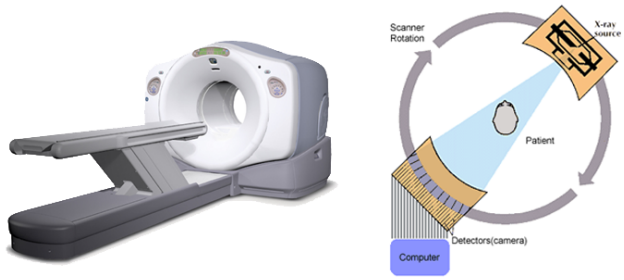

Figure 4.2: X-Ray Computed Tomography device.

inverse of the Radon transform, a volumetric image of the internal anatomy of the patient can be recovered, as shown in Figure 4.3.

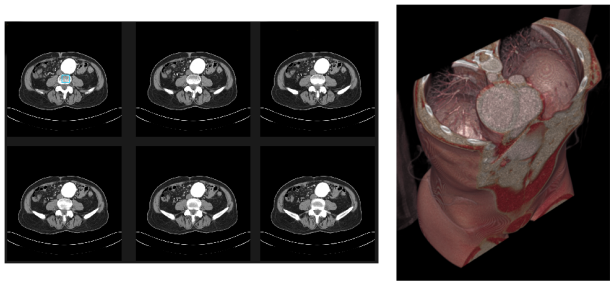

Figure 4.3: (Left) 2D axial slices of a 3D volumetric CT image, (Right) 3D volume visualization of reconstructed image.

The Radon transform of a continuous function $f(x,y)$ can be defined as 4.7,

$$g(\rho,\phi) = \iint f(x,y)\delta(\rho - xcos\phi + ysin\phi)dxdy \tag{4.7}$$

where δrepresents the delta Dirac function 4.8.

$$\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases} \tag{4.8}$$

The equation 4.7 can be interpreted as the transformation of image space $f(x,y)$ into a parameter space $g(\rho,\phi)$, where each point of that space represents the integral of values of $f(x,y)$ along a line l, given the pair of parameter (ρ,ϕ), i.e. the

parameters of a given line intersecting f. Parameters (ρ, ϕ) do not have lower or higher limits, but in discrete implementations, as will be shown later in this chapter, a limited number of samples in both directions are needed.

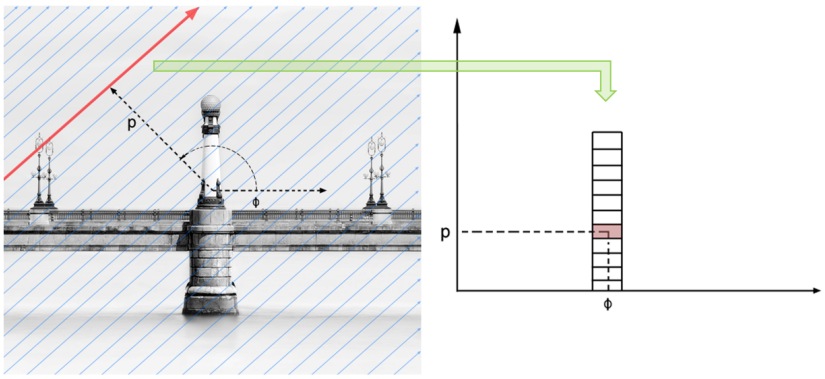

Figure 4.4: (Left) Input image and sampling lines, (Right) Column of the Radon Transform matrix corresponding to orientation ϕ and sampling ρ.

Figure 4.4 shows a 2D image where several straight lines l' at distances ρ from image center and orientation $\phi$are intersected with it. By using Equation 4.7, pixels forming each image line l' are summed up,i.e. integrated, and stored in the Radon image given by the coordinates (ρ, ϕ). The matrix resulting by intersecting the set of lines l', given the ranges of ρ and ϕ,forms the Radon Transform. This transformation allows line integrals in the(x, y) domain to be mapped into points in the (ρ, ϕ) domain. The inverse transform performs the inverse mapping of generating ray paths (lines) in image domain from points in (ρ, ϕ) space.

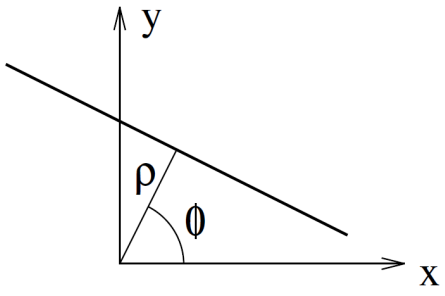

Figure 4.5: The two parametersϕ and ρ used to specify the position of the line.

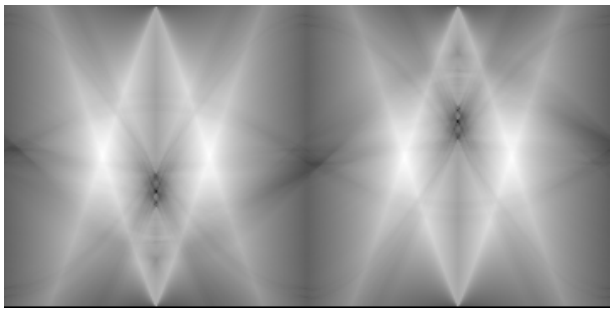

Figure 4.6: RADON Transform of the bridge image.

Figure 4.6 shows an image of the Radon transform of the previous image with ranges $\rho(-200,200)$ and $\phi(0,2\pi)$.

Recalling the Trace transform, we can see that the Radon transform is a particular case of the Trace transform, where the trace functional T is the integral function. Thus, as described in [7] the Trace transform can be considered as a generalization of the Radon Transform, where other functions can be used in place of the integral function.

Similarly, other transforms such as the Hough transformation can be seen as a special case of the Trace Transform. Hough transform is usually applied in computer vision based application for the detection of lines. This transform is usually applied to binary images edge maps, obtained by any type of edge detector such as Cany [89], and counts the number of edgels along each tracing line. This number is then plotted as a function of the two line parameters, as in Radon or Trace Transform, to form the Hough space. In this space, the maximum values represent the most probable areas,i.e. pixels, in original image space S where a line passes through. As in Radon and Trace transform, the ranges of both directions (ρ,ϕ) must be defined beforehand, i.e. the dimensions of the parameter space must be described before computation.

The image depicted in Figure 4.7(left) shows 4 lines at different locations and orientations with respect to the horizontal axis. In Figure 4.7(right) the Hough transformation of previous image is shown. As can be seen, there are 4 maximum points represented as the most brightest gray values in the image. These points are located in 25,45,70 and 88 degrees, representing the 4 lines in the original image space S. By getting these maximums the parameters of the 4 lines in the original image can be recovered.

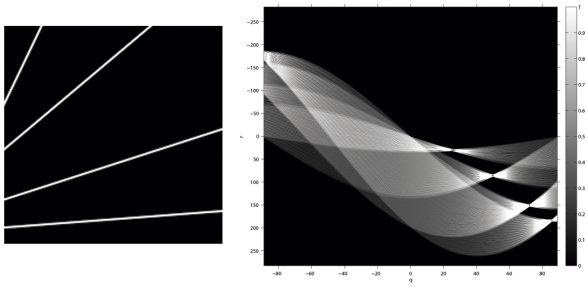

Figure 4.7: Image of four lines or edges(left). Hough transfor(right).

As we can see, Hough transform can also be seen as a particular case of the Trace transform where the trace functional acts as a voting scheme, by increasing in one unit the cell of the Hough space corresponding to parameters (ρ, ϕ), each time the corresponding pixel in image space S contains a value different from a neutral value such as 0. In fact, suppose we have a function $g(x,y)$ that contains a certain line as in 4.9,where the function has non-zero values only when (x,y) lies on the line of parameters (ρ', τ').

$$g(x,y) = \delta(y - \rho' x - \tau') \tag{4.9}$$

Applying the Trace transform with integral functional as in Radon transform we have:

$$\begin{aligned} g(\rho,\tau) &= \int\int \delta(y - \rho' x - \tau')\delta(y - \rho x - \tau)dxdy \\ &= \int \delta((\rho - \rho')x + \tau - \tau')dx \end{aligned} \tag{4.10}$$

where when $\rho = \rho'$and $\tau = \tau'$,i.e. we are in the line defined in g. The result is written as an infinite function integrated over an infinite interval, hence the result is infinite in that point ([90]) . Therefore, the trace transform with integral functional,i.e. the Radon transform, of a line produces a peak with infinite value in the parameter domain. Conversely, the position of the peak in parameter domain matches the line in spatial domain. This property forms the basis of the mentioned Hough transform, as well as many other curve parameter detection algorithms ([91]).

4.2.3 Trace transform functionals

As we have seen in the introduction, mainly what differs the Trace transform of the Radon transform is the generalization of the function to be applied to every straight line that compose the parameter space domain. While in Radon transform such a function is the integral, in trace transform can be any other type of operation. In [7] the authors carried out an extensive evaluation of several functionals, looking for those that shows relevant properties such as invariance to spatial transformation. In addition to the functionals T (*trace* functional) to be applied to every line while constructing the parameter space domain, the authors proposed to use two more functional named the *diametrical* and *circus* respectively. The study was validated on a recognition application of images of several fishes. The diametrical functional D would be applied to the columns of Trace transformed image as depicted in Figure 4.8.

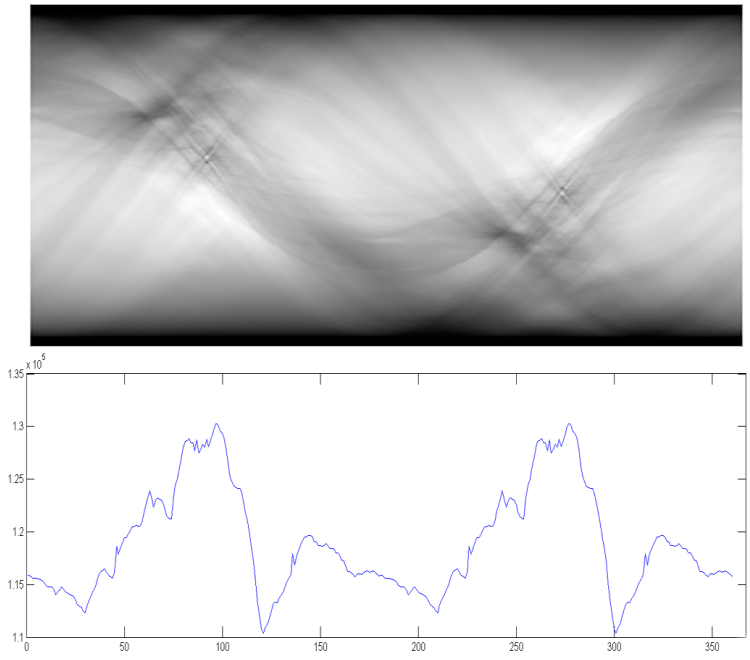

Figure 4.8: (top) Trace transform image computed with integral trace functional. (Bottom) Diametral functional F applied to the image of the trace transform.

As with trace functional *T,* diametrical functional can be any type of function.

In Figure 4.8 a functional consisting on selecting the maximum value in each column was applied. Finally, circus functional C is applied to the values resulting of applying diametrical functional D, as depicted in Figure 4.9. In this image, the values of diametrical functional appears as a 1D array. The result of applying C over this array is a single number or value. This value acts as a descriptor or as a representation of the original image. The combination of these three functionals was coined in [7], as the triple feature extraction method.

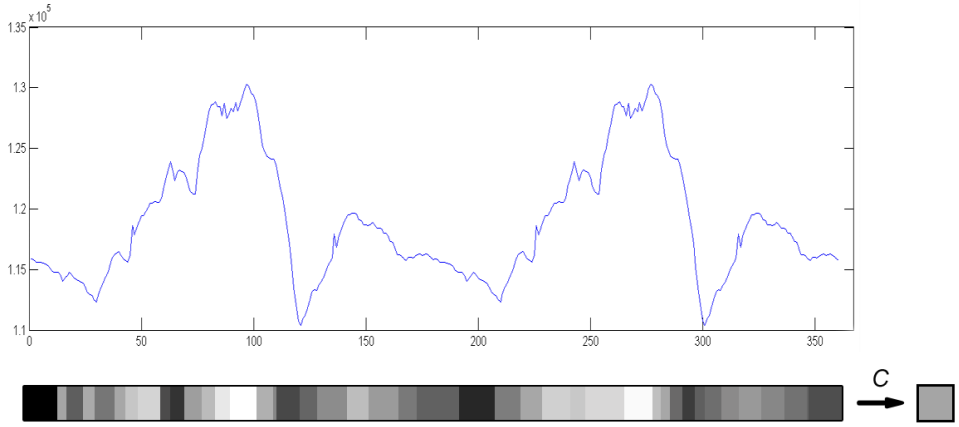

Figure 4.9: (top) Diametrical functional plot, (bottom) Diametrical functional as 1D array, and result of Circus functional C.

Clearly, the definition and combination of different Trace functional and Circus functionals respectively results in different properties of the final descriptor.

In [92] an implementation of a Trace transform based feature extraction for CBIR in specialized hardware is proposed, obtaining a throughput of 2725 images per second.

4.2.4 Properties of the Trace transform

As described in [93, 90] Radon transform has some relevant properties related with spatial transformations, such as shifting or translating, rotation and scaling. These geometric transformations in addition with other geometric transformation such as projectivities, along with photometric transformations are very important for several computer vision application such as object recognition or camera tracking, because a change of view point or acquisition parameters can severely change the

Name	Functional
IF1	$\int \xi(t)dt$
IF2	$(\int \lvert\xi(t)\rvert^q dt)^r$
IF3	$\int \lvert\xi(t)'\rvert dt$
IF4	$\int (t-(\int t\xi(t)dt/IF1))^2 \xi(t)dt$
IF5	$(IF4/IF1)^{1/2}$
IF6	$max(\xi(t))$
IF7	$IF6 - min(\xi(t))$
IF8	Amplitude of 1st harmonic of data set$\xi(t)$
IF9	Amplitude of 2nd harmonic of $\xi(t)$
IF10	Amplitude of 3rdt harmonic of $\xi(t)$
IF11	Amplitude of 4th harmonic of $\xi(t)$

Table 4.1: List of Trace Transform functionals proposed in [7].

final appearance of the objects rendered in the images. Thus, a mechanism that can be robust or invariant to those types of transformations could be used or applied as a basic tool for image description.

Following some properties of the Radon transform are shown. It worth noticing that its generalization, i.e., the trace transform shares the same properties but heavily depending on the trace functional employed. For example, some functionals such as *IF6* or *IF7* are not continuous nor differentiable, thus these properties are not directly applicable and should be studied separately.

- Translation or shifting

Imaging we have a 2D point translation (x_0, y_0) of $f(x,y)$ such that $h(x,y) = f(x - x_0, y - y_0)$. The Radon transform of $h(x,y)$ is given by:

$$\begin{aligned} g(\rho,\phi) &= \int f(x-x_0, \phi x + \rho - y_0)\, dx \\ &= \int f(\tilde{x}, \phi(\tilde{x}+x_0) + \rho - y_0)\, d\tilde{x} \\ &= \tilde{f}(\rho - y_0 + \phi x_0, \phi) \end{aligned} \tag{4.11}$$

where $\tilde{x} = x - x_0$. Thus, only the offset parameter is changed after image translation, i.e. the change occur only in the ρdirection.

- Scale

Given $f(x,y)$ as the original input image function, assume we have a new scaled version os such a function like $h(x,y) = f\left(\frac{x}{a}, \frac{y}{b}\right)$ where $a > 0$ and $b > 0$. In case $a = b$ scale transformation would be isotropic and anisoptric otherwise. The Radon transfor of $h(x,y)$ is given by:

$$\begin{aligned} g(\rho,\phi) &= \int f\left(\frac{x}{a}, \frac{\phi x + \rho}{b}\right) dx \\ &= a\int f\left(\tilde{x}, \frac{\phi a\tilde{x} + \rho}{b}\right) dx \\ &= \tilde{f}\left(\frac{\rho}{b}, \frac{\phi a}{b}\right) \end{aligned} \tag{4.12}$$

where $\tilde{x} = \frac{x}{a}$. As shown in 4.12 in case of isotropic scaling $(a = b)$, only a change in ρ occur. In any case,4.12 shows that Radon transform nor the trace transform are not scale invariant, but changes only in one dimension accordingly with the value of the isotropic scale.

- Rotation

Given image function $f(x,y)$in polar coordinates (r,φ) and given the rotation φ_0 the Radon transform can be expressed as:

$$\begin{aligned} g(\rho,\phi) &= \int\int f(r,\varphi)\delta(\rho - rcos(\varphi - \phi))drd\varphi \\ &= \int\int f(r,\varphi - \varphi_0)\delta(\rho - rcos(\varphi - \phi))drd\varphi \\ &= \int\int f(r,\varphi')\delta(\rho - rcos(\varphi' + \varphi_0 - \phi))drd\varphi \\ &= g(\rho,\varphi_0 - \phi) \end{aligned} \tag{4.13}$$

where $\varphi' = \varphi - \varphi_0$. As shown in Equation 4.13 after an in-plane image rotation, Radon transformation and therefore trace transformation using functionals such as *IF1* or *IF2* changes only in ϕaxis. More precisely, $\varphi_0$rotation is represented as an equivalent linear translation along ϕdirection.

From image in Figure 4.10 we get a circular region or interest (ROI), as depicted in Figure 4.11(a). This region

As shown in plots depicted in 4.12, the Trace transform along with some specific functionals can represent or can retain directional information. The rotation

Figure 4.10: Input testing image.

in spatial domain leads to circular translation along the ϕaxis in the parameter domain. This feature has been successfully employed in approaches for texture estimation ([94]) or texture recognition ([88]). Analyzing the diametrical functional result, we could detect the maximum value and then rotate back the image till this maximum is at 0. This way, we could rectify the image prior to further processes such as feature extraction. This rotation can be seen as an image normalization that can improve image recognition or matching.

As a summary we can stated that any translation in image space $f(x,y)$ leads a translation in the ρ direction in trace transform parameter domain. An isotropic scaling of the original input results in a scaling in ρdirection and the value of the transform is also scaled relatively to the form of the given trace functional. And finally, a rotation in the space domain leads to circular translation along the ϕaxis in the trace transform parameter domain. These results are depicted in Table 4.2.

Transform	**Input image function f**	**Trace Transform ($IF1$)**
Identity	$f(x,y)$	$g(\rho,\phi)$
Translation	$f(x-x_o,y-y_0)$	$g(\rho-x_0cos\phi-y_0sin\phi,\phi)$
Isotropic Scale	$f(\alpha x,\alpha y)$	$\frac{1}{\|\alpha\|}g(\alpha\rho,\phi)$
Rotation	$f_{polar}(r,\theta+\theta_0)$	$g(\rho,(\phi+\theta_0)\,mod\,2\pi)$

Table 4.2: Results of Trace transform after image geometric transformation.

Graphically, the results described in Table 4.2 are depicted in Figure 4.13.

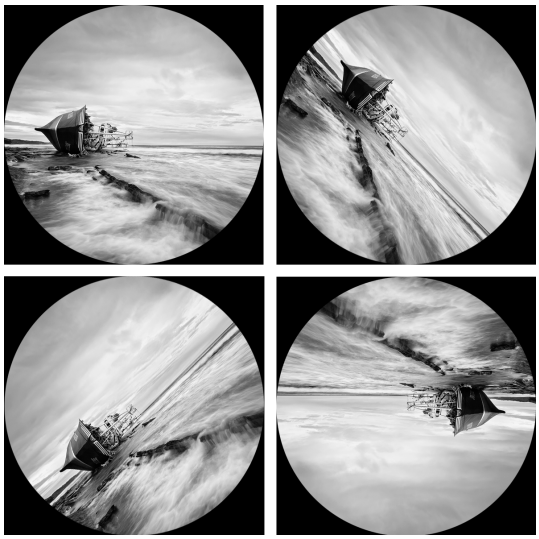

Figure 4.11: Oriented ROIs.

4.3 Implementation of the trace transform as local descriptor

DITEC descriptor does not perform interest point detection, so the detection relies in any point detector does perform scale estimation. Even if trace transform with appropriate functionals exhibits some degrees of invariance to scale transformation, better results are obtained by normalizing image patch according to the scale estimated by other mechanism, such as the detection of the extrema of a Laplacian based function applied in a space-scale framework, similar to SIFT ([46]) or BRISK ([50]). Therefore, DITEC descriptor rely on an interest point extractor mechanism in order to be robust to scale transformation. Moreover, as will be explained later, thanks to the performance of DITEC descriptor against in-plane rotation, it does not need that an interest point extractor estimates a dominant orientation, as many of the approaches do, as described in Chapter 3. Figure 4.14 depicts the processing pipeline we propose for local image description using DITEC.

As the first step of the pipeline there is the interest point extraction mechanism. As seen in chapter 3 there are approaches that are robust or co-variant to different types of geometric or photometric transformations. Approaches that are co-variant to isotropic scale transformation used to integrate in their processing

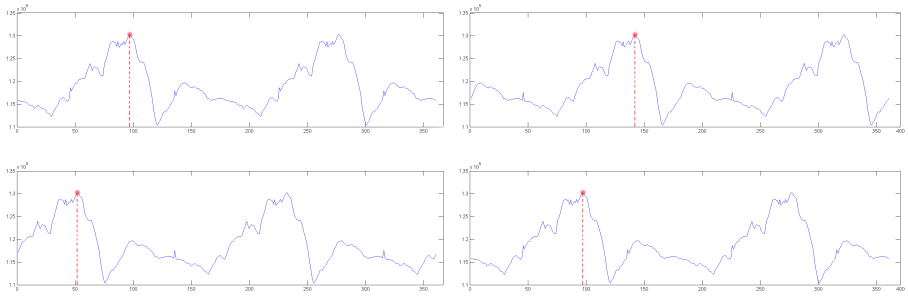

Figure 4.12: (a) Diametral functional IF6 applied on Trace Transform of original image, (b) same functional applied on Trace transform of image rotated 45º, (c) and (d) same functional with image rotated -45º and 180º respectively.

pipelines a mechanism for scale estimation. Once the set of interest points are extracted, DITEC approach is applied independently to every detected point. The first task them is to perform scale approach is inspired by the human visual system. In many computer vision applications, such as those based on augmented reality [95], the camera can be freely moving around the scene. Because of these camera movements, world structures are observed differently given different points of view. Therefore, those structures will be rendered differently in the images due to different perspective distortion. In this way, if, for example, the camera moves back and forth from a given world structure, this structure will appear with different apparent scale, given the relative position between it and the camera, and given that the camera's focal length remains constant during acquisition. Image patches extracted from those images will show structures and patterns that will differ, at least, in a given scale factor. As described in chapter 3 many interest point approaches are invariant, or better said co-variant, to scale geometric transformation, and therefore are able to estimate locally an apparent scale for a given interest point.

We use an approach similar to [46] for scale normalization. We use the scale value estimated by interest point extracted to get a rectangular patch of size $32k\,x\,32k$,where k represents the estimated scale value. Once this patch is extracted, we apply size normalization by scaling the size of the patch to a nominal value of 32 pixels with either down sampling or up sampling. Scale normalization is performed by using bi-linear interpolation for avoiding the generation of aliasing artifacts.

After scale normalization we propose to use a simple histogram equalization

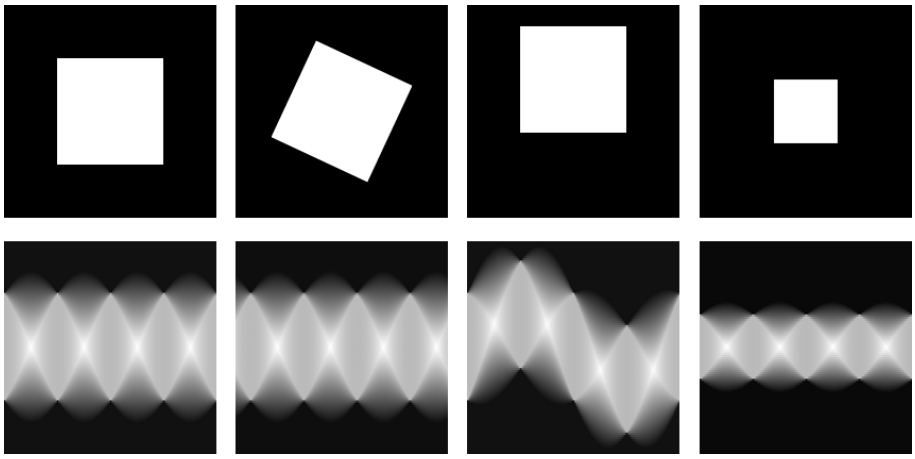

Figure 4.13: (Top) Input images, (Bottom) Images of Trace Transform.

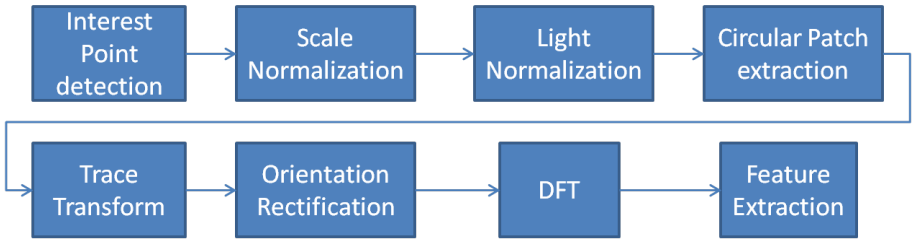

Figure 4.14: DITEC Local descriptor Pipeline.

for normalizing the dynamic range of the patch. This normalization improves the performance of the descriptor against light intensity photometric transformation. It is worth mentioning than many functionals, such as integral function used in Radon transform, are not invariant to exposure or light intensity photometric transformation. This functional only integrates,i.e. summed up the values of the intensity(luminance) of pixels, thus a simple linear intensity transformation can degenerate the image of the trace transform, thus generating a different descriptor.

Once dynamic range is normalized, we propose the extraction of a circular patch from the original rectangular patch. As described in Chapter 3 interest point detectors selects pixels that represents special regions of images, such as blobs, corners, or regions that correspond with specif criteria such as the value of local curvature. After interest points are extracted, usually squared regions of image are extracted around each detected point. As described previously, in some computer

vision applications such as augmented reality or SLAM camera can be freely moving around world points. Depending on the orientation of the camera with respect to a world reference plane, for example, distortions introduced by perspective can be very acute, such as the one depicted in Figure 4.15.

Figure 4.15: Frontal view of a world plane (left), perspective distortion induced by camera-to-world plane orientation(right).

If we extract a square region around the manually selected interest point match (green line) in both images, we obtain the patches shown in Figure 4.16. As we can see, in the patch extracted from perspectively distorted image appear new textures not present in the fronto-parallel view extracted patch, hence rectangular patch can not cope with the geometric deformations caused by the change of view point. As a consequence, descriptor computed from both patches may be very different, thus matching those interest points may be impractical.

Figure 4.16: Rectangular patch from fronto-parallel view(left), rectangular patch from oblique view(right)

Ideally, a region detector should be covariant with perspective of affine transformation, i.e. should extract regions already adapted to the transformation. Some approaches such as MSER ([71]) directly extract elliptic regions around interest points that can well approximated to affine transformations. These regions can be rectified back to as canonical circular shape before descriptor computation. As shown in the evaluation in Chapter 3these techniques are very robust against acute

projective transformations, but requires well structured images, with wide homogeneous well described regions. However, shows weakness when images contain high frequency, i.e. very small regions, or repeated patterns.

In our approach, we propose to extract a circular patch from detected interest points. By extracting a circular patch we somehow favor central region closer to the detected interest point. If detector mechanism correctly identified the same interest point in both images, at least the central part will be common in both images, irrespective of the deformation due to perspective projection or scale transformation. This approach is similar to the one proposed in [6]. This approach is inspired by the human visual system and more precisely on the geometry of the retina, arguing that the spatial distribution of ganglion cells(photo receptors) reduces exponentially with the distance to the fovea, that is the area of the retina responsible of the central (high resolution) human vision. They mimic this cells distribution by sampling and weighting pixels in the extracted patch accordingly. Circular patch is also applied in other similar approaches like ORB ([52]) where mask is used for central moments computation, for dominant orientation estimation.

As shown in Figure 4.17 we first extract a square scale normalized patch and then apply a circular mask before descriptor computation.

Figure 4.17: (Left) Source patch, (Center) Circular Mask, (Right) Final Circular patch

Is important to mention that by using only the mask we will be altering the trace transform image of the original patch. When the mask is directly applied to the original patch, black pixels are introduced in the corners of the image. If we apply a trace transform to the resulting image, not distinguishing those pixels, we would get a different descriptor between original patch and masked patch. Figure 4.18 shows the shape of two DITEC descriptors of 128 dimensions applied on original image of Figure 4.17 and on the masked patch, by using *IF7* as trace functional. As we can see, both plots do not match, having an Euclidean distance error between them of 1.76. This difference is clearly introduced by the noise added in form of

black pixels due to masking process.

Figure 4.18: Shape descriptors with mask(red) and without mask(green).

This effect can be more acute if we use a non-linear Trace functional such as *IF*2 (see table 4.1) where values equal to zero may have a big influence in the final trace result. In opposite, some linear functionals such as Radon functional, i.e. *IF1* of table 4.1, are less affected by the noise added due to the use of masking.

In order to alleviate this phenomena, we remove the pixels on the corners during trace transform computation, thus achieving a very similar descriptor compared with the non-masked patch while still retaining a circular patch, as shown in Figure 4.19.

In this case, we compute the intersections between the curve represented by the circular mask, and every line representing the directions of the trace projections, given the parameters (ρ, ϕ) as described by Equation 4.14. Therefore, by calculating these intersection points we can limit the computation of the trace transform to only the pixels inside the mask.

$$\begin{aligned} y &= R\left(\sqrt{1-\frac{\rho^2}{R^2}}cos\phi + \frac{\rho}{R}sin\phi\right) \\ x &= R\left(\sqrt{1-\frac{\rho^2}{R^2}}sin\phi + \frac{\rho}{R}cos\phi\right) \end{aligned} \tag{4.14}$$

In this case, both plots are much more closer one from the other, showing

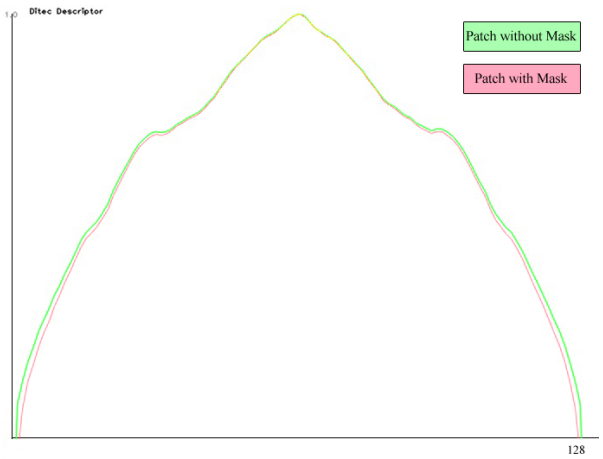

Figure 4.19: Shape descriptors with mask(red) and without mask(green) treated as circular patch.

an Euclidean distance error of 0.289. As will be described later in this chapter, computing DITEC descriptor as a circular patch instead of as a rectangular patch do have a very important influence during descriptor matching.

4.3.1 Trace Transformation

The core of DITEC descriptor resides in the trace transformation. As described in the introduction of this Chapter, the trace transform is a generalization of the Radon transform. In this step, we apply trace transform with a given trace functional. Our implementation is flexible enough to allow the definition of which functional to apply during descriptor computation, or even a combination of functionals. Trace transformation is applied only to the pixels that are inside circular mask, after intersection points are estimated. The trace transformation is formed by the computation of all trace projections, corresponding to the number of samples of parameter space for for ϕ and ρ.This number of samples must be supplied beforehand, during an initialization stage. When computing each trace projection, given a value ϕ and ρ respectively, pixels along that projection must be sampled. We have evaluated different strategies when dealing with trace projection extraction.

4.3.2 The importance of sampling

As shown in Figure 4.20 depending on the line parameter orientation ϕ, and due to image space discretization, only a reduced number of samples can be evaluated. This dependency on orientation ϕcan degenerate the image of the Trace transform because not all parameter space regions are equally sampled. For example, in Figure 4.20(Left) if orientation $\phi$of line l' would be parallel to x image axis, the resulting trace functional on that line would be computed by using 10 samples, while the line l' depicted in the Figure would be computed with only 6 different samples, if only pixels intersecting with line were used without any type of interpolation.

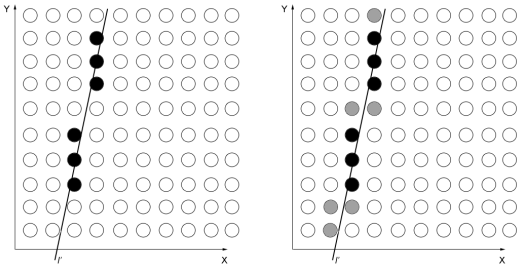

Figure 4.20: (Left) Intersection between straight line and image pixels without interpolation. (Right) Intersection between straight line and image pixels with Bresenham algorithm.

We use the Bresenham algorithm ([96]) to evaluate the pixels intersecting every line while computing the image of the Trace transform, as depicted in Figure 4.20(Right). In this way, every region of space parameter domain is better equally evaluated. Thus, the resulting Trace transform is more homogenous and therefore better representative of the underlying structures.

It must be noticing that the number of pixels visited by the Bresenham algorithm does depend on the line orientation, thus varies depending on the line parameters. In order to evaluate how different sampling strategies explore the parameter space (ρ, ϕ), we created a simple testing image as shown in Figure 4.21. In this experiment we used *IF1* (see Table 4.1) as trace functional. This functional is linear and by using a constant image,i.e. all pixels sharing the same value, the result of the trace transform can be interpreted as how the algorithm is exploring the parameter range in the space (ρ, ϕ),i.e. how many times a specific combination of values of ρ and $\phi$are evaluated.

Figure 4.22 shows the result of applying trace transform with functional *IF1* to

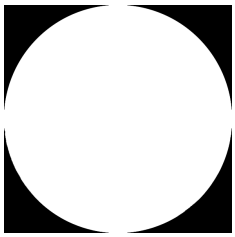

Figure 4.21: Circular patch image.

the image depicted in Figure 4.21. As can be seen, there are some dark lines along the image and small gradients from white to black around them. These darker lines coincide with the values of ϕ to $\frac{\pi}{2}$,π, and $\frac{3\pi}{2}$. These ρ values represents the vertical and horizontal orientations of trace lines.

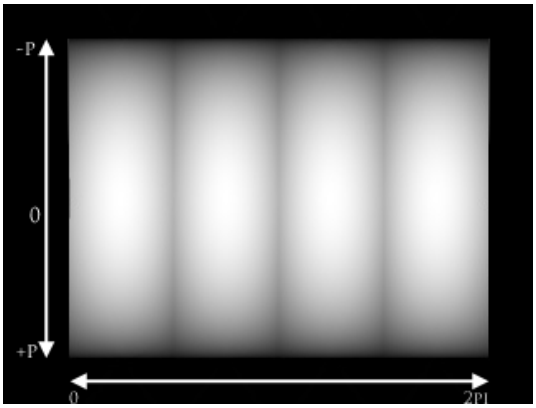

Figure 4.22: Result of (ρ, ϕ) space exploration with Bresenham.

Clearly, when evaluating these orientations Bresenham algorithm visits less pixels that, for example, orientations such as $\frac{\pi}{4}$ or $\frac{4\pi}{3}$. These differences in the number of visited pixels or samples can be interpreted as deformations of the trace Transform parameter space, induced by the sampling strategy employed while exploring it.

Figure 4.23 shows the result of applying the same functional to the same image but making two single rotations of $(+\frac{\pi}{4})$ and $(-\frac{\pi}{4})$ respectively. Then, we explore the space accessing to the positively rotated or the negatively rotated versions, depending if we are approaching to $\frac{\pi}{2}$ or π orientations. In this way, we ensure that all sampling lines are always being done at 45º around the diagonal of the image,

where Bresenham's algorithm more densely visits the image space.

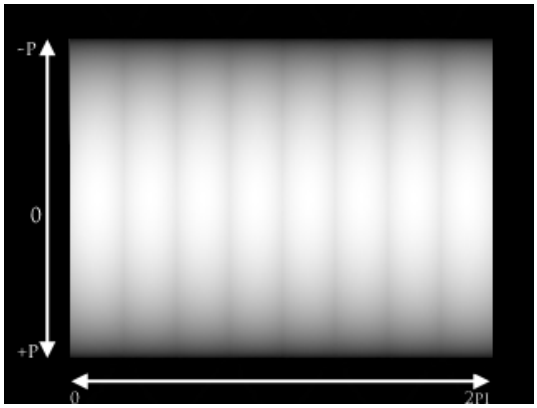

Figure 4.23: Result of (ρ, ϕ) space exploration with Bresenham with two image rotations.

As opposite to previous result, this image shows smoother gradients, thus is more uniform and therefore better sampled. As expected, deformations now appear each $\frac{\pi}{4}$radians due to image rotations.

Extending the idea of rotating the image for accounting the differences in sampling due to line orientations, we performed the experiment of rotating the image each ϕ step before each scan line is processed. In this way, each scan line given a value of ϕcorrespond with image columns. Therefore, all scan lines, i.e. trace projections, are computed by just rastering image columns, thus each line is equally sampled. As shown in Figure 4.24, all columns from 0 to 2π shows no difference, hence the parameters space (ρ, ϕ) is equally sampled, having no distortions in opposite to previous results.

Figure 4.25 depicts three different plots extracted at the same row of three different trace transformed images, each one sampled with one of the three different strategies described previously. As expected, the most uniform plot is given by the approach based on full image rotation. The most distorted plot is generated by using Bresenham sampling with no bi-linear interpolation and no image rotation. The approach based on two single $\pm 45^{o}$ rotations gives much better results compared with the no-rotation based approach, and does perform very well compared with full image rotation.

Table 4.3 shows the computation times needed to compute a Trace transform of 582x582 pixels with *IF1* as trace functional. The parameters space range for

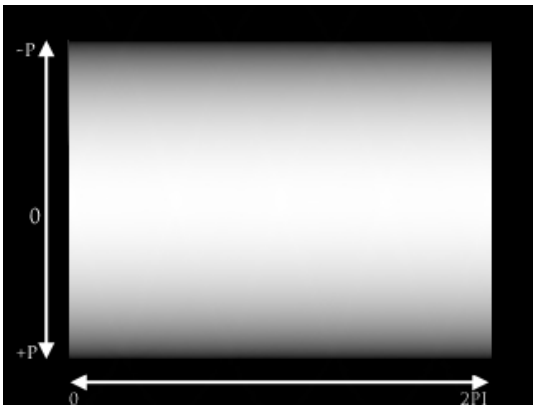

Figure 4.24: Result of (ρ, ϕ) space exploration with Bresenham image rotation each.

each trace transform was $\rho \in [0, 200]$ and $\phi \in [0, \pi]$. As can be seen, the method of full image rotation is the slowest because a whole image rotation must be accomplished for each value of ϕ from 0 to π. Image size is the most critical factor in this approach, because a lot of memory accesses are needed to perform whole transformation. In opposite, methods based on no image rotation or two single rotations are much faster. Tacking into account the distortion introduced

Sampling Strategy	Computation Time (ms)
Bresenham (No rotation)	**798**
Bresenham (Two rotations)	849
Full Image Rotation	2521

Table 4.3: Computation time (ms)

We performed the same experiment but applying the trace transform as a local descriptor. In this test, computation times measures the time needed to perform 1000 trace transform to the corresponding 1000 patches of size 20x20 extracted from a given image and with ρ and ϕ both set as 16. The test was carried out 10 times for statistical soundness.

Clearly, The fastest sampling strategy is the one where neither rotation nor interpolation is performed, but the approach based on two single rotation does not introduce much computation overhead and clearly introduces less distortion (see Figure 4.25), thus improving trace transform image. It is worth noticing that as im-

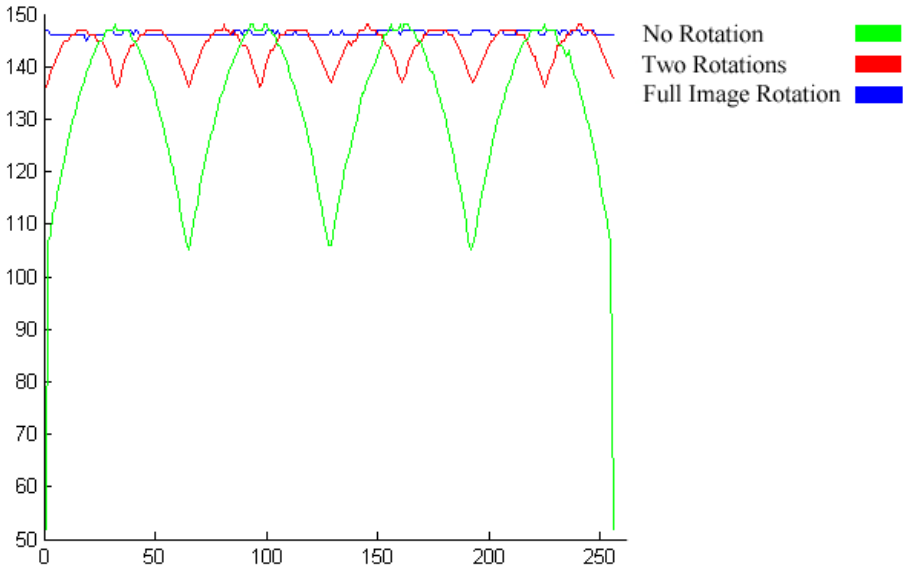

Figure 4.25: Result of different sampling strategies of (ρ, ϕ) space.

Sampling Strategy	Computation Time (ms)
Bresenham (No rotation)	**416**
Bresenham (two Rotations)	450
Full Image Rotation	540

Table 4.4: Computation time (ms)

age size decreases, the differences between the approaches is reduced significantly.

4.3.3 Orientation Correction

As described previously, patch orientation can be recovered by using circus functional. Maximum value of such function can be computed and then rotate the patch back till that maximum is set to 0. This process can normalize patches before descriptor matching, thus improving patch or image recognition and identification. A similar procedure can be found in several approaches such as SIFT ([46]) or ORB ([52]) where local discrete gradient orientations are computed around the interest points, and then the patch is rectified,i.e. rotated back, in the direction of the

maximum gradient value, as a means of normalization. As we can see later during descriptor evaluation, one of the advantages of our approach is that even not carrying out patch rectification it shows great performance against rotation transformation, being almost insensible to this transformation. This feature allows us to remove the orientation estimation step during descriptor calculation, thus saving computation time for other tasks such as increase the number of samples while exploring the (ρ, θ) parameter space.

4.3.4 Feature Extraction

The final stage of DITEC descriptor computation is the extraction of the features that finally will represent the descriptor. We propose to use Discrete Fourier Transformation applied to the image t of the trace transform, as described in Equation 4.15, where $\mathscr{F}(k,l)$ represents the frequency domain transformed image of time domain or spatial domain image$t$.

$$\mathscr{F}(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} t(i,j)e^{-i2\pi\left(\frac{k_i}{N}+\frac{l_j}{N}\right)} \tag{4.15}$$

It's worth mentioning that we apply DFT of t row by row, thus we perform a 1D discrete forward Fourier transform to every row of the trace transform image individually. This transformation allows us to rearrange or represent every wave of the sampled θ values of the trace transform image, more efficiently and effectively. Because we are interested only in the image structure, we therefore compute the magnitude of the DFT with Equation 4.16, where *RE* and *IM* represents the real and imaginary part of the Fourier transformation respectively.

$$M = \sqrt[2]{RE(\mathscr{F}(k,l))^2 + IM(\mathscr{F}(k,l))^2} \tag{4.16}$$

By using only the magnitude we remove information about phase coefficients, hence we obtain a phase-normalized signal characterization. By removing phase information we also get in-plane rotation invariance, since the magnitude of the Fourier transform is invariant with respect to circular translation of any function $f(x)$ in $[0,X]$, as described in[97]:

$$|DFT(f(x))| = |DFT(f((x+x_0)\,mod\,X|) \tag{4.17}$$

Figure 4.26shows the chain of transformations from input image to DFT image.

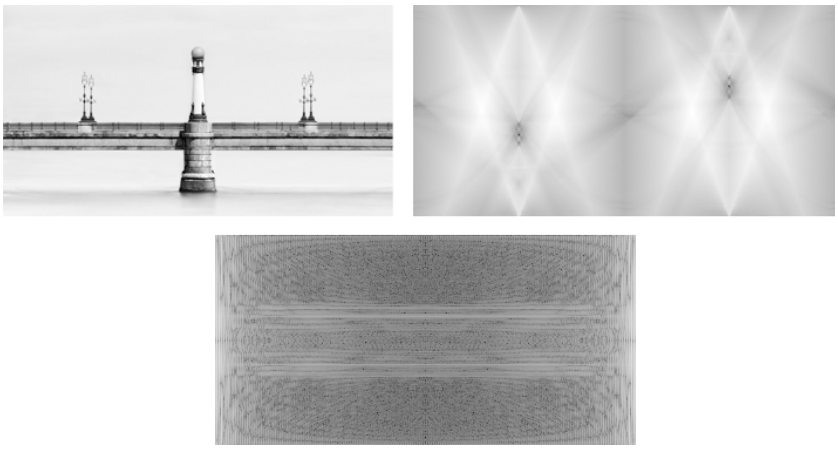

Figure 4.26: (Left) Input image, (Right) Trace transform image ,(Bottom) Magnitude of DFT rows of Trace Transform.

Final step consist on extracting the features from the image of the magnitude of the DFT. The number of features,i.e. the number of dimensions of DITEC descriptor must be defined beforehand. We will study the influence of dimensionality later in this chapter, during the evaluation part. For now,imagine that the number of dimensions of the descriptor is set to n. We therefore sample $\frac{n}{N_\theta}$ consecutive points of every row of the magnitude of DFT, where $N_\theta$represents the total number of bands of the DFT. The number of bands N_θ corresponds with the number of different values of θsampled when the trace transform was computed. The final number of dimensions of the descriptor have a very important influence during descriptor matching. As described in previous chapter, in real-time oriented computer vision applications simple euclidean distance is usually employed for estimating descriptors differences. Therefore, the higher the number of dimensions the longer times required for descriptor matching process.

4.4 Parameters sensitivity analysis

One very important step during any mathematical, physical, environmental or social model development consist of the determination of the parameters which are most influential on the final model results. A sensitivity analysis of these parameters is not only important for a better understanding of the model but also for addressing new lines of research. As stated in [98] modelers used to conduct sen-

sitivity analysis for a number of reasons including the need to determine:

- Which parameters requires additional research for a better understanding of the underlying model, and therefore reducing the output uncertainty.
- Which parameters are non significant for the model and thus can be avoided or eliminated from the final model.
- Which parameters contribute most to output model variability.
- Which parameters are most highly correlated with the output.

Conceptually, the simplest method to sensitivity analysis is to repeatedly vary one parameter at a time while holding the others fixed [98]. A sensitivity ranking can be obtained quickly by increasing each parameter by a given percentage while leaving all other parameter constant, and therefore evaluating or quantifying the change in the output model. This type of approach for sensitivity analysis is referred as a local sensitivity because only some discrete values of input parameters are evaluated and not the entire distribution. In the case of out implementation of DITEC as local descriptor, all input parameters can only be set to discrete values.

We made a sensitivity test for the different parameters that forms our local DITEC descriptor implementation. More precisely, we evaluated how the following parameters affects the final model in form of matching accuracy, i.e. number of correct matches.

- **Number of Phi Samples:** Number of angular samples obtained from a given image when computing the trace transform.
- **Number of Rho Samples**: Number of perpendicular sample lines obtained along trace projections when computing the trace transform.
- **Size of image Patches**: Square size of local image patches extracted around detected interest points.
- **Dimensionality of DITEC local descriptor**: Number of of features extracted from the DFT of the trace transform.

We carried out the next experiments in order to obtain a better understand of how different ranges of the input parameters influence the final behavior of local DITEC descriptor. We studied which input parameters as well as how their ranges mostly

affect the final accuracy of DITEC approach. This issue is very important when the approach is going to be employed by end users in their computer vision based applications. Is very useful for non-researchers, i.e. for practitioners, to have some parameters fixed and to known in advance some optimal ranges for the rest of parameters.

Following experiments are conducted with same image data set, compound of 6 different images, covering different image sizes, as well as geometric and photometric transformation, such as image translation, rotation, and projection, or image blurring, noise, or light exposure changes. For each parameter we evaluated the influence of the input value in the final matching ratio as well as the impact in the computation time. Every particular experiment was repeated 10 times for statistical soundness.

By default we used the following values for the rest of parameters that are not being changed during the evaluation of a particular parameter: phi and rho equal to 16, patches size equal to 20, descriptor dimensionality equal to 128 and sampling strategy based on two single rotation approach.

Phi Value

Figure 4.27 shows the results of changing the number of angular samples when computing the trace transform for each interest point detected in every image in the evaluation data sets. Phi parameter shows a clear convergence around 15 to 20 angular samples, using image patches of 20x20 pixels. Due to the small size of images and due to pixel discretization, introducing more angular after 16-20 samples does not add new data to the trace transform, thus does not improve the final performance.

Figure 4.28 shows the time needed to compute the trace transform of an image with different values of angular sampling. As expected, as the number if phi samples increases the computation time increases almost linearly.

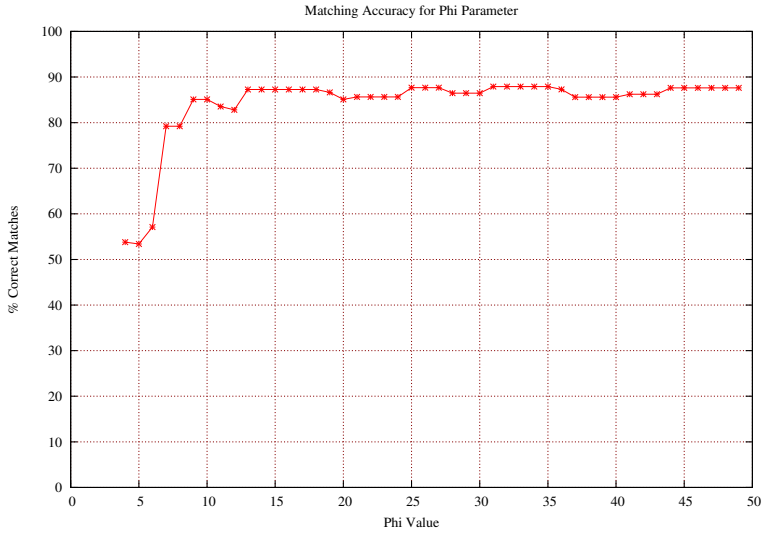

Figure 4.27: Matching accuracy depending on the number of phi samples.

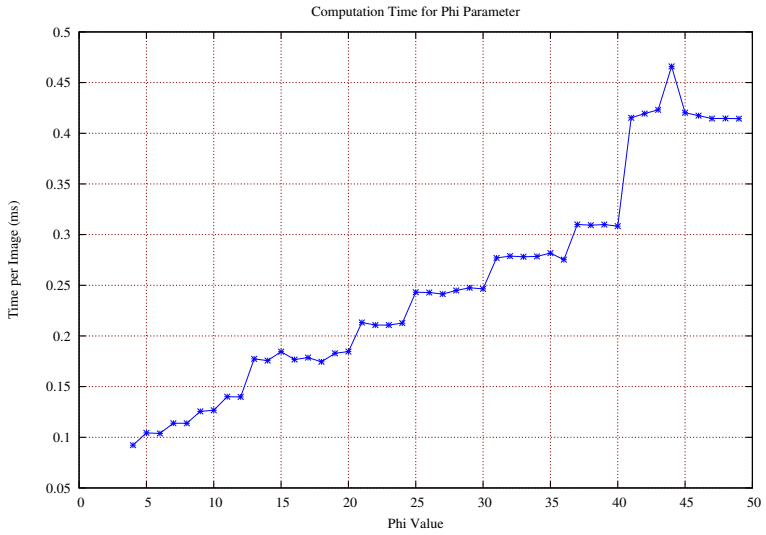

Figure 4.28: Computation time depending on the number of phi samples.

Rho Value

In this test we set all parameter values fixed as described previously but changing the Rho samples. In this case stabilization or convergence is reached around 15 Rho samples. From 25 and onwards the performance does not increase and also starts to decrease slowly at around 30. We think that this decrease in performance is due to oversampling of parameter space. When the number of samples exceeds available data, sampling starts to repeat already sampled regions, thus introduces artificial patterns that somehow distort the trace transform.

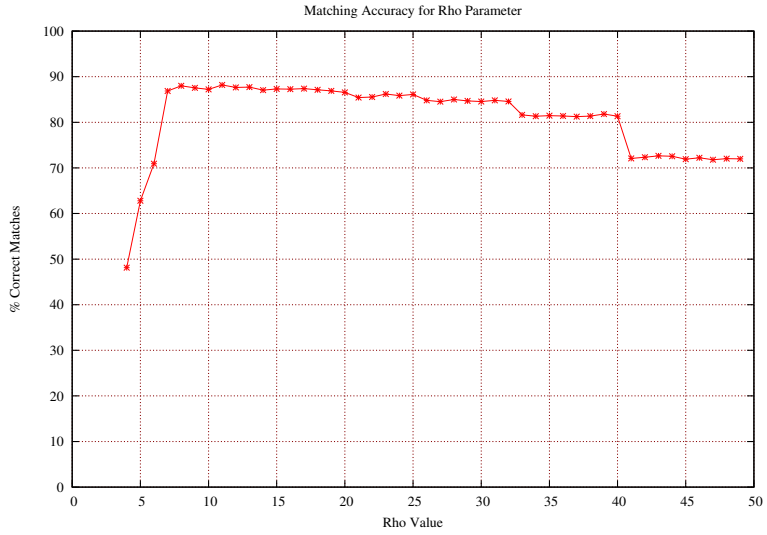

Figure 4.29: Matching accuracy depending on the number of rho samples.

Rho parameter computation times shows a clear linear behavior as depicted in Figure 4.30. As increasing the number of Rho samples not only increment linearly computation times but also may degenerate the final performance, it is important to set this value as low as possible, trying to get the best trade off between accuracy and computation time.

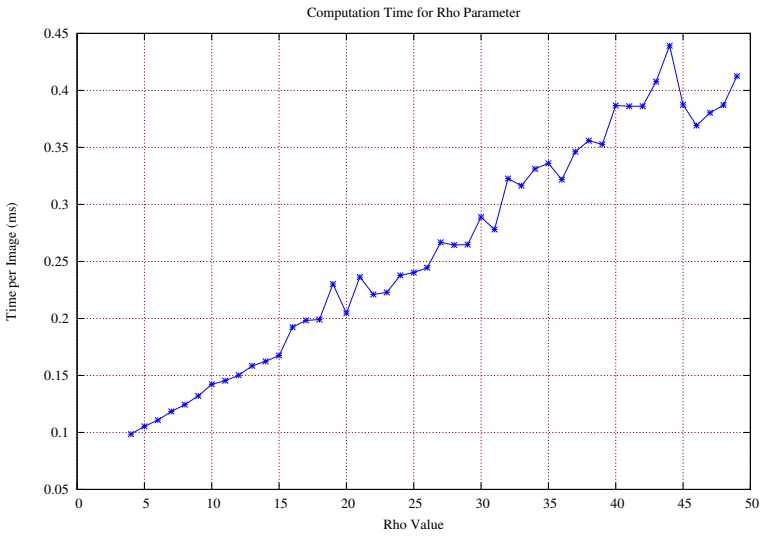

Figure 4.30: Computation time depending on the number of phi samples.

Phi and Rho Parameter

We also conducted an experiment where Phi and Rho, i.e. sampling parameters, were changed both simultaneously, and in the same quantity. Figure 4.31shows that convergence is reached about 15 to 20 and that the performance does not degenerate till both values are around 40. As described in Rho parameter experiment, when trace transform oversamples available data DITEC performance starts to decrease due to the generation of noise data in form of artificial patterns. Results depicted in Figure 4.31 also shows a big correlation between both parameters.

Figure 4.32 shows no linear trend between Phi,Rho parameters and computation time.

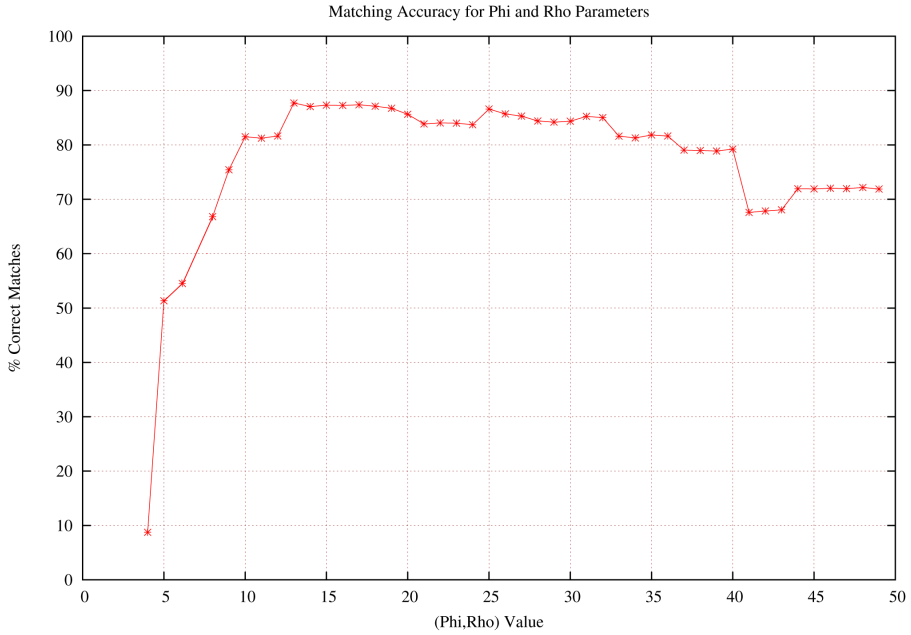

Figure 4.31: Matching accuracy depending on the number of phi and rho samples together.

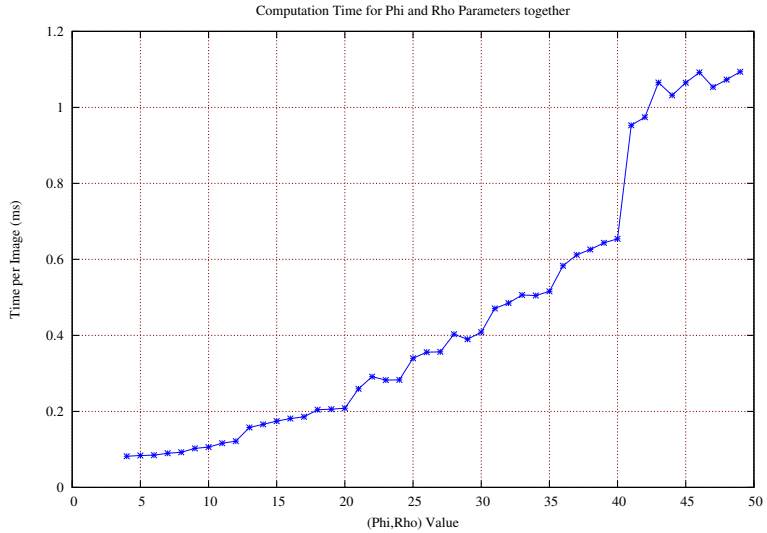

Figure 4.32: Computation time depending on the number of phi and samples together.

Patch size Parameter

In this test evaluated the performance of DITEC descriptor depending on image patch size. We therefore set all parameters fixed but changed square patch size from 12 to 34. All local descriptors found in the literature such as SIFT, SURF,BRIEF, etc used default patch size of around 16 to 26, so the range we evaluated sounds reasonable. Results depicted in Figure 4.33 shows that the best descriptors performance converges around of 20 pixels of image size.

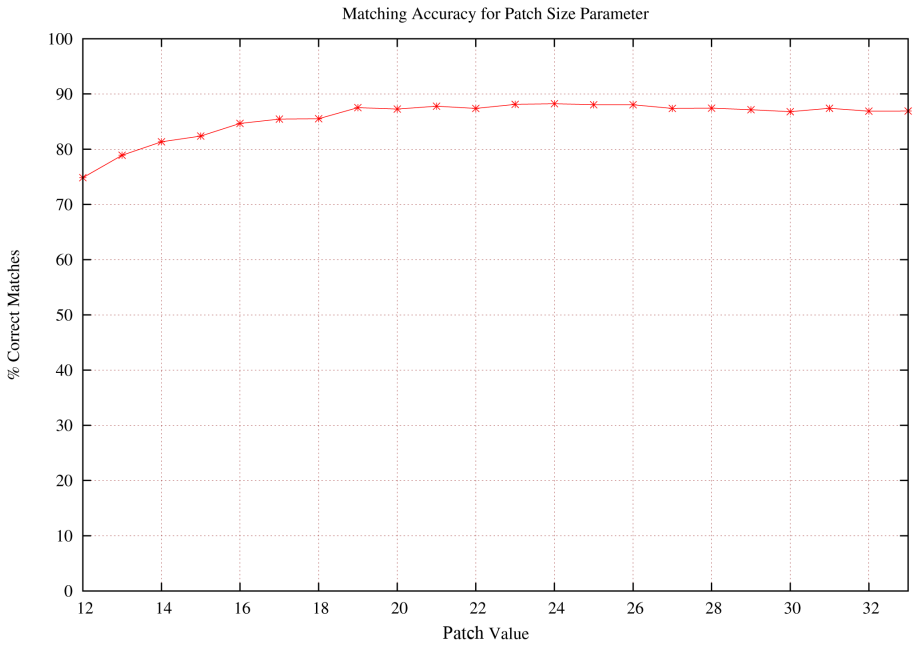

Figure 4.33: Matching accuracy depending on image patch size.

Computation times depicted in Figure 4.34 shows a perfect linear relation between image patch size and computation times. Because Phi and Rho parameters remain constant along the evaluation, the difference in time is due to the image patch extraction process itself around each interest point and due to the two image rotations needed to perform the trace transform.

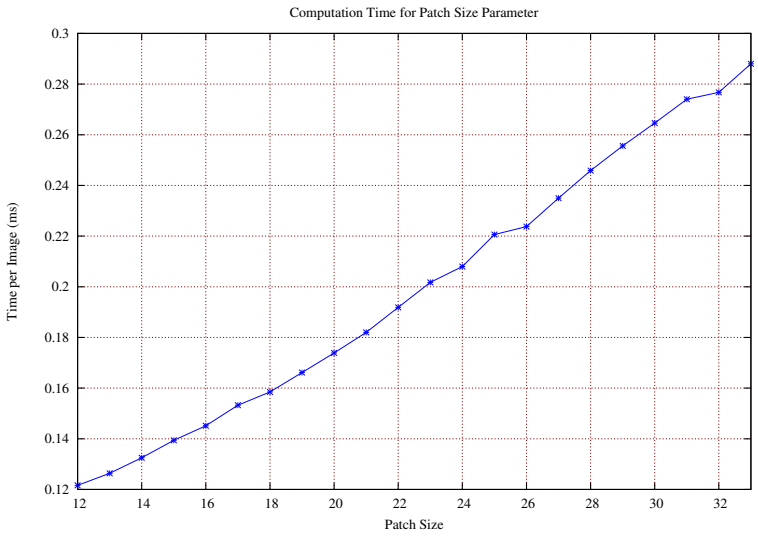

Figure 4.34: Computation time depending on patch size.

Descriptor dimensionality

Next test shows how different feature dimensionality influence in the performance of the descriptor. In this case we set a range of 20 to 250 dimensions. As shown in Figure 4.35 from values up to 30 dimensions the performance is very poor. Clearly, in this range the descriptor does not represent or retain enough discriminative power in order to be successfully matched against other descriptors. In 32 dimensions there is a clear gap in descriptor performance. Convergence is reached at the value of around 120 and continue increasing but very slow.

For computation time dependency of descriptor dimensionality, we measured the time needed for the descriptor matching process. As described in Section 4.3.4 the feature extraction process is nothing more than read a number of values of a matrix representing the module of the DFT of the trace transform, corresponding with the number of specified dimensions. The difference in time for reading linearly 20 to 250 different values from a matrix is almost negligible in current CPU-Memory architectures. As shown in Figure 4.36 there is almost no difference between different values of feature dimensions. The different peaks shown in the image are

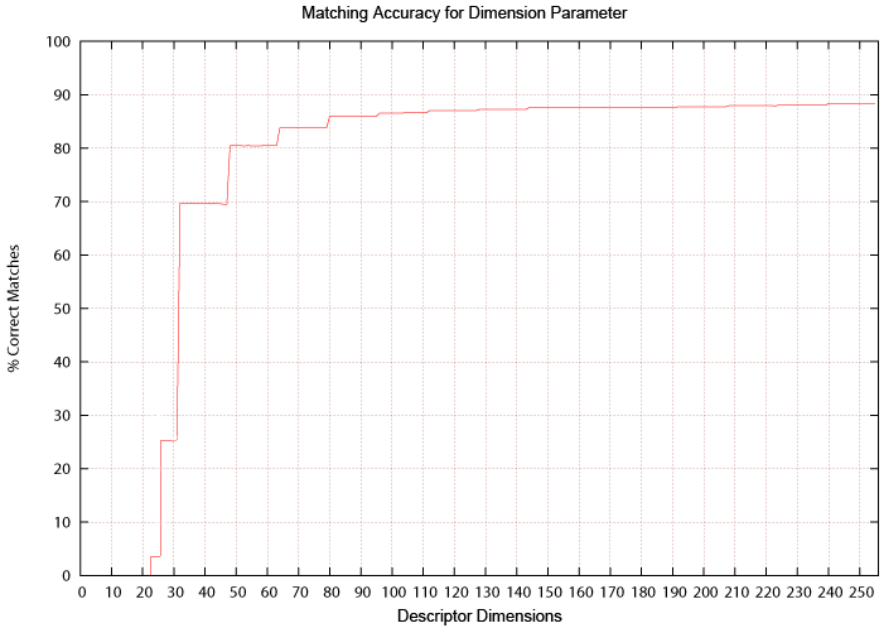

Figure 4.35: Matching accuracy depending on descriptor dimensionality.

due to non use of real time Operating System and therefore more processes are competing with the experiment thread.

Parameter sensitivity

One simple method for determining parameter sensitivity is to calculate the output model percentage of difference when varying one input parameter from its minimum value to its maximum value [99]. This percentage is known as the Sensitivity Index(SI), as is calculated using 4.18,

$$SI = \frac{D_{max} - D_{min}}{D_{max}} \tag{4.18}$$

where D_{max} and D_{min} represent the maximum and minimum output values, respectively, resulting from varying the input parameters over its entire range. Table 4.5 shows the Sensitivity Index for the evaluated parameters. In our case D_{max} and D_{min} represents the percentage of correct matches given the image data set and the input parameter values of local DITEC approach.

Clearly, the most influential parameter is represented by descriptor dimension-

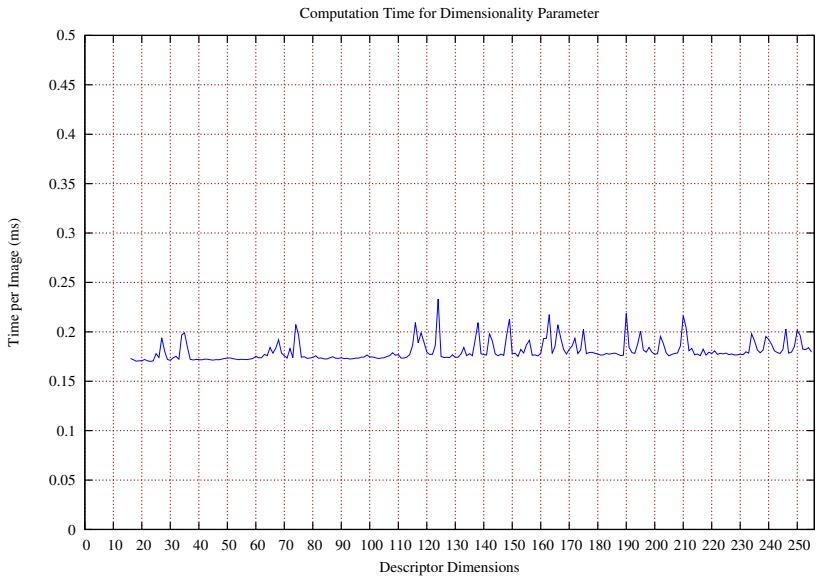

Figure 4.36: Computation time depending on descriptor dimensionality.

ality, followed by the number of samples in ρ and ϕ directions respectively, showing great correlation between them.

4.5 Feature descriptors

This section reviews some of the state-of-art local feature description approaches. In worth mentioning that are are continuously appearing new mechanisms for region description, thus the following list is not an exhaustive description or all existing approaches. However, it can be considered as a representative group of the most relevant or successful ones and those that have generated more variations, extensions or modifications since their publication.

SIFT

As described in Chapter 3 SIFT detector performs scale-space analysis that leads great performance regarding scale invariance ([100]). In addition, SIFT performs

Parameter	*SI*
Number of Samples in Phi	0.386
Number of Samples in Rho	0.331
Descriptor Dimensionality	**0.714**
Size of local image Patch	0.138

Table 4.5: Sensitivity Index (SI).

patch rectification in order to obtain robustness against in-plane transformation, along with an improvement of descriptor distinctiveness. For such rectification SIFT computes a dominant gradient orientation in the neighborhood of the interest point, by estimating a local histogram of 32 bin gradient directions computed at characteristic scale s, estimated during interest point extraction step, and accumulated over a window proportional to s. Gradient directions are weighted with the gradient magnitude and with a Gaussian window centered in the patch center x_c([67]), in order to improve orientation estimation in the presence of noise. Finally, dominant orientation is set as the peak of the histogram. However, SIFT also allows to assign multiple dominant orientations for histogram peaks within 80% of the highest peak, thus generating several descriptors for the same interest point.

After dominant orientation computation, a rectangular grid of $4x4$ sub-regions are defined oriented according with this orientation, as shown in Figure 4.37. Then, in each quadrant of the $4x4$ grid histograms of 8 bins of gradient orientations weighted by its magnitude are built. By joining the results, i.e. gradient orientations, of each quadrant yields a descriptor of $128\,(4x4x8)$ dimensions.

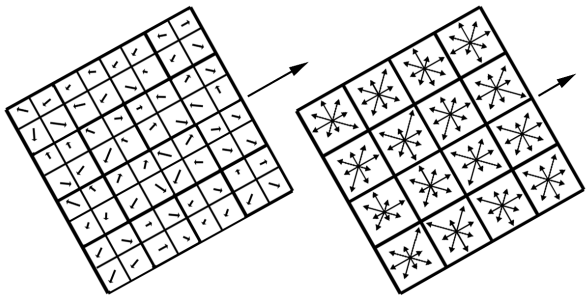

Figure 4.37: (left) Patch oriented along dominant orientation, (right) $4x4$ oriented grid with 8 gradient orientations each.

There are several variations of SIFT such as PCA-SIFT, where principal com-

ponent analysis is performed over SIFT descriptor in order to find correlation between dimensions, selecting only the most discriminant, hence reducing the original dimensionality of 128 values to a lower cardinality. In this way, posterior matching process is improved due to a reduction in time for computing descriptor distances. Another relevant variation is ASIFT [49]. This approach is focused on the improvement of original SIFT against severe affine transformation. Though SIFT is not by definition invariant to affine transformation, it shows a very good performance in moderate affine transformation. ASIFT improves even more SIFT robustness against affine transformation by computing several affinely transformed versions of input patches. During matching ASIFT compares each input descriptor with the whole set of transformed patches, hence improving matching ratio. A similar approach to ASIFT but using a trained classifier were proposed by [101, 95].

SURF

Similarly to SIFT, SURF forms a distribution-based descriptor. Descriptors generated by SURF are half the size of the original approach of SIFT. SURF also follows an strategy for in-plane rotation robustness very similar to SIFT, by estimating a dominant orientation. However, SURF employes first order Haar wavelets, instead of standard normalized derivatives. The first step in the descriptor generation involves assigning a reproducible orientation to each keypoint based on Haar-wavelet responses within a circle of radius $6s$ around the keypoint, where s represents the estimated scale for the interest point. The responses are computed at a sampling step of s, and each response is weighted by a Gaussian centered at the origin. The weighted responses are then interpreted as vectors and summed over a sliding window of width $\frac{\pi}{3}$ from 0 to 2π,as illustrated in Figure 4.38. The orientation of the largest resulting vector is assigned as the keypoint orientation.

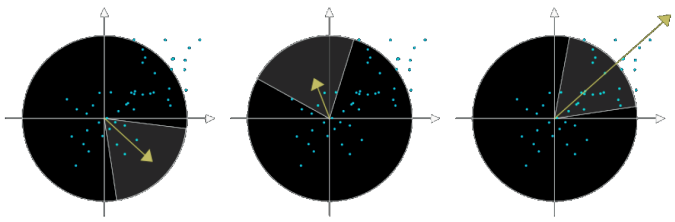

Figure 4.38: SURF Orientation Estimation..

BRIEF

This descriptor proposed by [78] can be seen as a extension or evolution of approaches like[101] where a tree classifier ensemble is trained for image patch matching process. BRIEF descriptor avoids the ensemble classifier training phase by directly building an string (descriptor) of 1's and 0's as results of comparing the intensities of pairs of points around a key point, extracted by a detector such as FAST [4]. This descriptor is oriented to have a good compromise between computational efficiency, in CPU and memory consumptions terms, and description capability. This descriptor is a binary bit string where each dimension of the vector is a 1 or 0 depending on the results of several tests distributed along image patch of the form:

$$\tau(p;x,y) = \begin{cases} 1 & p(x) < p(y) \\ 0 & p(x) \geq p(y) \end{cases} \tag{4.19}$$

where $p(x)$ represents the intensity value of the pixel point x. In [78] the authors proposed several sampling strategies for generating the set of tests T that will form the final descriptor. Different sampling strategies does not show much differences in performance. The authors propose to use a pixel sampling based on an isotropic Gaussian distribution like $G = \left(0, \frac{1}{25}S^2\right)$centered in the image patch of size SxS. The feature descriptor is finally generated by joining the n responses, i.e. 1's or 0's, usually 64 to 128, of simple tests defined in Equation 4.19. Taking only the information at single pixels into account is very noise-sensitive, so prior to pixel differencing a Gaussian smoothing is applied to the patch in order to increase stability and repeatability in presence of noise.

BRISK

BRISK descriptor was proposed by Leutenegger et al. in [50]. This descriptor as FREAK or ORB is based on binary data only. This approach is very similar to BRIEF in the sense that proposes to form descriptor dimensions based on the responses of simple tests by comparing pixel intensities, as in Equation 4.19. One of the key differences with BRIEF is that BRISK proposes to make simple pixel tests in an specific sampling pattern. More precisely, BRISK samples pixels on circles concentric with the interest point, as shown in Figure 4.39. This sampling pattern is very similar to the one used in DAISY descriptor [102]. Concentric

circles are computed within a neighborhood region of size relative to characteristic scale s, computed during interest point extraction using AGaST detector. This pixel sampling retains more information than BRIEF because of characteristic scale s allows BRISK to sample pixels further or closer depending on s. In addition, before pixel tests computation BRISK proposes to rectify image region around interest point according to dominant orientation, estimated during point extraction.

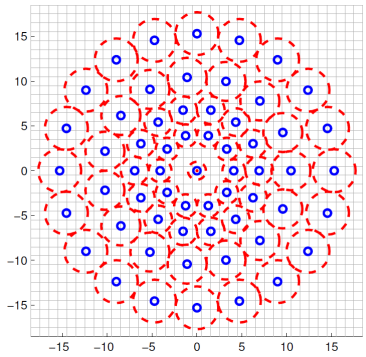

Figure 4.39: BRISK local sampling pattern.

FREAK

FREAK (Fast Retina Keypoint) descriptor proposed in [6] extends the works of [50] on the BRISK descriptor. The authors proposed, as BRISK or BRIEF, a descriptor as a bit-stream, but proposes a new sampling pattern inspired by the human retina. The first contribution is an allocation of concentric distribution of size exponentially increasing with the distance to the interest point. The second contribution is to choose a pattern that creates overlaps between the different concentric discs, as shown in Figure 4.40. Overlapping between sampling regions adds redundancy that increases final descriptor discriminative power. The authors argue that this redundancy is also present in receptive fields of the human retina [103].

Sampling pattern depicted in Figure 4.40 descriptor contains 43 sampling regions or receptive fields, which leads to 903 possible binary tests (see Equation 4.19). Receptive fields are weighted with Gaussian filters according with estimated characteristic scale s and For efficiency purposes and to avoid having too much correlation between pairs, the authors propose a method for selecting a subset of 512 pairs, retaining the most informative and discriminant tests.

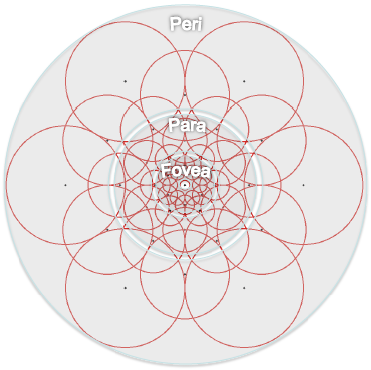

Figure 4.40: FREAK sampling strategy (image extracted from [6]).

4.6 Experiments and Results

In this section we show an evaluation carried out comparing the most relevant feature descriptor in the state-of-the-art. We included in the evaluation our approach based on the Trace transform. We evaluated the behavior of different approaches, mainly in relation with their robustness against geometric and photometric transformations. We used the our experimental setup and evaluation framework described in Appendix A.

A similar study can be found in [104]. In addition to [104], we conducted several experiments evaluating the performance of recently published feature descriptor algorithms, such as BRIEF or ORB. In opposite to that study, we did not include in our evaluation approaches based on classifier ensembles such as [95], because their performance mainly relies on several parameters specific for their respective training steps. These parameters are too specific and not applicable to the non-machine learning oriented algorithms.

Similarly to the study carried in Chapter 3 we evaluated the performance of several description approaches and DITEC, against several geometric and photometric transformation. In this case, performance is evaluated as the ratio between wrong and correct matches,i.e. accuracy, between several real and generated images, given the ground truth data. Because depending on the content of the images the number of interest points detected can be very different from image to image,

we perform accuracy normalization given the number of detections in each test. As the experiments in Chapter 3 we used pair of images where geometric transformation can be represented by a 2D homography.

In worth noticing that results obtained by every approach may change depending on the settings of their parametrization. We set those parameters as default values, as suggested by their corresponding authors, described in their publications.

4.6.1 Geometric Transformations

Figure 4.41 shows the results obtained in the evaluation of in-plane rotation geometric transformation of an input image. As can be seen, two approaches BRIEF and DAISY are not robust to rotation, degrading its performance when rotation values is over 30º. DAISY approach does not perform local gradients calculation, oriented to any dominant orientation, i.e. the descriptor is not rotationally normalized, thus severe in-plane rotation transformation clearly degrades matching accuracy. However, in worth mentioning that there exist a variation of DAISY called O-DAISY that is invariant to rotation, but it must be implemented in FPGA for reducing computational cost [105]. BRIEF binary descriptor is based on the comparison of several points around an interest point p. The order of such comparisons defines BRIEF descriptor. That ordering is not adapted like other descriptor like SIFT does, hence any rotation transformation applied to image generates completely different descriptors thus matching is unfeasible. ORB shows a clear benefit respect to its non-rotation invariant version,i.e. BRIEF, by steering BRIEF descriptor along dominant orientation computed using central moments.

Freak and BRISK show great performance, being stable along the rotation transformation range. FREAK improves BRISK performance because of using overlapping receptive fields (see Section 4.5) instead of isolated regions for sampling, thus resulting in a more discriminant descriptor, taking into account that both uses the same approach for scale estimation. The best performance is obtained by DITEC descriptor obtaining a mean around 90%, but when rotation is at 180º that decreases to 85%. It worth mentioning that contrary to the rest of rotation invariant approaches, DITEC is invariant to rotation transformation but having no dependency with the dominant orientation estimated by the interest point detector. Hence, in case of using DITEC the computational burden for orientation estimation could be removed from the detector.

SURF shows non-stable results along transformation range, showing poor results at angles modulo 45º. In SURF detector, dominant orientation is computed by using sums of Haar wavelet filter responses that clearly shows quantization effects at 45º angles due to its discretization. SIFT however computes orientation by using scale-normalized derivatives that clearly represents more accurately the neighborhood gradients of an interest point than Haar approximation. .

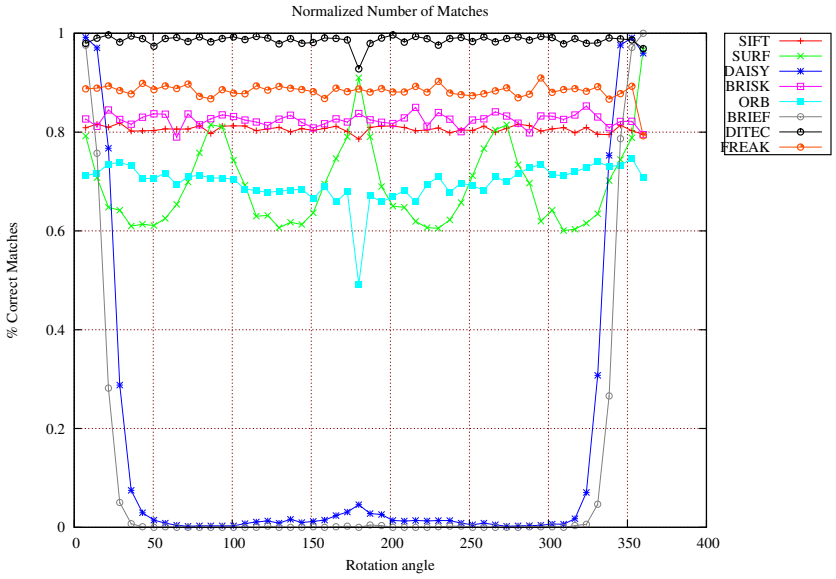

Figure 4.41: In-plane Rotation Transformation matching results.

Figure 4.42 shows the results obtained in the evaluation of scale transformation of an input image. It worth mentioning that the worst results are obtained by BRIEF as expected. As described previously this descriptor depends on original implementation of FAST that does not perform any scale estimation. In this way, when transformation value differs more than ± 0.5 from the original scale, the descriptor does not generate any correct match. However, when scale value is around original scale, BRIEF descriptor performs almost perfectly what shows the high discriminative power contained in binary BRIEF strings. BRISK shows also very good results compared with its most similar approach BRIEF, due to characteristic

scale estimation using AGAST. SIFT shows the most stable results, being almost insensitive to transformation value.

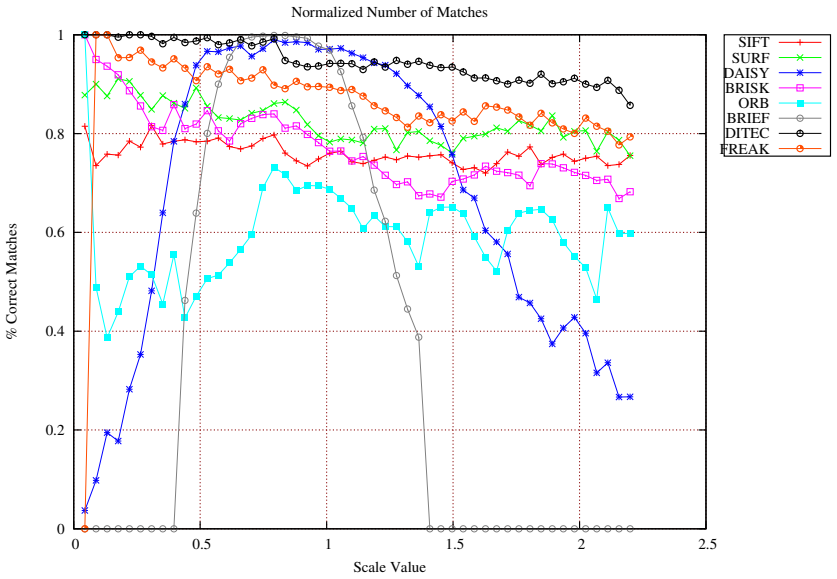

Figure 4.42: Scale Transformation matching results.

ORB shows a clear improvement over its non-invariant scale version BRIEF, but even that the results are stable along the transformation range the results are poor, given around of 55% mean accuracy. We think that this performance is due to a non-existence of a truly characteristic scale estimation process. Instead, they perform FAST point extraction in an image pyramid separated $\sqrt{2}$ and apply HARRIS cornerness measurement for non-maxima suppression.

FREAK nor DITEC do not perform characteristic scale estimation, nor dominant orientation because they are only limited to be region descriptors. Therefore, we employ characteristic scale estimation given by SIFT in both cases. As Figure 4.42 depicts even that both DITEC and SIFT share the same estimation for characteristic scale all points detected in the data set, DITEC shows the same accuracy stability than SIFT but at approximately 10% higher ratio. We think that this is due to the higher discriminative power of DITEC descriptor that retains more informa-

tion, compared with the bin oriented, i.e. histogram discretization, approaches such as SIFT or SURF. FREAK also outperforms SIFT using the same scale estimation. Oriented receptive fields proposes by FREAK seems to retain a lot of discriminant information thanks to redundancy. It worth noticing that both FREAK and BRISK uses the same characteristic scale estimator integrated in Agast detector. However, FREAK outperforms BRISK regarding scale invariance, hence FREAK descriptor is less sensitive to scale changes verlapping regions instead. Overall, DITEC shows the best performance regarding robustness to scale transformation.

Figure 4.43 shows the results over the first four images of Graffiti data set ([22]). In this test mainly robustness against perspective transformation are measured. As described in Chapter 2 perspective or projective distortions can be locally approximated by an affinity, assuming that scene can be locally plane. Hence in this study we also evaluate the behavior of descriptors against affine transformation. When perspective transformation is not severe DITEC shows overall the best performance. When transformation is more severe, as in image 4 of the data set, SIFT and FREAK obtains the best results. It worth mentioning that even SIFT is not by definition invariant to affine transformation, it shows great stability in such situation. Local gradient histogram orientations retains a lot of information that is preserve even in moderate affine deformation. SURF, though it is also based on the computation of local gradients shows worst results compared with SIFT. Box-filtering used in SURF are less accurate than Laplacian of Gaussian during estimating orientations, being box-filter more sensitive to severe affine or projective transformations. As expected BRIEF descriptor shows the worst performance. Graffiti data set requires robustness to rotation transformation that BRIEF does not provide. Moreover, clearly not estimate a characteristic scale has a negative impact when affine or projective transformation is present. Perspective distortion has somehow a component of scale transformation, more noticeable in structures that are farther away from the camera viewpoint.

4.6.2 Photometric Transformations

In addition to geometric transformation, we evaluated the behavior of state-of-the-art descriptor against photometric transformation. Figure 4.44 shows the results obtained in the evaluation of the exposure change data set, described in A.

In this test clearly DAISY and BRIEF are clearly superior to the rest of descriptors. Both approaches heavily rely on local difference of pixels values. Because no

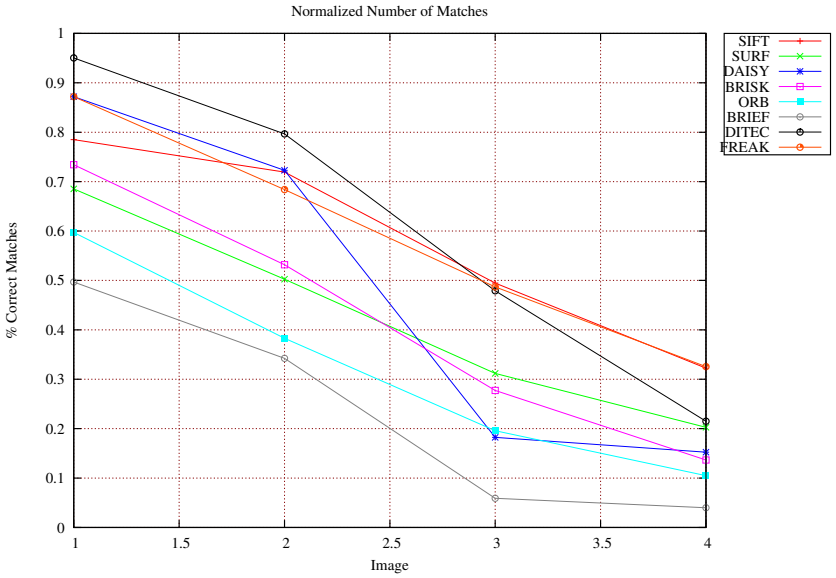

Figure 4.43: Projective Transformation matching results.

geometric transformation is applied to the images forming the data set, BRIEF does not suffer for its lack of invariance to rotation or scale. In the rest of descriptors the change in photometric conditions has a negative impact in matching accuracy, nevertheless all of them shows very stable results. It worth noticing that BRISK descriptor obtains the worst results starting from 3.5 f-stops of light reduction, because of their respective point detector, i.e. Agast, is not able to cope with such a change in light intensity, hence no detection is generated.

SIFT and SURF are both very robust to intensity light changes. Clearly, the use gradient orientations, i.e. image derivatives, reduce the influence of photometric changes. It worth noticing that also DITEC approach is very robust against exposure changes. As described in Section 4.3 trace transform is sensitive to affine light changes, if using some trace functionals such as *IF1*. In this way, we included in our processing pipeline a light and contrast normalization, by normalizing the descriptor vector, as suggested by Lowe in [46]. We also remove from the descriptor, the first component of each row of the DFT. This component represents the mean

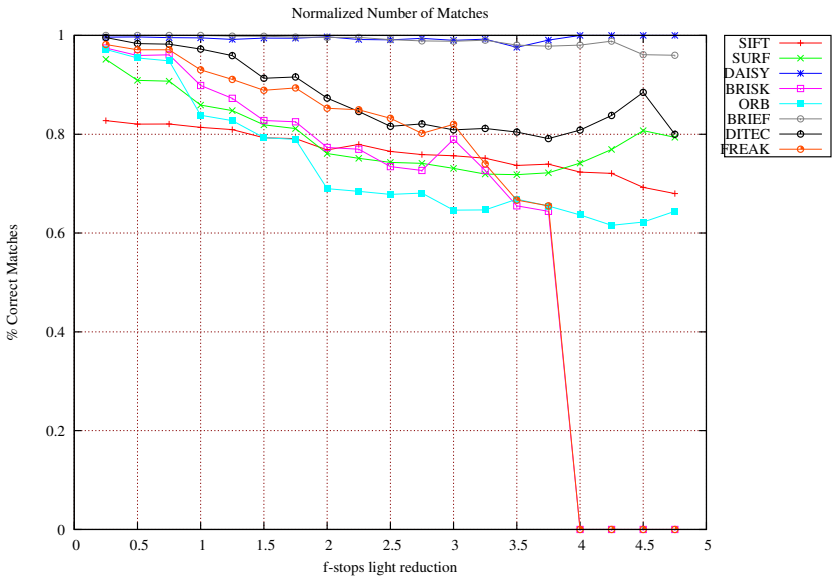

Figure 4.44: Exposure change photometric transformation matching results.

value of the signal, thus removing it we get the value of the signal independently of its intensity, thus obtaining robustness against light intensity variations. We also evaluated a Trace functional, *IF3* in table 4.1, that uses only image gradients. By using only derivatives we are removing or ignoring intensity information, such as SIFT ([31]) does. Therefore, robustness or invariance to affine light, intensity or exposure transformation can be achieved. However, using this functional as the trace functional for DITEC feature extraction we detected very poor results in geometric transformations tests compared with other functionals such as *IF1* or *IF2*. We concluded that this behavior is due to the reduction or loose of information by eliminating intensity and contrast data contained in image patches due to the use of derivatives. Trace transformation using *IF3* does not retain enough information when applied to local, i.e. small, image regions. This lack of information causes DITEC descriptor using *IF3* not to be able to cope with severe geometric transformations such as 180º angle of rotation, or severe perspective distortion due to point of view.

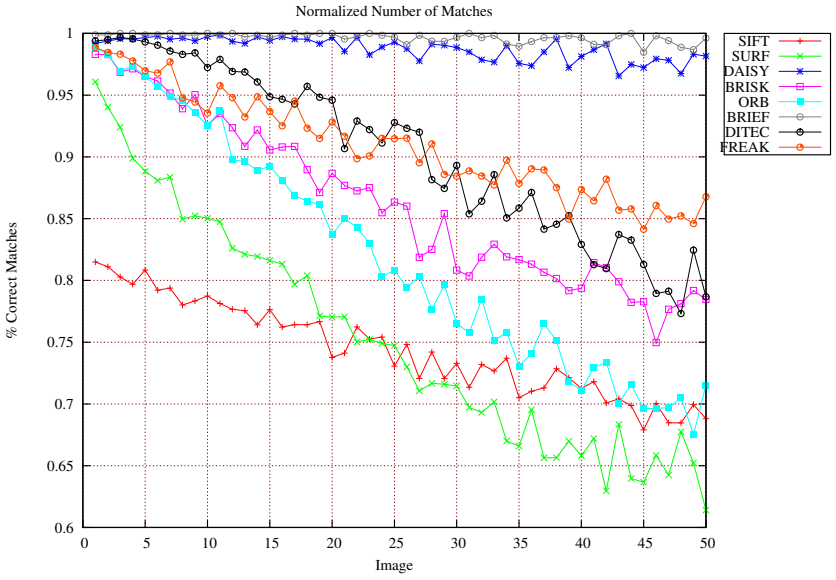

Figure 4.45: Noise (SNR) change photometric Transformation matching results.

Similarly to results obtained in affine exposure change test, DAISY and BRIEF show the best results. BRIEF descriptor is based on relative difference of intensity of several pixels around an interest point. The addition of luminance noise is uniform randomly distributed along image space does not affect to approaches based pixel intensities distribution. In opposite, approaches based on first or second order derivatives such as SIFT or SURF suffers with the addition of luminance noise. Sensitivity to noise is more noticeable in case of SURF because of the use of second order derivatives.

4.6.3 Computation Time

In addition to the evaluation of the robustness of described feature descriptors against geometric and photometric transformation, we measured computation times. More precisely, we measured separately the time each approach need for extracting each descriptor, given an input image and a set of interest points, as well as the time needed for matching. As described in Chapter 2 depending on the nature of fea-

ture descriptors they generate vectors of different size, i.e. dimensions, hence the time needed for matching them is directly proportional with their dimensionality. It worth mentioning that computation times are very dependent on the algorithm implementation and the parameter settings of each approach. As mentioned before, we set parametrization as suggested by the authors and we used Open Source implementations of algorithms that may differ from their original non-free sources, such as the cases of SIFT or SURF. We measured time by incorporating both feature descriptor generation as well as matching process. We considered interest point extraction a different task, that can be performed apart from feature description, however matching is directly related with the nature of the descriptor, hence must be also measured along with the time needed for descriptor generation.

Table 4.6 shows average timing results of extracting and matching one single descriptor of every different approach, measured in milliseconds. Results were averaged out after executing 10 runs of 1000 descriptors generation and matching on a 2Ghz Quadcore CPU, and non real-time Operating System. Times are shown in Table 4.6 Clearly, the slowest approach is DAISY. This descriptor was originally proposed for dense depth map estimation where instead of computing a set of sparse interest points every pixel is taken into consideration. In this way, though we apply DAISY descriptor computation to a set of previously extracted interest points, DAISY algorithm needs to perform descriptor computation at every pixel of input images before descriptor selection corresponding with the input interest points. This mechanism introduces a computational burden making DAISY descriptor not suitable for real-time operation. The next slowest approaches are SIFT and DITEC. Both approaches show very similar results. It worth mentioning that current implementation of DITEC is not optimized while SIFT implementation uses several techniques for faster computation. DITEC approach spends most of the time computing the trace transform matrix. This process can be clearly optimized by parallelizing the computation of each trace projection, hence a complete parallel implementation of DITEC is possible. SURF descriptor follows a very similar approach compared with SIFT but being less computationally demanding, due to the use of integral images and box filtering for approximating the Laplacian. Clearly BRIEF is currently the fastest feature descriptor approach. This descriptor is implemented by using low level CPU instructions, is highly optimized and uses only difference operations over integer values.

Descriptor	Extraction (ms)	Extraction+Matching (ms)
DAISY	0,86	0,89
SIFT	0,113	0,174
DITEC	0,112	0,173
SURF	0,06	0,11
FREAK	0,05	0,099
ORB	0,017	0,044
BRISK	0,015	0,069
BRIEF	0,008	0,036

Table 4.6: Descriptors computation Time.

4.7 Discussion and conclusions

In this chapter we have described a new approach of local image or feature descriptor. This descriptor is based on the generalization of the trace transform. As shown in the evaluation section it shows very good results against rotation transformation given the best results. It worth mentioning that DITEC descriptor does not depend on a previous estimation of dominant orientation, carried out by the interest point extraction approach, in order to be robust to rotation transformation, thus allow to reduce the computational burden of the interest point extractor.

SURF was proposed as an alternative to SIFT but computationally less demanding. Our results shows that SURF is about approximately two times faster than SIFT, however SURF shows more unstable results along the evaluation compared with SIFT being very sensitive to some transformations such as rotation. BRIEF is the less CPU demanding descriptor. This descriptor is suitable for applications where computational resources or real-time operation does matter such as SLAM, where reducing the computation burden is preferable than having a completely robust feature descriptor. By using less demanding approaches, allows the imaging pipeline to apply or impose more filters or restrictions in order to limit the influence of outliers or wrong correspondences.

FREAK descriptors seems to retain a lot of discriminant information thanks to redundancy generated by overlapping receptive fields. FREAK uses the same scale estimator than BRISK and also follows a very similar approach. The main difference relays on overlap regions while computing local gradients. Overlapping regions can be also found in DAISY, however this descriptor does not perform orientation correction thus is very sensitive to rotation transformation. The gen-

eration of redundancy can also be found in DITEC, where the generation of trace functionals along $0-2\pi$ range cause visiting same pixels many times at different orientations. We think that this redundancy along with FFT gives DITEC descriptor its discriminative power.

In general, binary descriptors are computationally less demanding than real valued descriptors due to more efficient matching processes. Compare or compute the distance between two binary descriptors can be carried out very efficiently by computing the Hamming distance. This operation corresponds to a simple bit count on the result of a binary XOR operation between two vectors. This bit count becomes particularly efficient on CPUs supporting instructions such as the POPCNT ([106]).

Chapter 5

Machine Learning based descriptor matching

In this chapter we proposed a mechanism based on Machine Learning techniques for feature descriptor matching, applied on an Augmented Reality scenario. This chapter is structured as follows: Section 5.1 gives a brief introduction to Augmented Reality technology, Section 5.2 presents the computational methods, specifically the tracking by detection. Section 5.3 introduces the Random Forest. Section 5.4 gives our computational results on benchmark images. FInally, Section 5.5 provides come conclusions.

5.1 Real-time Optical Markerless Tracking for Augmented Reality

The term Augmented Reality (AR) refers to a technology that allows to add virtual information to the scene seen by the user. In contrast to Virtual Reality where the entire environment is completely virtual, AR combines both virtual and real objects in the same scene. Therefore, while Virtual Reality substitutes the reality, AR enhances it. It is furthermore important to distinguish between AR and the special effects of the film industry or TV production, where some virtual characters or virtual objects appear perfectly integrated within real objects. The main difference is that AR is meant to be used in real-time, while special effects production can be processed off-line allowing the use of sophisticated techniques which can be computationally expensive.

Most of the computational costs are due to the tracking process, in order to align properly the real and virtual objects with respect to each other and to produce a realistic illusion of fusion between the two worlds. The eyesight is one of the most important senses for the perception of the human being. Hence, any discrepancy between real and virtual object would be automatically detected by the human's eye and the AR effect would be missed. Despite the rapid development of computational power and specialized hardware such as programmable graphics units (GPUs), tracking technology still suffers from a notorious lack of robustness and high computational costs. These drawbacks get drastically worse in an uncontrolled context such as outdoor, where it is difficult to calibrate the environment, add landmarks, control lighting and limit the operating range to facilitate tracking. In this paper we address the tracking problem for AR applications in uncontrolled environments.

While a large variety of tracking systems are commercially available (mechanical, acoustic, magnetic, inertial and optical sensors), most of those systems are meant to be used in perfectly known contexts, where the variables that affect the tracking can be controlled easily. In uncontrolled environment, the tracking process should work without adapting the object or the environment to be tracked, such as placing special landmarks or references. This issue is known as markerless tracking. Optical sensors have been recently widely explored to answer markerless tracking [107].

Optical markerless tracking uses natural features such as edges, corners or texture patches, extracted from the images acquired by a camera. By using natural features, the use of artifacts such as reflective markers is avoided, allowing the system to be more flexible and being able to work in non-well controlled conditions. In our approach we use the fact that natural plane surfaces are common structures either in an indoor or outdoor scenario. The ground, the building facades or walls can be seen as planes. Therefore we propose to focus our work on optical markerless tracking for planar structures in unprepared environments.

Figure 5.1 shows a typical design of an AR application based on optical tracking. A camera is capturing images from the world (environment). These images are transferred to a computing workstation where they are processed to extract useful information, such as the camera pose transformation.

The term camera pose refers to the transformation, translation and orientation, between the objects or environment coordinate system and the camera coordinate system, as depicted in Figure 5.2.

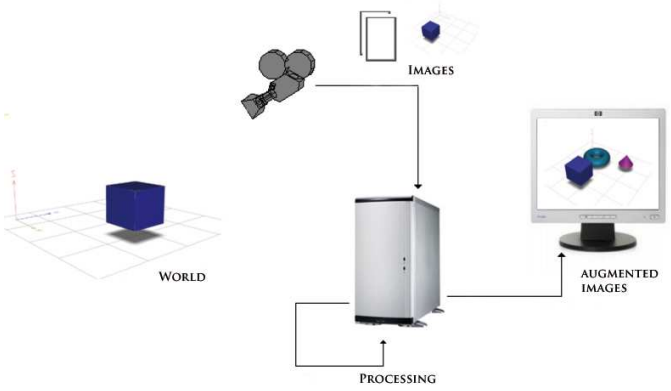

Figure 5.1: Schema of an augmented reality application.

This transformation should be estimated dynamically, as fast as possible, in order to realistically integrate virtual objects between real ones, during the tracking sequence. This estimation must be very accurate so that virtual objects appear rigidly fixed to the real world. If this transformation is inaccurate, the objets will not appear correctly aligned, as shown in Figure 5.3. Depending on the quality of the images, or the user motion, the pose estimation might be difficult to solve and the tracking process might fail. In these cases, some user interaction, such as the manual input of some specific points, or initial camera pose estimation could be required to help the system to compute the next camera values.

The requirements stated in this section pay special attention to the software features as well as its integration with the hardware. Key features of software are usability of the user interface, efficiency, real-time performance and robustness; the most important aspects related to hardware are portability, reliability, and costs.

5.1.1 Robustness

The tracking must be achieved with a minimum level of robustness, without failing, or continuously re-initializing the tracking process. Besides tracking loss, some other problems may appear such as drift or jitter. The drift problem refers to the displacement of the origin of the world coordinate system. When recursive techniques are applied to estimate the camera pose, some error accumulation over time may occur. This error accumulation causes the impression that the virtual objects are floating between real ones. The jitter problem is caused by small variation

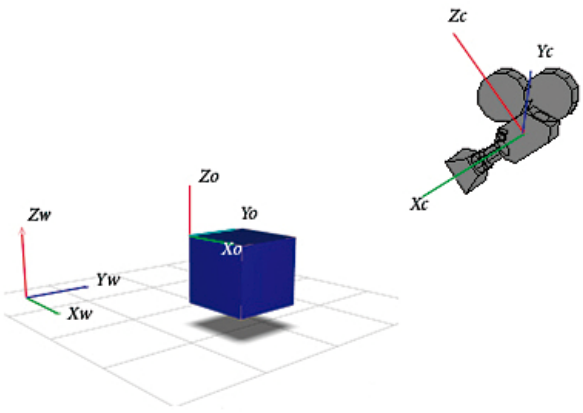

Figure 5.2: World, object and camera coordinate systems.

Figure 5.3: (left)Wrong camera pose estimation, (right) Correct camera pose estimation, the virtual object appears correctly aligned with real world.

in the camera pose transformation between frames, even when there is no variation in neither the objects nor the camera. This difference causes the effect that virtual objects appear flickering in the images, not being rigidly fixed in the real world. Such effects should be avoided as much as possible.

The tracking process must be fast, minimizing the time needed for the pose computation, so that achieve near real time system performance is achieved. Any delay between image acquisition and final image generation will deteriorate the effect of integration of virtual and real objects. In some cases, the rendering of the final image can be synchronized with the virtual object generation, for example when using a video see through head mounted display. However, with this approach the frame rate could be lower than real-time. If the generated images are highly delayed, the user will perceive the difference between the physical stimulus

of its movements and the visual stimulus, making the AR application uncomfortable. Such delay should be reduced as much as possible.

5.2 Methods

Our approach to obtain the camera pose is based on the tracking of plane surfaces. The 3D world planes (ground, building facades, walls...) and their projection in the image are related by a plane to plane projective transformation,also known as homography or collineation. It can be modeled as a $3x3$ matrix H with eight degrees of freedom. The camera pose can be recovered by estimating this homography between a world plane and its image. This estimation can be carried out by tracking points lying on the world plane, and matching them frame by frame.

For optical markerless tracking two main groups can be distinguished: recursive and tracking by detection techniques. Recursive techniques start the tracking process using an initial guess or a rough estimation, and then refine or update it over time. They are called recursive because they use the previous estimation for calculate the next one. Contrary, tracking by detection techniques can do a frame by frame computation independently from previous estimations. In this case, some a priori information about the environment or the objects to be tracked is needed.

We have worked on the camera pose estimation problem using two different approaches. The first one is based on recursive tracking and the second one based on tracking by detection method. The latter requires the implementation of a keypoint classifier, which directly impacts on the tracking performance. In the following, we present these methods in detail.

5.2.1 Camera Pose Estimation

As described in section5.1 this problems tries to find the geometric transformation between two coordinate systems, more precisely, between the world coordinate system and the camera coordinate system. When this transformation is obtained, through point correspondences for example, a virtual camera can be transformed accordingly, and so the virtual objects can be accurately aligned in the images. In Chapter 2 a description about how world points are related to image point, through a camera model is given.

If we choose that the Z world coordinate equals zero for all points M of the world plane π, we obtain:

$$m = PM = [p_1 p_2 p_3 p_4] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = [p_1 p_2 p_4] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5.1}$$

where each p_i represents a column vector of the matrix P (see Chapter 2). Then the mapping between points on the world plane $M_\pi = (X,Y,1)^t$and their image m is a planar homography $m = HM_\pi$, where $H = [p_1 p_2 p_4]$. This expression can be written as follows:

$$H = K[r_1, r_2, t] \tag{5.2}$$

where r_i are the columns of the rotation Matrix R and t the translation vector.

For the estimation of the homography H, it is needed to find some point correspondences in both planes $M_{\pi i} \Leftrightarrow m_i$, where $M_{\pi i}$ is the point in the world plane π and m_i is its corresponding point in the image. In the context of markerless tracking, these points are known as natural features. The expression $m = HM_\pi$can be rewritten as:

$$mxHM_\pi = 0 \tag{5.3}$$

Each correspondence $M_{\pi i} \Leftrightarrow m_i$ gives rise to two linearly independent equations in the entries of H. As explained in Chapter 2 an homography has eight degrees of freedom, thus only four coplanar non-collinear points are needed in order to estimate it. Also, in Chapter 2 a linear method for obtaining homography transformation from more that four points is given.

As mentioned before, a calibrated camera is represented as $P = KRt$. Once H and the internal camera parameters K are known, the camera pose Rt can be recovered from equation 5.2, as:

$$K^{-1}H = (R^1 R^2 t) \tag{5.4}$$

where K^{-1} is the inverse of the internal camera parameters matrix, t is the translation vector, $r_1 r_2$ are the two columns of the camera rotation matrix. The third column of the rotation matrix R^3 can be obtained by the cross-product of r_1 and r_2.

Since we are using the internal camera parameters for the camera pose estima-

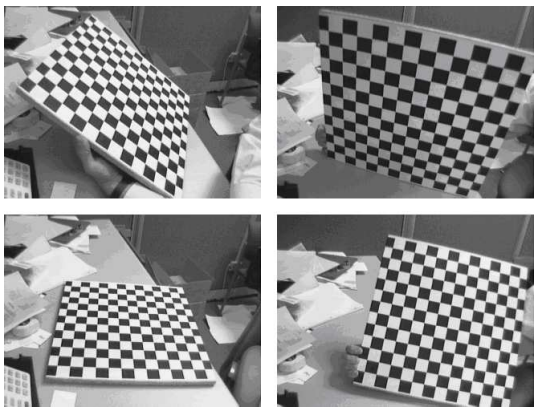

Figure 5.4: Images of a calibration pattern taken with different orientations and scales.

tion, the camera must be calibrated beforehand. This calibration task is carried out by taking several images of a calibration pattern, for example a picture with white and black squares from different distances and points of view as shown in Figure 5.4 [26].

5.2.2 Recursive Tracking

Recursive tracking techniques start the tracking process from an initial guess or a rough estimation, and then refine or update it over time. They are called recursive because they use the previous estimation to propagate or calculate the next estimation. In a recursive approach scenario, the initialization of the tracking process must be carried out by a mechanisms such as selecting manually four coplanar points lying on the same plane in the 3D world placed in a non-degenerated configuration, or by automatically detecting a maker or a known planar structure ([108]). Once these four points are available, the first homography estimation takes place starting the tracking process. During the tracking process, this homography is continuously updated or propagated by extracting points from images, using any interest point extractor such as the Harris operator [20], matching them between previous and current images and calculating a new homography.

When only four points are used to estimate the homography, it is said that a minimal solution is obtained. In this context, minimal means that any error generated in the location of any of the four points will degenerate the estimation. Depending on the generated error, the estimated homography can be completely

distorted. For this reason the homography estimation is typically performed using more than four points correspondences. A point correspondence is considered an outlier if it is generated from another plane or if it is a wrong correspondence (mismatch) between points of the same plane. Algorithms using random sampling shown in Appendix B are well-known methods for robust estimation even in the presence of outliers. Those algorithms, applied to homography estimation search randomly a combination of four points from the available candidates and estimate a transformation. The estimated transformation with the minimal solution is then tested with the rest of points. The transformation that obtains more support (number of inlier points) after some fixed number of iterations is selected as the best one.

During the estimation process several errors may occur, such as point missmatching due to severe changes photometric conditions such as illumination conditions or fast camera movements generating motion blur, thus generating a drastic reduction in the number of interest points detected, as shown in evaluation of Chapter 3. Due to the recursive nature of this kind of tracking, recursive tracking is highly prone to error accumulation. The error accumulation over time may induce a tracking failure, requiring a re-initialization of the tracking process, which can be cumbersome and not feasible in practical applications.

5.2.3 Tracking by Detection

Other approaches are known as tracking by detection. In this kind of techniques some information of the environment or the object to be tracked is known a priori. They are also known as model-based tracking because the identification of some features in the images (texture patches or corners) corresponding to a known model are used to recognize such objects.

This kind of tracking does not suffer from error accumulation because, generally, does not rely on the past. Furthermore, these methods are able to recover from a tracking failure since they are based on a frame by frame estimation. They can handle problems such as matching errors or partial occlusion, being able to recover from tracking failure without intervention [109].

Tracking by detection needs data about the objects to be tracked prior to the tracking process itself. This data can be in form of a list of 3D edges (CAD model) [110], color features, texture patches or point descriptors [46, 2]. Then the tracker is trained with this a priori data, to be able to recognize the object from different

points of view. A good survey about different model-based tracking approaches can be found in [107, 111].

Some authors propose the use of machine learning techniques to solve the problem of wide baseline keypoint matching [112, 113]. Supervised classification systems requires a pre-processing, where a system is trained with a determined set of known examples (training set) that represents variations in all their independent variables. Once the system is trained, it is ready to classify new examples. Some of the most widely used supervised classifiers are for example, k-Nearest Neighbors, Support Vector Machine or decision trees. While k-Nearest Neighbors or Support Vector Machine can achieve good classification results, they are still too slow and therefore not suitable for real-time operation [114].

Recently the approach based on decision trees has been successfully applied to tracking by detection during feature point matching task, by training the classifier to establish correspondences between detected features in a training image and those in input frames [113].

In previous work [115], and based on the work of [101], we showed that Random Forest is a suitable classifier that can be applied in markerless tracking. This classifier is computationally fast and able to support a large number of different classes in high dimensional spaces (the number of features in each class).

In the following section the approach based on Random Forest is described in more detail.

5.3 Random Forest

We propose to use Random Forest for interest point matching. The use of a matching learning technique for point matching allow to correctly match points in the presence of severe geometric and photometric transformations. Next, a brief introduction about ensemble learning and Random Forest classifier are given.

5.3.1 Ensemble Learning

Ensemble learning is the process where multiple models, such as classifiers or experts, are generated and combined to solve a particular computational problem. Ensemble learning is primarily used to improve the classification or prediction performance of a given model. Such processes can be also known as multiple classifier systems. Classifier ensemble approaches are suitable in scenarios where a single

classier approach does not t correctly. For example, non-linear complex decision boundaries are difficult to be addressed by a single model. By averaging multiple models, generalization performance of the classier can be improved. Moreover, when large volume of data is available, training a single classier is usually not practical. Better performance and lower training times can be obtained by parallelizing classier ensembles members.

A Random Forest ensemble (also known as random subspace) is an algorithm proposed by Breiman [116] that has been widely used by related community [95], because it shows good trade-off between accuracy and computational cost. This approach uses a large number of individual, unpruned decision trees as base learners which are created by randomizing the split, i.e. features to be tested, at each node of the decision tree. This type of ensemble can be seen as a combination of Bagging and a random selection of features, in order to induce diversity. Each tree is likely to be less accurate than a tree created with exact splits,i.e. complete analysis of all sample features, but by combining several of these approximated trees in an ensemble, accuracy is improved. Term Bagging or bootstrap aggregating, is one of the earliest and simplest ensemble based algorithms, with a surprisingly good performance [117]. As mentioned before, diversity is one of the key factor of success of classier ensembles. In the case of Bagging, diversity is obtained by using bootstrapped replicates of the training data samples. That is, different training data subsets are randomly drawn, with replacement, from the entire training data set. During decision stage, every individual classier is then combined by taking a simple majority vote of their decisions. Similar to Bagging, Boosting also creates an ensemble of classifiers by re-sampling the data, which are then combined by majority voting. However, in Boosting, resampling is directed to provide the most informative training data for each consecutive classier. The main idea is to assign a weight to each sample in the training set during training. This weight varies by increasing the value of those misclassified samples, while weights of correctly classified are decreased. In this way, the algorithm generates classifiers (ensemble members) iteration by iteration, focusing on difficult or misclassified data. This procedure provides a series of classifiers that complement one to each other, thus diversifying their decision outputs.

5.3.2 Diversity

Many authors [118, 119] agreed on the fact that classier ensemble success relies on diversity between the members that form the ensemble. Diversified classifiers lead to uncorrelated errors,which in turn improve classification accuracy. Increasing diversity favors the ability to correct the errors done of any of its members. If all ensemble members provided the same decision output, correcting a possible mistake would not be possible. Therefore, ideally individual classifiers in an ensemble need to make different errors on different samples. During output or final decision stage, a combination strategy must be applied. This combination acts as a filter, averaging or smoothing step where total error can be reduced. Specifically, an ensemble system needs classifiers whose decision boundaries are adequately different one from each other. Such a set of classifiers is said to be diverse. Classier diversity can be achieved by using several strategies. Preferably, the classier outputs should be class-conditionally independent or negatively correlated. The most popular method is to use different training data sets through resampling to train individual classifiers. Such data sets are often obtained through re-sampling techniques, where training data subsets are drawn randomly from the entire training data. In our case, we use weak or unstable learners such as decision trees. These type of learners can yield significantly different decision boundaries even for small perturbations in their training parameters or training samples, thus increasing diversity.

In our case, in order to induce diversity in the ensemble we randomize each tree in the ensemble by both randomizing training sample set for each tree, and by randomizing feature selection of each descriptor in each node of the trees.

5.3.3 Classifier Training

In a typical supervised learning scenario, a training set is given and with the goal to form a description that can be used to predict previously unseen examples and recognize known examples. Each class must be defined and described before the training process itself. In supervised classification a class C_i is defined as a set of attributes a_i, known as features $C_i = \{a_1, a_2, ..., a_n\}$.

In order to define the classes that will be recognized by the classifier, we use a point extractor [4] to get the candidate points and their surrounding patches, as shown in Figure 5.5.

Then, the classifier assigns a class number to each point, and their class de-

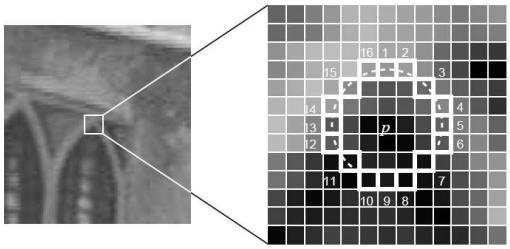

Figure 5.5: (a) Interest point p, (b) Pixels surrounding the interest point p (image extracted from [4]).

scriptor is defined. The descriptor of each class is constructed as the pixel intensity values of the extracted patch centered at the interest key-point. Once the classes to be recognized by the classifier are defined, a training set must be generated. As described in [101], we can exploit the fact that the patches belong to a planar surface. Therefore, we can then synthesize different new views of the patches using warping techniques as affine deformations. These affine transformations are needed to allow the classifier to identify or recognize the same class seen from different points of view and at different scales. This step is particularly important, when the camera will be freely moving around the object.

Once the training set is ready, the training task can be started. During this task, a number of examples are randomly selected from the available ones. These examples are pushed down in the trees. In order to decrease the correlation between trees, and thus increase the strength of the classifier, different examples from the training set must be pushed down in each tree. This randomness injection favors the minimization of trees correlation and avoids over fitting as well. This term refers to the situation in which the training algorithm generates a classifier which perfectly fits the training data but has lost the capacity of generalizing instances not presented during training.

While building up the tree, each non-terminal node of every tree is treated as follows:

- N training examples from the training reach the current node.
- A random set of n pixel positions within the image are selected and written in that node.
- The examples are tested with the selected set of pixels. Depending on the

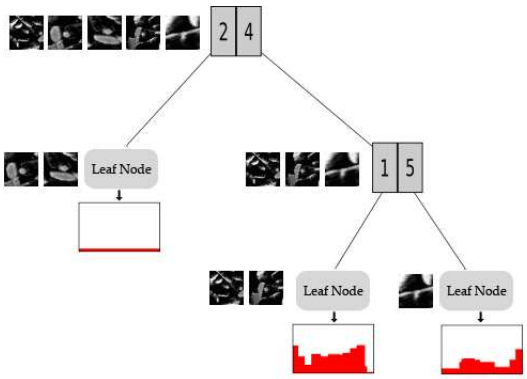

Figure 5.6: Random Tree construction: When the examples reach leaf nodes the posterior probability distributions are updated.

result of this test, they are pushed down to their corresponding child node.

- The above process is recursively applied to the children nodes, whether until there is only one example, or only one class is represented in the remaining examples or the maximal predefined depth is reached. As shown in Figure 5.6 when the examples reach a leaf node, the posterior probability distribution are updated with those examples.

Once the descriptors reach the bottom (maximal depth) of the tree, it is said that they have reached a terminal node or a leaf node, and the recursion stops. In leaf nodes the class posterior probability distributions are stored. These distributions represent the ratio class examples from the training set that has reached that node, with respect to the total number of examples in the training set. When an example of a given class has reached a leaf node, the posterior probability distribution stored in that node must be updated accordingly.

The tests to be performed in each node j in every tree K can be, for example, binary tests based on the comparison of the intensity values of two pixels as:

$$n_{k,j} = \begin{cases} GoLeftChild & if(p_{j,1} - p_{j,2}) \geq t \\ GoRightChild & otherwise \end{cases} \tag{5.5}$$

Where $v(p_{j,1})$ and $v(p_{j,2})$ represent the intensity values of two pixels located respectively at positions $p_{j,1}$ and $p_{j,2}$stored in node j. The values of these positions were randomly selected during the training step. The value of t represents a thresh-

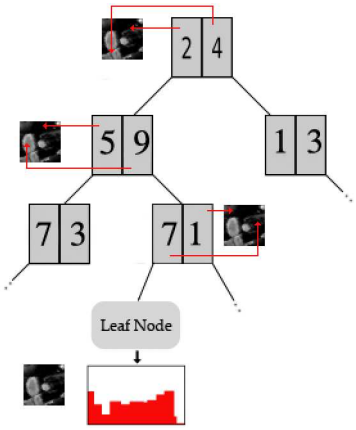

Figure 5.7: Example of image patch classification: The image patch traverse the tree until a terminal node is reached.

old that can also be randomly selected during training. We have also experimented that, given the weakness of the tests, smoothing every patch before training and classification, significantly increases the final reliability of the classification.

5.3.4 Tracking

Once the classifier is built, i.e. the pixels to be tested in each node and the class posterior distributions of all classes are calculated, it is ready to identify keypoints. During the classification task any example (image patch) is dropped down in every tree that constitutes the forest. These examples will be dropped down the tree until they reach a leaf (terminal) node. The node they reach will depend on the results of the tests,i.e. results of applying Equation 5.5 obtained in the previous non-terminal nodes they visit, as depicted in figure 5.7.

Examples to be classified traverse the tree until it reaches a leaf node. When the example reaches a leaf node, the tree will return the posterior probability distribution vector stored in that node. This probability vector represents, for each class, the probability of the example to be an instance of one of the trained class. This is $P(Y = C_i|T_i, n = \eta)$ where T_i is a given tree of the forest and η is the reached node by the example (image patch) Y and C_i represent every class that was previously trained, during the training step. The size of the posterior distributions vector equals the number of different classes trained by the classifier.

As mentioned before a Random Forest is a multi-classifier, i.e. is a set M of

classifiers $T_iM = T_1, T_2, ..., T_n$. The main idea of a combination methodology is to combine a set of models (*classifiers*), each of them solving the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than those obtained from a single model. Like any other multi-classifier, the Random Forest needs to combine the independently generated output by each tree in the forest in order to assign a final class label to the examples to be classified. In our approach we are using a distribution summation combining method [120]. This method sums up the conditional probability vector obtained independently by each tree in the forest. The selected class is chosen according to the highest value in the total vector:

$$Class(x) = argmax_{c_i} \sum_k P_k(Y = c_i|x) \tag{5.6}$$

During tracking, the Random Forest classifier is applied to interest point matching between points m extracted from images and points $M$of the model. With the set of potential matches $M_i \Leftrightarrow m_i$ the homography estimation can be obtained as explained in 5.2.1. After the classification step, wrong classified examples (outliers) can be removed by using robust estimation techniques such as RANSAC (see Appendix B) in order to obtain a more accurate homography estimation. Furthermore, the final estimation can be refined by using Levenberg Marquardt non-linear minimization starting from the estimation obtained by RANSAC, and using all the inlier points. This non-linear minimization favors the reduction of the jitter problem, obtaining more accurate estimations.

5.3.5 Application to markerless tracking

The approaches for markerless tracking described previously were applied within an innovative system for collaborative mobile mixed reality design indoor and outdoor review. In the next section we describe this application in more detail.

The application involves the integration of different modules such as visualization device (HMD, display wall), rendering, image transmission and tracking. All these modules can interact and share information through a communication module or communication backbone. Similarly the other subsystems proposed in the design, the tracking subsystem needs to be connected to the communication backbone in order to deliver tracking information to other modules, like the rendering module. The rendering module will uses tracking information (camera pose) to update the virtual camera accordingly and therefore renders the virtual objects correctly

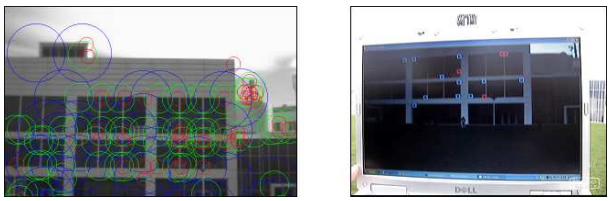

Figure 5.8: Keypoints extracted from a building facade for training.

aligned with real ones. The connection of the markerless tracking module with the communication backbone is realized by using OpenTracker [121]. OpenTracker is an open software architecture that allows the interaction between different tracking approaches and tracking input devices. During the tracking process, every new camera pose estimation must be converted to an OpenTracker state structure and delivered through the communication backbone to be accessible for other clients.

As described earlier, tracking by detection techniques require an off-line process when the classifier is trained. For this task, one image of a highly textured plane, such as a building facade or a picture over a table, must be acquired. After the acquisition, some feature points and their surrounding texture patches are extracted from the image, and synthetic views of the plane are generated. Afterward, the training step starts automatically using the generated views as the training set. Once the training period is finished, the system is ready for tracking as shown in figure 5.8.

5.4 Results

5.4.1 Matching Results

Figure 5.9 shows the results of 225 keypoints recognition, after 10-fold cross validation using 50 trees and 400 samples per class,i.e. keypoint. In this evaluation there was not change in scale, i.e. the random affine transformation applied to training samples performed in-place rotation only, no modifying scale. As can be seen the classifier is very robust to rotation transformation, hence being invariant to this type of transformation.

Results depicted in Figure 5.10 shows the results obtained after 10-fold cross validation of a Random Forest, consisting of 50 trees. This classifier was trained with 225 classes and 400 samples per class, where both full range in-plane rotation

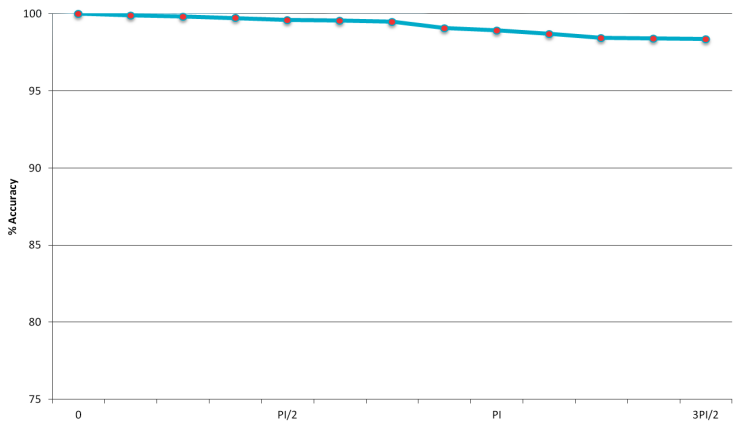

Figure 5.9: Rotation Transformation Matching Accuracy.

transformation and scale transformation were applied. We created several training sets by changing the allowed ranges of scale transformation. As can be seen, as the range of scale transformation increases matching accuracy decreases, from 98% accuracy obtained in $[0.8-1.0]$ scale range to 61.5% obtained in $[0.5-1.5]$ range. Clearly, the classifier is more sensitive to scale transformation than to rotation transformation.

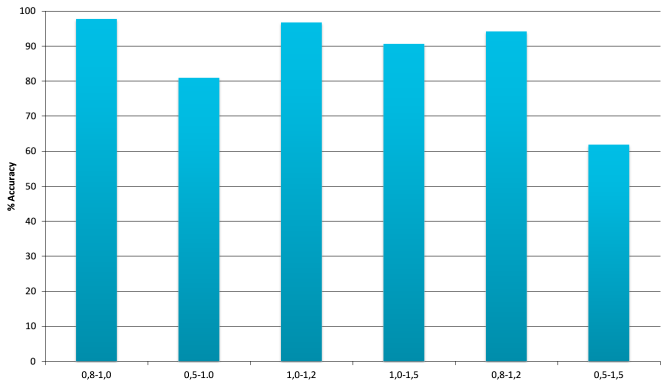

Figure 5.10: Scale Transformation Matching Accuracy.

We also evaluated the impact of training set size in the performance of the classifier. Results shown in Figure 5.11 we obtained by using a classifier consisting of 50 random trees of 15 deep levels each, and 255 different classes. In this case, we

trained the classifier for supporting full in-plane rotation range and $[0.5-1.5]$ scale transformation range. As can be seen, as the number of samples in the training set increases, the accuracy also increases. Clearly, classification accuracy converges around 2100 samples per class. Using for than 2100 samples per class does not improve classifier performance, while training time increases significantly, as shown in Figure 5.12.

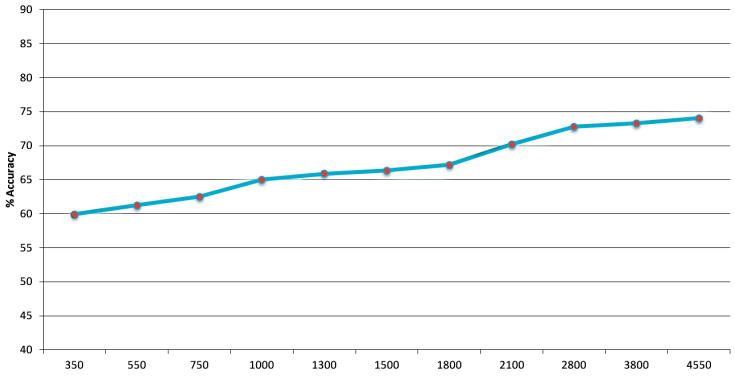

Figure 5.11: Training Size.

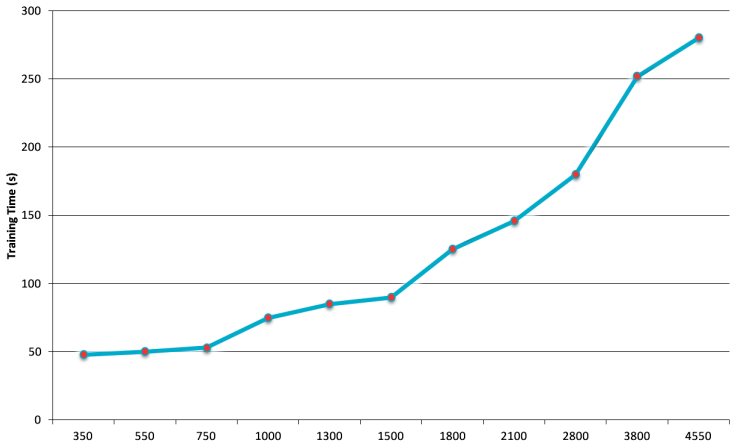

Figure 5.12: Training Time..

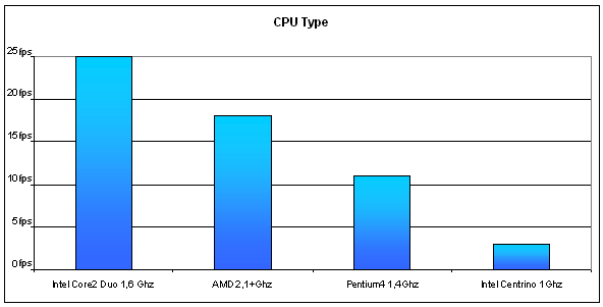

Figure 5.13: Frame Rate Tracking Results.

5.4.2 Tracking Results

The approach based on recursive tracking is very unstable, tending to fail easily. Moreover, it does not allow to move rapidly the camera, as of image blur generates tracking error. In comparison with the recursive tracking implementation, the tracking by detection module allows the tracking to run faster, being more robust against partial object occlusion, or fast camera movement.

The classifier integrated in the tracking by detection module is trained to be able to recognize about 150 different classes (image patches). The Random Forest classifier is constructed with 20 trees. As mentioned before, every tree that forms the forest is independent from the rest and will generate an individual output. The forest is trained with a training set of 1000 synthetically generated new examples. This training step, i.e. the set up preparation takes about 10 minutes. This size of the training set is a good compromise between training time and final accuracy of the classifier.

In order to evaluate the computational costs of the method, we conducted some tests using different hardware with the same memory amount but different CPUs. The obtained frame rate on different CPUs is given on figure 5.13.

The obtained frame rate with SVGA image resolution is about 20-25 frames per second (near real-time) on a 1.6Ghz dual core CPU by using the Random Forest based classification technique. This frame rate may vary depending on the accuracy of the tracker, i.e. depending on the number of different points to be recognized.

For about 150 points the tracker obtains good accuracy and the frame rate is near real-time. The drift and jitter are well controlled, so no severe displacements of the objects occur. On older CPUs, the obtained frame rate is lower, for the same number of points and trees. Better frame rates can be obtained by switching off the

jitter filtering module or by reducing the maximum number of identifiable points, but the accuracy decreases, increasing object jittering. Regarding robustness and practicality, the tracker can run indefinitely without requiring a new initialization.

5.5 Conclusions

In this work we have presented two approaches to solve the camera pose estimation problem in uncontrolled environment. While the recursive approach is computationally light, it is also very unstable and tends to fail, losing tracking information easily. The approach based on tracking by detection is more robust.It does not accumulate tracking errors over time and can obtain real-time frame rate.

We selected the Random Forest based classifier, as being fast, accurate enough and supporting a high number of identifiable classes, which makes it more robust against partial object occlusions. As a drawback, this approach requires a preprocessing to train the classifier with images of the plane to be tracked.

We think that machine learning techniques such as Random Forest is a very promising technique for optical marker-less tracking. We project to extend our work to support on-line training classification, like in [113]. The advantage of online training is that it allows the tracking to update the model with new feature points not present in the original training set. This increases the robustness of the model and the overall accuracy of classification rate. As described in [109] on-line training can be exploited in several frameworks such as Simultaneous Localization and Mapping (SLAM), or other recursive techniques [122] as a tracking initialization mechanism.

Chapter 6

Conclusions

This Chapter summarizes the main contributions and conclusions provided in this dissertation. More specific conclusions and future work lines have been exposed in their corresponding Chapters.

- We carried out an exhaustive evaluation of several state-of-the-art interest point detectors, by measuring their performance against several transformations. More precisely, we evaluated the repeatability index when severe scale, rotation, affine, projective transformations, along with photometric transformations. One of the main conclusions is that calculating an estimation of characteristic scale is essential for getting invariance not only to scale transformation but also for affine and projective transformation. Another conclusion is that, in general, point detectors based on local gradient computation seems more stable compared with those based on pixel intensity distribution or texture. However, gradients or derivatives computations, for scale-space calculations for example, are computationally expensive even though efficient approximations such as box-filtering([3]) or triangle-shaped([123]) do exist. Recent approaches like KAZE looks very promising because of its new proposal of non-linear scale-space estimation, however its computational performance is still to far away to be applied in real-time applications. Apart from computational costs, SIFT detector is still one of the best approaches in several situations.

- We propose a new local descriptor based in Trace transform, that can also be applied as a global descriptor. DITEC local descriptor shows performance, in terms of matching accuracy, similar to well-known SIFT descriptor at a

similar computational cost. In addition, DITEC does not require rotation normalization, hence a detector that does not estimate dominant orientation could be integrated in the pipeline, thus saving computational costs. However, current implementation of DITEC local approach is still computationally expensive to be applied in real-time scenarios or in devices with limited resources such as mobile phones.

- Similarly to interest point detectors, we evaluated several state-of-the-art local feature descriptors by measuring matching accuracy against the same geometric and photometric transformations. An overall conclusion is that in addition to the characteristic scale estimation, local dominant orientation estimation improves overall matching performance, because it allows to either rectify image patches or extract information relative to this orientation, thus resulting in better discriminant descriptors. However, DITEC approach demonstrated that very good performance regarding rotation invariance can be achieved even without the need of dominant orientation estimation. It worth noticing that nowadays there is a clear tendency towards bit-streams based descriptors. These descriptors have the advantage that simply CPU operations such as XOR can be employed for efficient descriptor distance computations. In addition, they can be packed in few bytes of memory compared with real-valued descriptors, hence can be easily ported to devices with limited resources such as mobile phones. Nowadays, the overall best solution found so far is FREAK descriptor. This approach shows the best trade-off between computation requirement and matching accuracy against geometric and photometric transformations. Both FREAK and DITEC descriptor relies on an interest point detector with characteristic scale computation, hence they are not completely independent. Additionally, FREAK requires for the point detector to estimate dominant orientation, while DITEC does not.

- In addition to conventional feature descriptor approaches, we investigated classifier ensembles for feature point matching. We evaluated how Random Forest classifier ensemble can be applied very efficiently for matching features. We validated this approach in an augmented reality scenario. However, this type of approaches need a training process step before any matching process takes place, thus its application is limited compared with previous evaluated approaches.

- We implemented an evaluation framework and a set of images captured in

well controlled condition, that allow precise performance evaluation of feature detectors, descriptors and matching algorithms. The proposed evaluation framework along with Ground-Truth data can be used by any researcher or Computer Vision practitioner regarding development of interest point extraction or feature descriptors, in order to reproduce the experiments described along this Thesis. Images, Ground-Truth homography data, binary files as well as C++, Python and Gnuplot source code is freely available in www.vicomtech.tv/keypoints. The proposed framework can be also extended by any researchers including other measures.

Appendix A

Evaluation Framework

This appendix describes the experimental framework of the evaluations of interest point detectors and feature descriptors described in Chapter 3 and Chapter 4 respectively. We provide a short review of antecedents in Section A.1, more details of the environment are provided in Section A.2, we report the protocol we used for producing a particular set I of images under controlled affine and photometric transformations in Section A.3.

A.1 Introduction

Antecedents There are several quality parameters of point detector and feature descriptor algorithms defined in Chapter 2 that can be measured [23], such as the point extractor accuracy, descriptor robustness or invariance. The fair comparison of different algorithms needs a normalized testing protocol and test benchmark. The seminal works of Mikolajczyk [124, 1] set the foundations for key point extractor and feature description comparative evaluations. Since then, several new approaches for key point or region extraction [125] and for feature descriptor [3, 126, 127, 50] were tested against Mikolajczyk's data set and evaluated with their scripts, which are freely available in `www.robots.ox.ac.uk/~vgg/research/affine/index.html`. An extension of [2] is proposed in [128] to the analysis of key point detection repeatability for non-planar scenes, using tri-focal tensor geometric restriction for estimating the ground-truth data of their own data set they found several differences in key point repeatability scores when applied to non-planar scenes. Another extension of [2] benchmark data proposed in [129] is

a collection of videos taken from some pictures with different types of textures and different light conditions, which are used to evaluate key point matching strategies specific to camera tracking applications. Very recently, in [6] the authors add to the framework of [22] their private approach, very similar to our own proposal in `computer-vision-talks.com`, allowing to compare the robustness of their descriptor against different geometric transformation values, in the form of ratio between correct and wrong matches.

Image transformations can be categorized in two different classes: geometric and radiometric transformations. Geometric transformations modify the shape or the location of a feature in the image space, while radiometric transformations change the feature appearance, i.e, the intensity or color value of the pixels. Changes in lighting conditions or camera acquisition parameters, i.e, sensor sensibility, exposure time or lens aperture, directly affect radiometric appearance of image by changing their luminance and/or chrominance information. Regarding geometric transformations, in our study we focus on linear ones: projectivities, homographies, euclidean transformations, rotations, translations and scaling. Nonlinaear transformations related to optics such as lens distortions are out of the scope of this study.

Robustness In the context of point matching, a robust key point is a point of the same structure in two views of the scene that can be extracted and matched even if some types of geometric or photometric transformations occur between the acquisitions of the images.

Environment availability All materials and tools generated in this Thesis work, i.e., images, evaluation framework code and binary executables are freely available on-line in `www.vicomtech.tv/V6`.

A.2 Evaluation Framework

Our data set and evaluation framework is based and inspired in [2]. In addition to Mikolajczyk's approach, our data set comprises a higher number of images, with higher resolution and with better and controlled capture conditions. Nowadays, mobile devices are becoming part of our everyday lives where computer vision applications are becoming very popular. Therefore, our testing data set reproduces

different aspects of mobile devices, such as a low dynamic range of their integrated image sensor. In this way, our data set includes a set of images that can be used to evaluate the robustness of key point extractors and descriptors approaches against photometric transformation, such as luminance and chrominance noise for mobile device environments.

We have implemented an evaluation framework as an extension on the one provided in the Open Source Computer Vision Library, OpenCV [130], following [2]. The OpenCV framework uses the class hierarchy implemented in OpenCV that nicely decouples key point extraction from key point description and descriptor matching allowing the user to easily define experiments by mixing several interest point extractor with feature descriptors and point matchers. Our evaluation framework is written in C++. Our approach can help in the evaluation of future extractor or descriptor approaches because it can be easy integrated in a research development closer to the final application. In this way, not only features such as repeatability of descriptor distinctiveness can be measured but also efficiency in terms of computation time or memory consumption can also be measured directly at the same time. Finally, our approach also supports reading of Mikolajczyk's file format, allowing the comparison with previous approaches or studies. The evaluation framework is able to generate many useful measures in order to allow the researcher or the computer vision practitioner to obtain valuable insight about the behavior of any interest point extractor or feature descriptor, being used in a matching scenario.

A.3 Image Data set acquisition

The image capture has been conducted using a methodology ensuring that only one kind of transformation occur for a series of images. This allows to determine how sensitive isA_m to a specific image formation factor. In order to supplement the data set of real images, we present an image generator that allows producing images with affine or photometric transformations for testing purposes

Our image acquisition setup is composed of a DSLR Canon 7D and an Ipad3 with a 5 Mega pixels on board camera. In the Canon 7D scenario we used a Tamron 17-50mm 2.8 and a Canon 100mm f2.8 macro lenses. The macro lens is able to render images with almost negligible geometric distortions, i.e, pincushion or barrel aberrations, along with rendering very sharp edges or boundaries, as shown in Figure A.1.

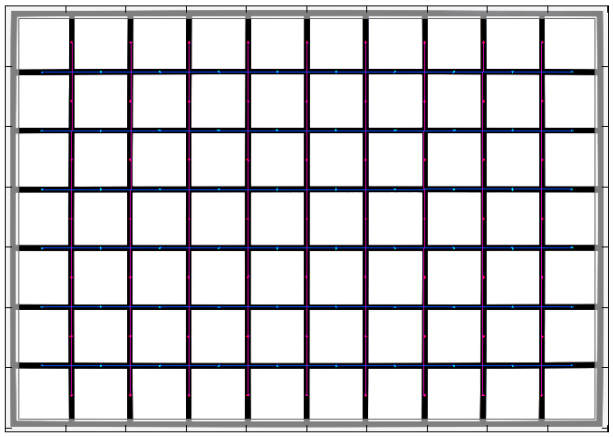

Figure A.1: Barrel distortion of Canon 100mm Macro 2.8 (photozone.de).

In opposite, Tamron lens, due to the use of more optical elements than in macro lens for covering a zoom range between 17 and 50 mm, shows a more noticeable barrel distortion, as shown in Figure A.2. In this case, we rectified acquired images by estimating distortion factors till second order.

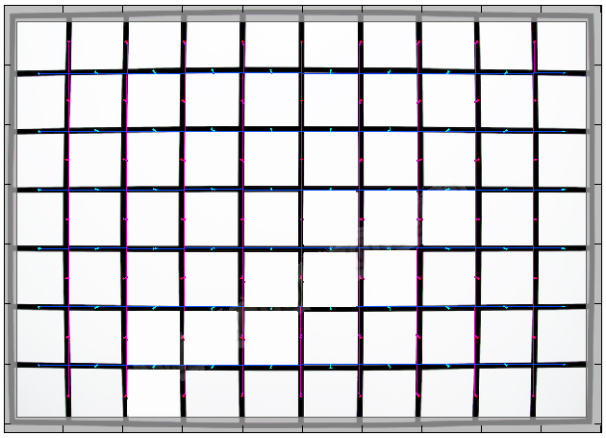

Figure A.2: Barrel distortion of Tamron 17-50 2.8 at 17mm (photozone.de).

In addition to the camera, we used two Canon 580EXII flashes with light diffuser, both wirelessly operated and synchronized with the acquisition. By operating the acquisition remotely we can ensure that only one transformation, either geometrical or photometrical, is applied during image capturing, avoiding any non

intended intervention in the position, orientation or camera and flashes.

In the case of the Ipad setup we can not synchronize the light with the acquisition, so we decided to use continuous light source instead of flashes.

A.3.1 Geometric transformations

In order to generate a set of images with perspective distortion, we used a robotic arm with a Canon 7D attached with Tamron lens in order to generate different points of view of the same target image, an approach similar to [129] to generate known, repeatable and precise positions and trajectories around the target scene. We used a Wacom Cintiq screen for displaying target images. The set of target images is a broad sample of different image types, such as those with structured or unstructured textures, images with low texture, or images with repeating textures or patterns. Many authors [56, 23, 129] agree on the importance of evaluating key point extractors and descriptors in such different conditions, in order to truly evaluate the robustness of their approaches.

The robotic arm is a KUKA LWR IV+, which has 7-DOF, a payload of 7 kg and a repeatability of 0.05 mm. The desired position and orientation of the robot's end effector can be commanded from a remote PC, using the KUKA Fast Research Interface (FRI).

We generated circular trajectories (arcs) to capture images from several points of view of the Wacom screen with different values of captured perspective distortion. Robot effector trajectories are specified by three points in 3D space with respect to the robot's base coordinate system. The specified trajectories are sampled into the desired number of points M where images are to be taken. The set $Q = \{Q_1, Q_2, \ldots Q_M\}$ constitutes the resulting discretized trajectory. Each Q_i is a 3x4 matrix that describes the *i-th* pose (position and orientation) of the camera. Analogously, the orientation of the camera at each Q_i is determined by a linear interpolation of the total rotation matrix R_T, defined by $R_T = R_M(R_1)^{-1}$, where R_M and R_1 correspond to the orientation components of Q_M and Q_1, respectively.

Each element of Q is used as a set point for the robot's Cartesian controller; all the points in the set are traversed in order. When the position and orientation errors with respect to a particular Q_i are below some predefined thresholds, a signal is sent to the camera in order to take N pictures synchronized with the images displayed in the Wacom screen. The first picture corresponds to the calibration pattern image, the following $N-1$ pictures capture the experimental images.

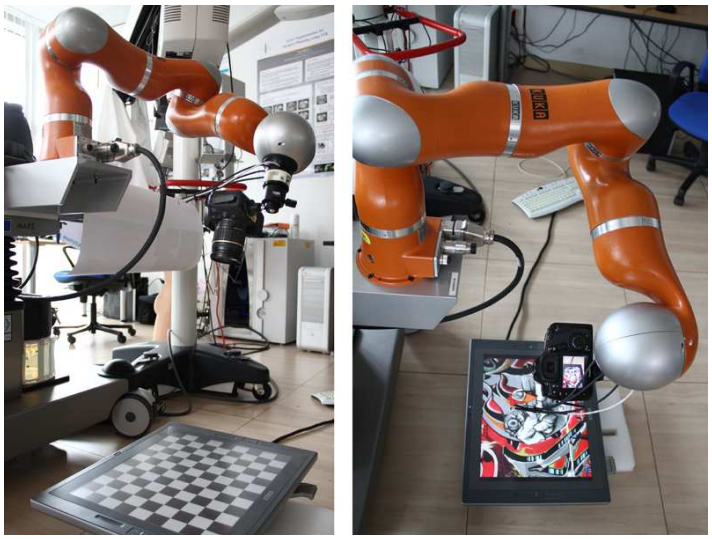

Figure A.3: Image acquisition setup with Kuka robot arm and Canon 7D attached.

Figure A.4 shows a 3D reconstruction of a known generated arc trajectory of the camera around the Wacom screen, from a circular sector of radius equal to 0.4 m, covering a total angle of 70 degrees. We used the calibration pattern image for calibrating the camera, i.e, estimate extrinsic and intrinsic parameters, and also for accurate estimation of the homographies between camera positions. We used the estimated camera calibration parameters for rectifying (undistort) the images acquired with Tamron lens, which exhibits around 2% geometric barrel distortion. Canon 100 Macro lens provides images with negligible geometric distortions. All images of our data set are geometrically corrected, thus neither barrel nor pincushion distortions are present.

A.3.2 Photometric Transformations

Focus effects: review of fundamentals

In a camera with lens, all rays coming from surfaces that are in the focus distance will be projected as a single point in the camera sensor. In contrast, rays from all other surfaces that are in front or behind the focus plane will be projected as a smoothed version of that point, because they will converge in front or behind the imaging sensor plane, as shown in Figure A.5. The shape of this projection, called the “circle of confusion” represents point response of the unfocused surface,

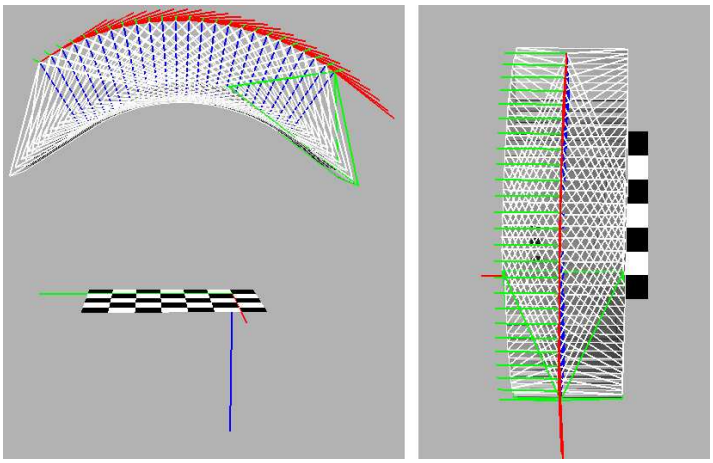

Figure A.4: Recovered trajectory of a Robot driven image acquisition.

whose radius increase with the distance of the imaged surface to the focus plane. The range of distances of surfaces imaged without blurring form the depth-of-field (DOF) of the lens. The DOF is inversely proportional to the aperture value of lens, as shown in Figure A.7.

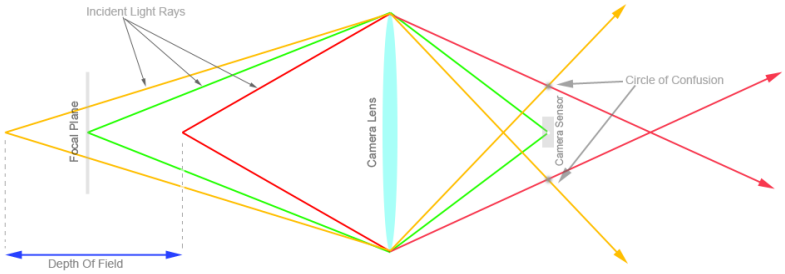

Figure A.5: Light rays projecting in camera sensor .

The boundaries of two surfaces of a 3D scene that are in focus, and therefore that are inside of the deep-of-field of the lens, will be rendered in the final image as a hard transition between them, while in case of out of focus structures the transition between them will be rendered as smooth, as shown in Figure A.9. In order to quantitatively measure the sharpness of an image, i.e. how well different projected surfaces are in focus, the acutance measure can be used.

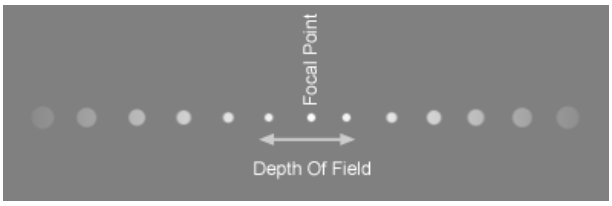

Figure A.6: Different circles of confusion .

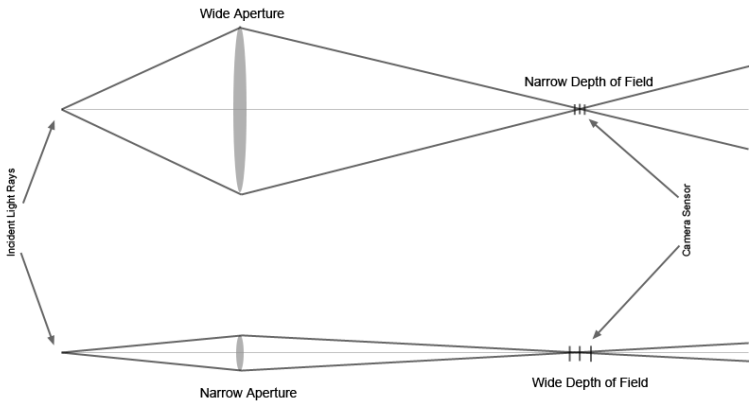

Figure A.7: Deep-of-field depending on lens aperture.

Pictures in Figure A.8 show the same scene correctly and incorrectly focused. Most of the approaches for interest point detection rely on computation of image derivatives of different orders. For example, the Harris [20] extractor uses a measure M_H based on the autocorrelation matrix A for interest point detection, as defined in Equation A.1:

$$M_H = det(A) - k\,trace^2(A). \tag{A.1}$$

The measure M_N proposed by Noble[131] modifies Harris' cornerness measure, by removing the user parameter k.

$$M_N = 2\frac{det(A)}{trace(A)+\varepsilon'}, \tag{A.2}$$

where ε' is an small positive constant, usually machine epsilon.

Figure A.8: (Left) Correctly focused scene, (Right) Incorrectly focused scene.

Figure A.9: (Left) Two surfaces ideally perfectly in focus, (Right) Soft boundary transition between two unfocused surfaces.

Images become unfocused when the main objects and surfaces in the scene are away from the focus plane. The values of the autocorrelation matrix decrease as images are increasingly unfocused, because local curvature decrease due to smooth transitions between borders, as shown in Figure A.9. Therefore, the number of detection decreases, as shown in Figure A.10.

Focus DataSet

We built an image data set where the focus point are varying from the correct focus point, i.e. all surfaces are accurately rendered as sharp, to a point where all objects appear blurred, as shown in Figure A.11.

We ensure that all surfaces in the scene fall in the deep-of-field of the lens, thus not having, for any given focus value, parts of the scene that are in-focus while others are out-of-focus. We can measure the amount of structures in-focus by computing the CTM acutance [132].

In this subset of images, although the camera was not moved during image capture of the sequence, the changes made in the camera focus required the homography transformation between images to be computed.

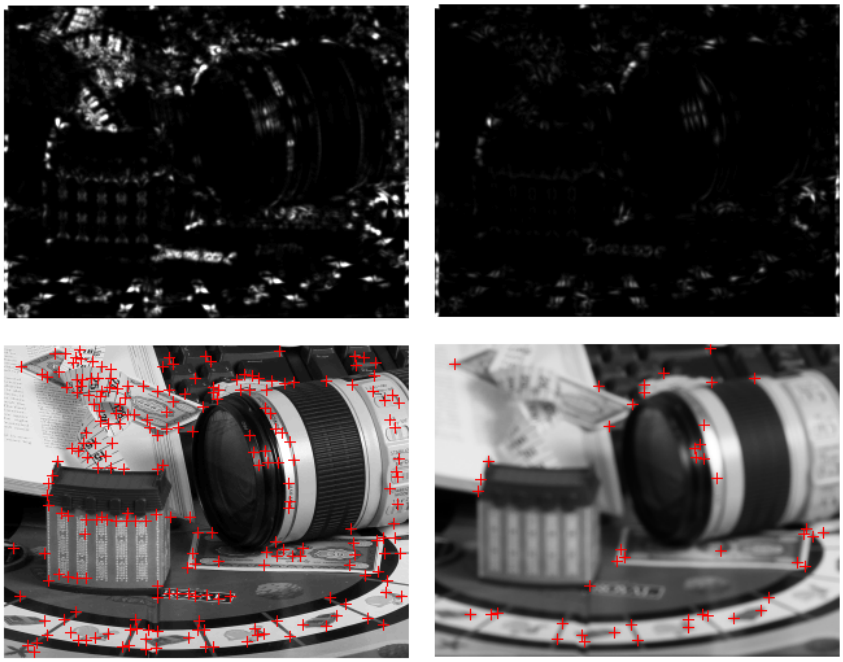

Figure A.10: Effect of focus on Harris point detectors. Left images correspond to the focused image, right to the unfocused. (Top) Cornerness measure images, (Bottom) Detected interest points.

Exposure Data Set

The purpose of this set of images is to allow the evaluation of the robustness of key point extractors, repeatability or feature descriptors robustness against illumination changes and noise. Image acquisition was carried out by using a protocol ensuring that only photometric transformations exist between images. To assure that no geometric transformations were applied during data set acquisition, both the illumination equipment and the camera were operated remotely. We control the flashes to vary the amount of light without changing any camera acquisition parameters, i.e. setting fixed the aperture value, the exposure time, and ISO speed. In this way, neither the DOF is varied along the images that constitute the data set, nor additional noise is added due to an increase of either ISO speed, or due to sensor heat because of longer exposure times. Every image that forms the data set is consecutively reduced approximately an 1/3 of a f-stop, starting from a correct

Figure A.11: Images of image focusing data set.

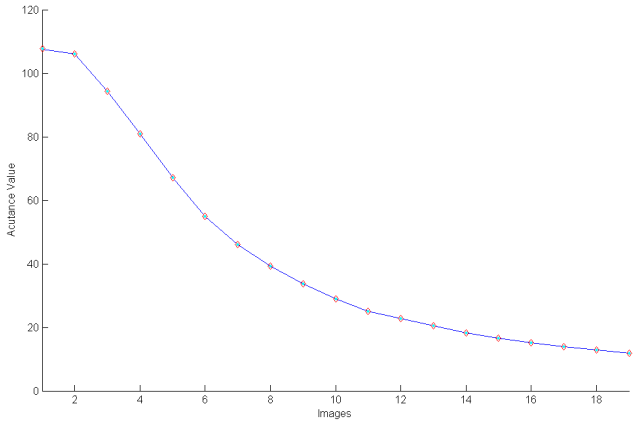

Figure A.12: Acutance measure for focus varying data set.

exposure in the first image, through a progressive reduction on the illumination power of the flashes. This data set is compound of 12 images resulting in a difference approximately of 3 f-stop, between the first and last images.

Figure A.13 shows two images of the same scene taken with an Ipad in controlled illumination conditions. Left image were captured with a correct value of exposure, while right image were captured with approximately 2.5 f-stops less of exposure. In the mobile device setup we used a continuous light source where light intensity can be set manually. Both the focus point and exposure metering point were fixed along the capturing of all images in the data set. In a mobile device, such as the Ipad3, those values are set automatically during image acquisition. Therefore, we used an application for image capture that allowed us to focus and measure exposure always in the same gray neutral part of the scene along the captures. This ensures that the exposure readings are consistent along images acquisition, given

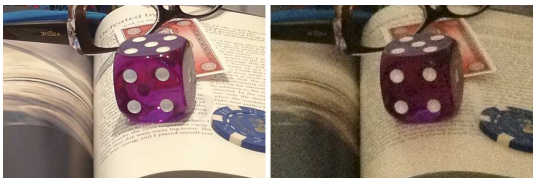

Figure A.13: Images from the photometric noise transformation taken with a mobile device.

different light conditions.

In both instances of image capture devices, as the amount of light decreases, i.e. the signal-to-noise ratio (SNR) decreases, the amount of digital noise increases. This is clearly more noticeable in the case of mobile device, due to the smaller size of its image sensor, and therefore a more limited dynamic range compared with the DSLR camera.

A.4 Synthetic Image data set Generator

In addition to the set of images captured under controlled conditions, we implemented a set of C++ functions and Python Scripts for the generation of testing images by applying either random or systematic geometric transformations, as well as photometric transformations. The Python scripts allow the user to determine the source image, the type of transformation, the number of images to be generated, and the minimum and maximum values for the given transformation. In this way, it is easy to generate tailored data sets, with different types of images, and several types of transformations and transformation ranges. We integrated in this set of scripts functions for generating geometric transformations such as in-plane rotation. iso-tropic and aniso-tropic scaling, as well as affine deformation,i.e. image rotation with anisotropic scaling.

In addition to geometric transformation, we incorporate into our synthetic image generator some functionalities allowing to simulate some photometric transformations, such as defocusing, exposure variations or digital noise addition. Although the resulting images of these functionalities are not completely realistic, because some physical parameters (e.g. diaphragm shape), or the specific features of a given CMOS or CCD sensor (e.g. patterned noise) are tackled, they can be used in many practical applications, such as the generation of addition synthetic samples for using during training step in any machine learning experiment. Digital

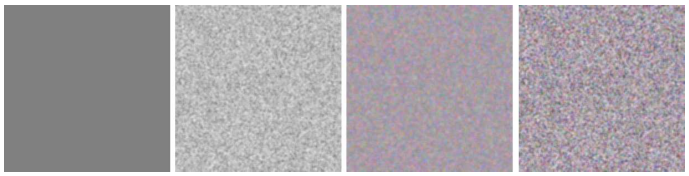

Figure A.14: Examples of different types of noise.

image noise can be divided mainly in two different categories, luminance noise and chrominance noise, depending if the errors are produced in luma(intensity) or in chroma (color). There are some others types of noise such as horizontal or vertical banding (patterned noise), but it does not degrade images as luminance or chrominance noise do.

Figure A.14 shows, from left to right, an image patch filled with 50% gray value, contaminated with just luminance noise, with just chrominance noise and with both types of noise at the same time. Depending on the nature of the camera, mainly the imaging sensor, or the camera parameters during acquisition, i.e. exposure and ISO speed, these errors may vary. For example, we can check in the exposure light data set how noise levels increase as light decreases(SNR decreases), more noticeable in the case of the Ipad3. Our image generator is able to create images contaminated with luminance or chrominance noise, or with both types simultaneously.

Appendix B

Robust Estimation Methods

In this Appendix we review some robust estimation methods that are useful for the robust correspondence and homography estimation. Section B.1 gives a short introduction. Section B.2 reviews the robust estimation based on random sampling approaches found in the literature. Section B.3 reports an evaluation of these methods on an example pair of images. Some conclusions on the relative performance of these methods are gathered in Section B.4.

B.1 Introduction

As described in Chapter 2 applications such as structure from motion, image registering, stereo matching or camera calibration, builds upon a standard pipeline consisting in extracting interest points or local features, compute identifiers or descriptors based on those points, matching descriptors to find correspondences and, finally, apply some type of filters, constraints or geometric verification in order to remove or reject wrong correspondences. As we described in Chapter 3 and Chapter 4, due to the nature of matching process several wrong correspondences or miss-matches can be generated. These errors may be due to the presence of similar structures in the scene that region descriptors are not able to univocally represent, or to the presence of strong perspective distortion that interest point extractors are not able to cope with. In practice, we must estimate the parameters of a model M from noisy observations containing outliers. Therefore, we must employ robust techniques in order to estimate accurately the parameters of the true model M. Robust estimation methods have been extensively employed in many computer

vision applications. In fact, one of the most powerful robust estimation algorithms, i.e. RANSAC [133], was developed for registering 3D to 2D points sets [8].

Minimal sets A key element of robust estimation methods is the random sampling of minimal sets. A minimal set is the minimum number of data samples n required to compute the estimation of the model parameters θ. In random sampling such minimal sets are built so that each point has an equal probability of being selected. A cost function $C(\theta)$ is used to determine the goodness of fit of the estimated hypersurface model to the data.

Outliers Estimation processes conventionally are based on some assumptions about the noise present in the data, such as normality and independence. However, these assumptions often do not hold in real application scenarios, such as the homography estimation based on correspondences of point features extracted from two images. According to [134], an outlier is a sample whose distance to the true model response instantiated by the true set of parameters falls outside some error threshold which specifies the maximum allowed deviation, aka the tolerated magnitude of noise. In homography estimation, errors arise from arbitrary point correspondences, so that errors relative to an specific model do not follow an specific probability distribution. In those scenarios least-square approximations generate poor estimations of the true model or generate completely wrong or degenerated estimations, depending on the magnitude of such observations. This fact is even more critical when the percentage of outlier data surpasses 50% of the data. Robust estimation methods seek to remove outliers so the estimation reaches a desired accuracy level. In fact, they search for the model that discards the minimum number of outlier samples under the desired accuracy level. Figure B.1 shows an example of outlier detection and removal by estimating underlying geometric transformation between images by using RANSAC.

Bias and Break-Down Point In the context of robust estimation *bias* is the difference between the true model M and the estimated model M^*, given that some of the input data were contaminated with errors. Let D_i be a set of inliers of the model M_i, and D_j be the set of inlier data samples after m inlier points were contaminated with noise of magnitude η, turning into outliers. The bias term associated to the model M and a set of samples D_i and D_j is defined by the following equation:

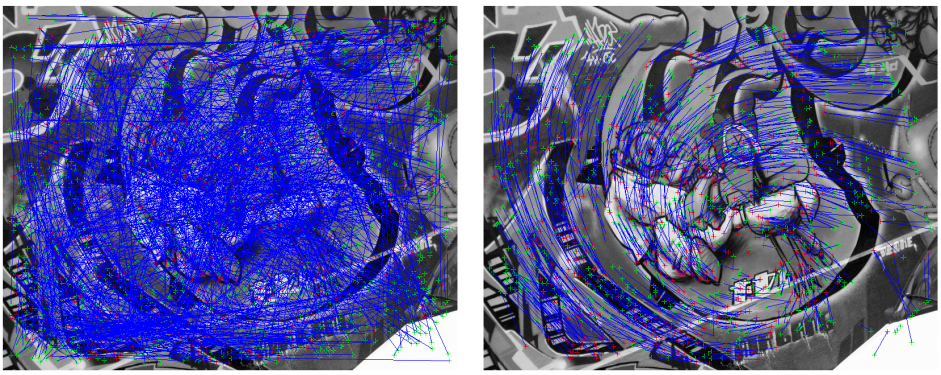

Figure B.1: (Left) Correct and wrong correspondences. (Right) Filtered matches obtained with RANSAC.

$$bias_M(m, D_i) = dist_M(f(D_i), f(D_j)), \tag{B.1}$$

where $f(D_x)$ represents the function that estimates a model M_x given D_x samples, contaminated or not with outliers, and $dist_M$ represents a function that measures the distance between two estimated models. In this case, the function $dist$ measures the difference between the model estimated using only inlier points D_i, and the one estimated after turning some inlier points in outlier points D_j. The bias measures the maximum difference that the presence of n wrong observations can cause to the estimation of the true parameter model M, using only inlier points. The appropriate distance function $dist$ depends on the nature of the models M. In the case of robust homography estimation, several measures have been proposed, such as algebraic, geometric or Sampson error [60]The best trade-off between accuracy, stability and robustness can be obtained by using a geometric distance, as defined in Equation B.2:

$$d_{ij} = d(x_{jb}, H_{ab}x_{ia})^2 + d(x_{ia}, H_{ab}^{-1}x_{jb})^2, \tag{B.2}$$

where H_{ab} represents the true (ground truth) parameter model to be estimated, relating images a, b, and (x_{jb}, x_{ia}) are point correspondences.

Break-down point is another estimator robustness measure. This measure represents the minimum number of outlier points in the observations that generates large bias [134]. It is given by the following equation:

$$BreakDown(D_I, H) = \min_m \left\{ \frac{m}{|D_I|} : bias(m; D_I) = \alpha \right\}, \tag{B.3}$$

where m represents the minimum number of wrong observations, i.e. outliers, out of D_I that tends the estimator to generate large α deviations from the true model H.

B.2 Random Sampling approaches

Since the original RANSAC [133] approach was presented, different new extensions and variations based on RANSAC have been proposed in order to obtain improved performance according to different criteria, such as the final estimation accuracy, the number of samples needed before convergence, or computational cost. According to [135] RANSAC approaches can be divided in three different groups:

- Oriented to improve robustness focusing on the dynamical adaptability of their parametrization in order to adapt to incoming data, such as MLESAC [136]

- Oriented to improve accuracy by generating local optimizations of generated models or by estimating the shape of inlier-outlier distribution, examples are Lo-Ransac [137] and [138].

- Focused on the optimization of the sampling process by guiding it or by changing the pure random sample selection of the original RANSAC. Approaches such as Guided-MLESAC[139], PROSAC [140], GROUPSAC [141] or SCRAMSAC [142] use feature matching distances, given by the feature matcher algorithm, as a prior information for guiding sampling. These priors are iteratively re-weighted as new models M_i are being estimated during iteration. In this way, these approaches build consensus set, i.e. the set of inlier points, iteratively and sampling process is guided by the weighted priors.

In the following, we briefly review some of the most extended approaches and variations of original RANSAC.

RANSAC

RANSAC algorithm [133] is the most widely used robust estimator in the field of computer vision. RANSAC algorithm has been employed in many computer vision applications such as short baseline stereo ([143]), wide baseline matching([144]) or image mosaicing. An in-deep review of RANSAC algorithm, as well as many definitions regarding robust estimation can be found in [134]. RANSAC algorithm is composed of *hypothesize* and *testing* steps that are repeated iteratively:

- **Hypothesize**: A sample of size m is randomly selected with uniform probability from the whole data set D composed of N noisy observations. A model hypothesis M_i is computed from this sample applying a given function f. The sample is a minimal set to determine the model parameters θ. For example, in case of homography estimation m should be four (see Chapter 2) and in case of fundamental matrix estimation, m should be equal to eight ([145]).

- **Testing**: The model hypothesis M_i generated in previous step is validated against the remaining data by counting the points consistent with the estimated model M_i applying the loss function B.4.

$$\rho(d,\theta) = \begin{cases} 0 & e(d,\theta) \leq \delta \\ 1 & Otherwise \end{cases}, \tag{B.4}$$

  where e represents an error function of a given data sample d with respect to parameters model θ, and δ is a user defined error threshold specifying the allowed maximum inlier distance. This loss function ignores inlier data ponits, while outlier data points score a constant value of 1.

Hypothesize and testing steps are iterated. The number of iterations is determined on the basis of the probability of getting a sample free of outliers for a given the fraction of inliers, the model M and the cardinality m of the minimal sample set. Denote ξ the probability of getting a random sample of m data points from D, aka inlier ratio. Therefore, the probability of getting a minimum sample set where at least one of the points is an outlier is $(1-\xi^m)$. The probability to obtain at least an outlier free model M_i in T trials is

$$P = 1-(1-\xi^m)^T.$$

Therefore, the number of minimum number of iterations that RANSAC should perform in order to guarantee at least a given probability P of using only inlier data for the model estimation is given by Equation B.5:

$$T \geq \frac{log(P)}{log(1-\xi^m)} \tag{B.5}$$

As the inlier ratio ξ of a given estimation problem is not known apriori, nor directly observable in many cases, often the user imposes a raw estimation of this ratio and then RANSAC iteratively determines ξ, by computing the fraction of inlier points that satisfies the best model found so far in iteration t.

Given the number of iteration calculated with Equation B.5, RANSAC seeks to minimize the cost function C (Equation B.6)

$$C = \sum_{i=1,N} \rho(d_i, \theta), \tag{B.6}$$

where ρ is the loss function of equation B.4.

Criticisms of RANSAC The speed of RANSAC depends on two factors. First, the level of data contamination determines the number of iterations, or the number of samples to be taken to guarantee a certain confidence in the optimality of the solution. Second, the time needed to evaluate the quality of every estimated or hypothesized model M_i against the whole data samples.

One of the shortcomings of RANSAC is the need to set the inlier threshold δ. As this value increases, more points are considered inliers, yielding in the limit that all observations could be inliers. If this value is set too low, accurate model estimations can be very difficult to achieve, because very small fraction of data will be considered inliers. The higher δ, more solutions with equal values of C (Equation B.6) are obtained, since loss function ρ scores 0 to all inlier points, independently on their distance error e. This clearly may lead to the generation of poor estimations of the true model parameter set [136].

M-Estimator Sample Consensus (MSAC)

The MSAC estimator is a simple but powerful adaptation of the standard RANSAC algorithm. In MSAC the loss function is given by Equation B.7:

$$\rho(d,M(\theta)) = \begin{cases} e_M(d,\theta) & e_M(d,\theta) \leq \delta \\ \delta^2 & otherwise \end{cases} . \tag{B.7}$$

Loss function ρ is an M-estimator where inlier samples, are weighted according their error given by the goodness of fit of the sample to the model, instead of 0. Outliers are weighted equally by δ^2, instead of 1. Therefore, outlier points are penalized, but all outliers have the same treatment. This new loss function does no add any additional computational burden to standard RANSAC algorithm.

R-Ransac with $T_{d,d}$ Test

Randomized Ransac(R-Ransac) [146] is designed to improve the time of convergence of RANSAC. In RANSAC-like algorithms most model hypotheses evaluated are influenced by outliers, computational savings can be achieved when the model hypothesis generates many outliers by evaluating only a fraction $n \ll N$ of data points

The R-Ransac algorithm exploits this fact, rejecting non-valid ones faster than standard RANSAC. The R-Ransac algorithm incorporates an intermediate step, called a pre-evaluation test, where the support for hypothesis parameters in iteration t is tested with a fraction of samples d, selected at random. If this pre-evaluation test success, then the algorithm proceeds as the original RANSAC. If the test fails, the algorithm proceeds to the next iteration, by selecting a new minimal sample set. In our experimental evaluation reported below we have set $d = 3$ as suggested by the authors.

NAPSAC

NAPSAC (*N Adjacent Points Sample Consensus*) [8] algorithm was proposed for coping with the estimation of high dimensional models with noisy data. The key concept in NAPSAC approach is that the set of outliers possess a diffuse distribution, thus a selection of minimal sets based on the proximity of initial candidates can significantly improve the probability of select new inlier points and therefore reduce the number of iterations T before convergence or optimal solution is met.

	Percentage of outliers		
Dimensionality	30%	40%	50%
2	5	7	11
3	8	13	23
4	11	22	47
5	17	38	95
6	24	63	191
10	105	494	3067
20	3753	81936	$3.1x10^6$
30	132910	$1.4x10^7$	$3.2x10^9$
40	$4.7x10^6$	$2.2x10^9$	$3.3x10^{12}$

Table B.1: Samples required to achieve a 95% of probability of selecting inliers, given noisy data and model dimensionality ([8]). .

The probability of selecting n inlier points at random is monotonically decreasing with increasing data dimensionality. Table B.1 shows the theoretical number of samples required for an algorithm using uniform point sampling to have a 95% chance of selecting a minimal set of size m composed only of inliers.

NAPSAC approach is intended to overcome in high dimensional spaces the limitations of uniform random sampling that ignore the spatial relationship between the inlaying data points. NAPSAC proposes to use the distribution of the inlaying data in the multi-dimensional space to modify the point sampling for improved minimal set selection and hypothesis generation. NAPSAC algorithm is as follows:

1. Select an initial point x_0 randomly from all points.
2. Find the set of points, S_{x_0} lying within a hyper sphere of radius r centered on x_0
3. If the number of points in s_{x_0} is less than the minimal size set for model estimation, then retry from step 1.
4. Select points from S_{x_0} uniformly until the minimal set has been built, including x_0
5. Generate model hypothesis with points selected from S_{x_0}.
6. Evaluate residual and proceed as RANSAC or MSAC.

During the evaluation in Section B.3 we use hyper spheres of radius set about 20% of the size of the interval ranging all the data, as suggested by the authors. In the case study of our evaluation, i.e. robust homography estimation, range data is defined by the area, width by height, of the input images.

MLESAC

MLESAC [136] has been specifically designed for homography estimation: successive minimal sets of correspondences $(x_i, x_i^{'})$ are used to derive hypothesized solutions, and the remaining correspondences used to assess the quality of each hypothesis. However, while original RANSAC assess the quality of hypothesized model M_i by counting the number of outliers for an hypothesis given a threshold δ, MLESAC evaluates the likelihood of the hypothesis by modeling the error distribution as a mixture between inlier and outlier points. Usually the point correspondences $(x_i, x_i^{'})$ are assumed to be corrupted by Gaussian noise with $N(0,\sigma)$. Thus, given a set of n correspondences d, the joint probability density function of such point coorespondences, given the view transformation H between both images is given by:

$$p(d,H) = \prod_{i=1..n} \left(\frac{1}{\sqrt{2\pi\sigma}} \right) e^{-\left(\Sigma_{j=1,2} \left(x_i^j - x_i^{j'} \right)^2 + \left(y_i^j - y_i^{j'} \right)^2 \right) / 2\sigma^2}, \tag{B.8}$$

for n correspondences. Taking the negative log of $p(d,H)$,we have as the expression of the log-likelihood of the correspondences:

$$-log\left(p(d,H)\right) = \sum_{i=1..n} \sum_{j=1,2} \left(\left(x_i^j - x_i^{j'} \right)^2 + \left(y_i^j - y_i^{j'} \right)^2 \right). \tag{B.9}$$

It is clear that the true parameter values of the homography transformation H minimize the distances between all correspondences, minimizing the log-likelihood. Therefore, the search for the true parameters of H can be interpreted as the Maximum likelihood Estimation (MLE) of H over d: the search for the best $\widetilde{x}_i'$ of the true positions x_i', that minimizes the expression in equation B.9. The partial error e_i of a given correspondence can be defined as:

$$e_i^2 = \sum_{j=1,2} \left(x_i^j - \widetilde{x}_i^{j'} \right)^2 + \left(y_i - \widetilde{y}_i^{j'} \right)^2. \tag{B.10}$$

Minimizing a cost function based on Equation B.10 , gives the MLE of the true

transformation H that maps x to x':

$$MLE(H) = argmin \sum_{i=1..n} e_i^2 \tag{B.11}$$

The Maximum Likelihood Estimation defined in Equation B.11 assumes a normal distribution in the location error of the correspondence points in both images, while in real situations this assumptions are not always satisfied [136]. As seen in Chapter 4, depending on the nature of the computer vision algorithm employed for the identification of matching points, a lot of miss-matches can be generated. Proposed wrong correspondences will be far away from the true location appearing as outliers almost for any homography, mostly for those approaching the unknown ture homography transformation H. Of course, outliers do not follows any normal distribution. In [136] the authors proposed a new model for the error of correspondence points in d given a transformation H, that combines a normal distribution for representing correct correspondences, the inliers, and a uniform distribution for representing outlier points:

$$P(e) = \left(\gamma \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{e^2}{2\sigma^2} \right) + (1-\gamma)\frac{1}{v} \right), \tag{B.12}$$

where γ denotes a mixing parameter combining both distributions, σ is the deviation of noise following the normal distribution, and v is a constant parameter representing the space for outlier points to be detected. Outlier corresponding points can be generated in any position all along the image space, so v can be set as the diagonal size of the image being processed. The Expectation Maximization (*EM*) algorithm [147] is used for the estimation of γ, starting from an estimation carried out by the standard RANSAC or MSAC algorithms, although other approaches such as gradient descendant are also possible. In the experimental evaluation reported below, we set the number of iterations of the EM algorithm to 5 as suggested by the authors.

LO-RANSAC

LO-RANSAC (Locally optimized Ransac) [137], is a RANSAC variation where an optimization step is introduced inside RANSAC iterations, in order to reduce convergence time. Optimization step consists of taking all data points with error threshold τ smaller than $K\delta$ and use a linear algorithm, such as SVD, to compute new model parameters M_o. The K parameter is iteratively reduced inside optimiza-

tion step until τ is equal δ. After the optimization step, next iteration of standard RANSAC is conducted by recalculating the inlier ratio, given optimized model M_o. In this way, each time a new improved model M_t is found in iteration t LO-RANSAC optimizes this model using putative inliers found so far in iteration t, hence accelerating the time for convergence.

SCRAMSAC

SCRAMSAC [142] is specifically designed for robust estimation of geometric models based on 2D point matching, using a spatial consistency filter. The filter relies on the assumption that inlier points should have both spatial, as NAPSAC [8], and scale similarities. Therefore, instead of performing uniform random sampling over the data SCRAMSAC applies a guided sampling over a subset of pre-filtered data to reduce the outlier points, thus increasing the inlier ratio. Increasing inlier ratio directly increases the probability of selecting only inlier data, thus time for convergence is reduced as a consequence. In addition, SCRAMSAC filter proposes that good correspondences should have similar estimation of apparent scale. As shown in Chapter 3, many interest point extractors such as SURF [3], SIFT [46] or AGAST [81] estimate a scale for each interest point. Lacking other evidence, the estimated scale can be interpreted as the scale of the underlying structure generating the interest points. In this way, good correspondences should come from the same real world structure, thus their apparent scale should also correspond. SCRAMSAC first generates sets of points defined as neighborhood sets, where all points in the set are coherent in scale-space. The generation of these sets allows to generate a subset $C_{red} \subseteq C$ with only "good" candidates in relation with space and scale coherence. Once C_{red} is generated, then standard RANSAC or any of its variations, is applied over this new subset of points.

B.3 Evaluation of RANSAC algorithms

We have carried out several studies trying to evaluate the effect of several factors of RANSAC-like algorithms. We focus our study on robust estimation of planar homographies. For each approach we apply the same termination criteria, based on the number of iterations needed to obtain a probability ε of 95% that selected data are all inliers, given the inlier ratio ξ:

$$t = \frac{log\varepsilon}{log(1-\xi^m)} \tag{B.13}$$

where m is the cardinality of the minimal set of data needed to generate a hypothesis, and ξ is probability to pick up an inlier, the inlier ratio, i.e. the number of true correspondences given the whole data. Tthe inlier ratio ξ is unknown in many practical situations. We, therefore, compute an estimation of inlier ratio at each iteration in order to estimate the number of iteration t needed for convergence. When the number of iterations t is reached, the algorithm stops returning the best estimated model in that iteration. Alternatively, the algorithm stops after reaching the maximum number of iterations allowed. We set this maximum to 500 for each test included in our evaluation. Finally, we integrate in every tested approach the filter proposed in [17] detecting degenerate minimal set configurations, so that no valid homography model could ever be extracted from them, prior to model estimation. We set the inlier threshold error for all evaluations at $\delta = 1.96\sigma$ as suggested by [136, 138] with $\sigma = 0.5$. Due to the random nature of all approaches, we repeated each test 1000 times for statistical soundness.

For this empirical evaluation we used the two images depicted in Figure B.2. We extracted $n = 1000$ SURF features from image 1. By using ground truth homography H_{12} we can estimate, for each feature point x_{1i} from image 1, their corresponding feature point x_{2j} in image 2. Thus, by projecting each x_{1i} with H_{12} we have the set of all n inlier points D. In order to test how different algorithms perform with different levels of noisy data, we contaminated inlier set D with different amount of noise, thus having different inlier ratios ranging from 30% to 100%.

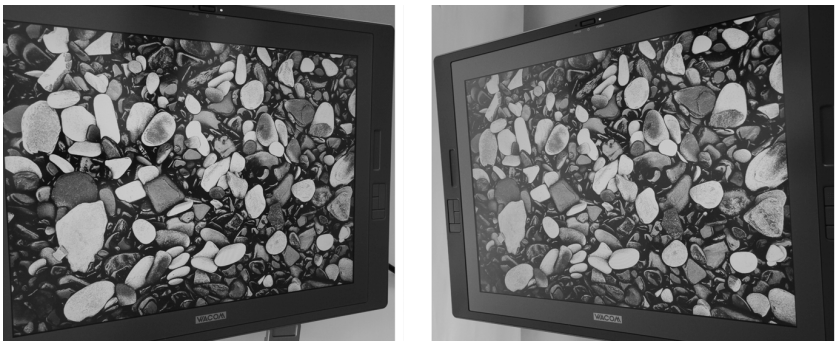

Figure B.2: Evaluation images.

Table B.2 shows the results of 1000 runs of each algorithm using different

Algorithm	30%	40%	50%	60%	70%	80%	90%
RANSAC	36.7	45.7	54.2	63.9	72.6	82.2	91.1
LO-MSAC	36.8	45.8	54.3	63.9	72.7	82.1	91.1
MSAC	**28.8**	**43.2**	**53.3**	**64.3**	72.4	82.2	91.1
MLESAC	35.4	44.6	53.9	63.8	72.6	82.1	91.2
RRANSAC	36.2	45.1	54.0	63.8	72.5	82.0	**90.4**
NAPSAC	37.4	45.8	54.4	64.1	72.8	**81.8**	91.1
SCRAMSAC	36.1	44.0	53.4	**63.3**	**72.1**	82.2	91.0

Table B.2: Estimated inlier ratios.

known levels of inlier ratio. We measured, for each algorithm and each level of noise, the estimated value of inlier ratio. It is worth noticing that all approaches overestimate the true number of inlier, more significantly when inlier ratios are below 60%. From 80% and onwards all approaches almost converge giving inlier ratios very close to ground truth.

Figure B.3 shows the number of different minimal sample sets tried before the algorithm stops, given different values of inlier data. A data set free of outliers is represented as 1, i.e. 100% of data is represented by inlier points only. As can be seen, the most demanding approaches are RRANSAC and MLESAC. RRANSAC need more samples than the remaining approaches, specially when the inlier ratio is lower that 50%. In this range of inlier ratios, the probability of discarding a valid model hypothesis, as RRANSAC does is high, due to the high probability of randomly selecting an outlier point out of d samples, because of the low inlier ratio. Approaches that guide minimal samples selection such as NAPSAC and SCRAMSAC are the most parsimonious in the number of selections. Both approaches pre-selects subsets of data where probabilities of selecting outlier samples are lower compared with the whole data set.

Figure B.4 plots computation time required by every approach before convergence. The algorithm computational resources requirements depend both on the number of samples evaluated and the time needed for evaluating each model hypothesis M generated in each iteration. The most time demanding approach is SCRAMSAC, followed by NAPSAC. SCRAMSAC introduces a lot of overhead computing the sub-set $C_{red} \subseteq C$. This task requires to generate neighborhood subsets for each point correspondences. If the number of such correspondences is high, on the order of thousands, this is a very high time consuming task. After C_{red} is computed, SCRAMSAC performs similar to standard RANSAC. Similarly

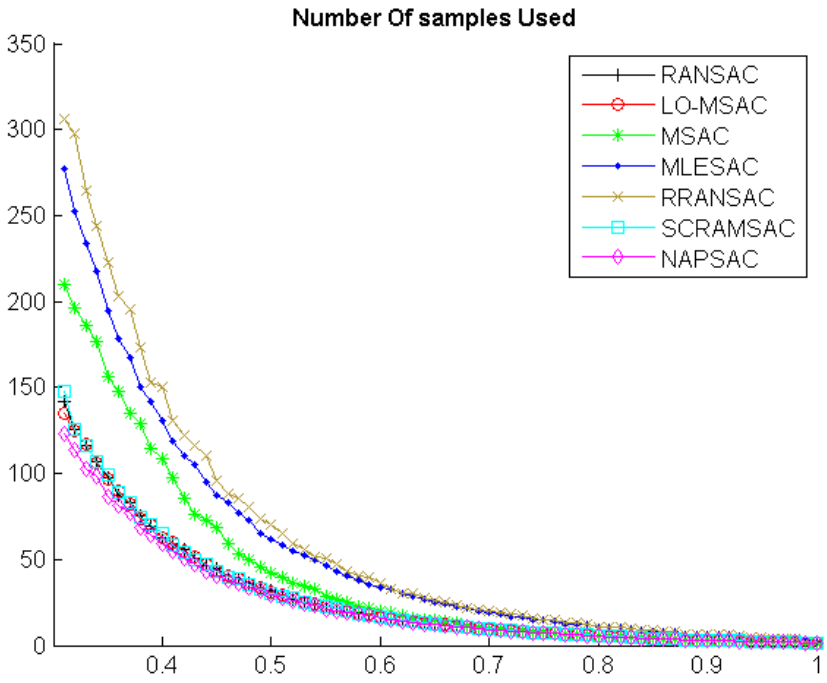

Figure B.3: Number of samples used given different values of inlier data .

to SCRAMSAC, NAPSAC also introduces an additional overhead compared with standard RANSAC due to the computation of neighborhood hyper spheres during iterations.

In addition, for each algorithm we evaluated the the accuracy of the generated models M, by measuring the normalized squared error of inliers (NSE) [148] defined in Equation B.14:

$$NSE = \frac{\sum_{di \epsilon D} Err(d_i;M)^2}{\sum_{di \epsilon D} Err(d_i;M^*)^2}, \tag{B.14}$$

where M and $M^*$are estimated and ground truth models respectively and D is the set of inliers i. NSE is close to 1 when the magnitude of the error of estimated model is near the magnitude of the error relative to the ground truth model.

Table B.3 shows the results obtained by tested approaches in relation with normalized inlier error. Most accurate results are obtained by approaches that includes

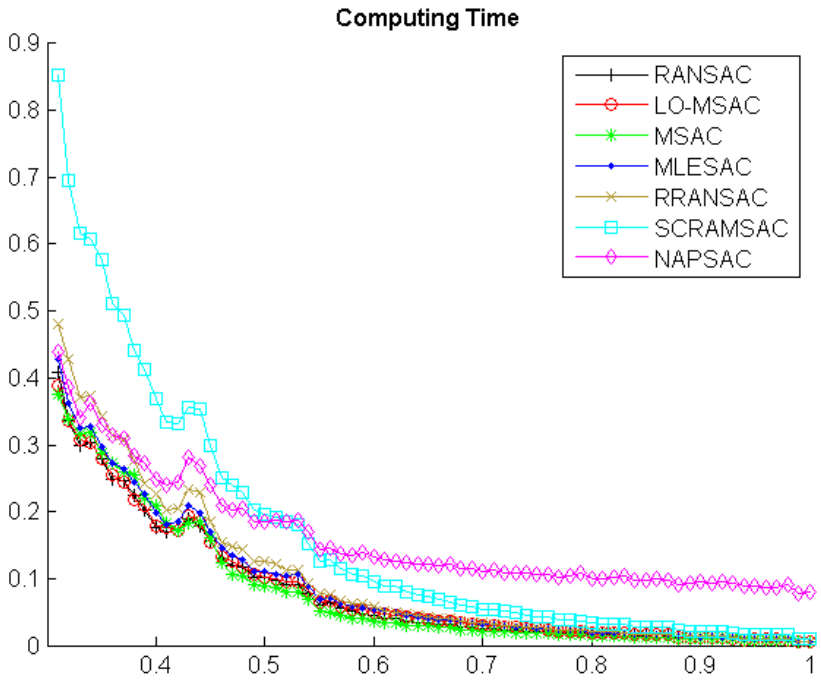

Figure B.4: Computation time given different values of inlier data. .

an optimization step during iteration, like Lo-RANSAC and MLESAC. The original version of RANSAC is the least accurate. The substitution of the loss function of RANSAC by an M-estimator in MSAC, improves the results without introducing addition computational overhead. It worth mentioning that only in applications where model accuracy is a critical factor, the differences between approaches could have a real impact.

B.4 Discussion and Conclusions

A first general conclusion is that all approaches are able to accurately estimate the true inlier ratio. The most significant difference regarding inlier ratio is when data is very noisy and inlier ratio is 40% or lower. In this scenario MLESAC, thanks to the gradient descendant optimization step is the most accurate.

Algorithm	NSE
RANSAC	4.041
LO-MSAC	3.603
MSAC	3.852
MLESAC	**3.598**
RRANSAC	3.973
NAPSAC	3.778
SCRAMSAC	3.960

Table B.3: Normalized Inlier Error.

As mentioned in [136] there is no reason to use RANSAC in preference to MSAC estimator because (1) the accuracy of estimations are similar or even better for MSAC, and (2) computation time is lower in most cases. If precision is the most important factor of our application, MLESAC and Lo-MSAC are the approaches that get lower errors, independently on the inlier ratio. However, the addition of a Expectation Maximization step after model generation with minimal sample set of cardinality m, introduces a computation overhead that can be cumbersome in some scenarios such as real-time tracking if the number of iterations for EM step is set too high. LO-MSAC gets similar accurate results compared to MLESAC at a very close computational cost.

NAPSAC shows improves over standard RANSAC, being able to converge to more accurate solutions using less samples, mainly over very noisy data, when inlier ratio is lower than 40%. However, NAPSAC adds a considerable computation overhead compared with other approaches such as MSAC or Randomized-RANSAC. NAPSAC can be effective in wide baseline scenarios where matches can be found between features detected anywhere along the image range. On small baseline matching problems, conversely, the outliers distribution will contain more structure and consequently NAPSAC will not be as effective.

Randomized-RANSAC heuristic approach does not show any real benefit over standard RANSAC. In fact, depending on the number of pre-evaluation samples d, the final number of samples needed before convergence can be significantly higher compared with RANSAC. This can be specially critical in high outlier ratio scenarios.

Guided sampling approaches such as SCRAMSAC are proposed to increase the probability of sampling good samples given a model M. In case of SCRAM-SAC the pre-processing or filtering of samples can introduce a significant compu-

tational overhead, if the number of correspondences is high. In addition, another drawback of SCRAMSAC is that it is only applicable to a fraction of robust estimation problems based on corresponding points, because it needs that those points include some type of scale estimation. As described in Chapter 3 there are several approaches, such as FAST [4], that do not perform scale estimation.

Bibliography

[1] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, pp. 43–72, 2005, 10.1007/s11263-005-3848-x. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-3848-x

[2] ——, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1, pp. 43–72, 2005.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[4] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 105–119, 2010.

[5] P. Alcantarilla, A. Bartoli, and A. Davison, "Kaze features," in *Eur. Conf. on Computer Vision (ECCV)*, 2012.

[6] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[7] A. Kadyrov and M. Petrou, "The trace transform and its applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 8, pp. 811–828, 2001.

[8] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock, "Napsac: high noise, high dimensional robust estimation," in *In BMVC02*, 2002, pp. 458–467.

[9] M. Chen, "Leveraging the asymmetric sensitivity of eye contact for videoconference," in *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2002, pp. 49–56.

[10] T. Tam, J. A. Cafazzo, E. Seto, M. E. Salenieks, and P. G. Rossos, "Perception of eye contact in video teleconsultation," *Journal of telemedicine and telecare*, vol. 13, no. 1, pp. 35–39, 2007.

[11] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010.

[12] G. Liew and J. J. Wang, "Retinal vascular signs: A window to the heart?" *Revista Española de Cardiología (English Edition)*, vol. 64, no. 6, pp. 515–521, 2011.

[13] D. Y. Leung, C. C. Tham, F. C. Li, Y. Y. Kwong, S. C. Chi, and D. S. Lam, "Silent cerebral infarct and visual field progression in newly diagnosed normal-tension glaucoma: a cohort study," *Ophthalmology*, vol. 116, no. 7, pp. 1250–1256, 2009.

[14] I. Leichter, M. Lindenbaum, and E. Rivlin, "Mean shift tracking with multiple reference color histograms," *Computer Vision and Image Understanding*, vol. 114, no. 3, pp. 400–408, 2010.

[15] D. A. R. Vigo, F. S. Khan, J. Van De Weijer, and T. Gevers, "The impact of color on bag-of-words based object recognition," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 1549–1553.

[16] I. Barandiaran, C. Cortes, M. Nieto, M. Grana, and O. Ruiz, "A new evaluation framework and image dataset for key point extraction and feature descriptor matching," in *VISAPP 2013 - International Conference on Computer Vision Theory and Applications*. Scitepress, 2013, pp. 252 – 257.

[17] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 772–779.

[18] M. Hu, G. Penney, M. Figl, P. Edwards, F. Bello, R. Casula, D. Rueckert, and D. Hawkes, "Reconstruction of a 3d surface from video that is robust to missing data and outliers: Application to minimally invasive surgery using stereo and mono endoscopes," *Medical image analysis*, vol. 16, no. 3, pp. 597–611, 2012.

[19] D. Van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *International Journal of Virtual Reality*, vol. 9, no. 2, p. 1, 2010.

[20] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988.

[21] T. L. Lindeberg, "On scale selection for differential operators," in *Proc. 8th Scandinavian Conf. on Image Analysis*, 1993, pp. 857–866.

[22] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *Computer Vision,ECCV 2002*, pp. 128–142, 2002.

[23] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[24] M. Agrawal, K. Konolige, and M. Blas, "Censure: Center surround extremas for realtime feature detection and matching," *Computer Vision–ECCV 2008*, pp. 102–115, 2008.

[25] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of computer vision*, vol. 37, no. 2, pp. 151–172, 2000.

[26] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[27] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2657–2664.

[28] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[29] I. Macía, "Medical image analysis for the detection, extraction and modelling of vascular structures," Ph.D. dissertation, University of the Basque Country, 2012.

[30] T. Lindeberg, *Scale-space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[31] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[32] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 525–531.

[33] T. Lindeberg, "Scale-space for discrete signals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 234–254, 3 1990.

[34] A. Witkin, "Scale-space filtering," *Proc. 8th Int. Joint Conf. Art. Intell.*, pp. 1019–22, 1983.

[35] J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.

[36] D. Lowe, "Organization of smooth image curves at multiple scales," *International Journal of Computer Vision*, vol. 3, no. 2, pp. 119–130, 1989.

[37] T. Lindeberg, "Feature detection with automatic scale selection," *International journal of computer vision*, vol. 30, no. 2, pp. 79–116, 1998.

[38] A. Witkin, "Scale-space filtering," *Readings in computer vision: issues, problems, principles, and paradigms*, pp. 329–332, 1987.

[39] ——, "Scale-space filtering: A new approach to multi-scale description," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, vol. 9. IEEE, 1984, pp. 150–153.

[40] J. Congote, I. Barandiaran, J. Barandiaran, T. Montserrat, J. Quelen, C. Ferran, P. Mindan, O. Mur, F. Tarres, and O. Ruiz, "Real-time depth map generation architecture for 3d videoconferencing," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*. IEEE, 2010, pp. 1–4.

[41] F. Bellavia, "Matching image features," Ph.D. dissertation, Universitá Degli Studi Di Palermo, 2011.

[42] M. Brusco, M. Andreetto, A. Giorgi, and G. M. Cortelazzo, "3d registration by textured spin-images," in *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*. IEEE, 2005, pp. 262–269.

[43] A. B. L. Larsen, S. Darkner, A. L. Dahl, and K. S. Pedersen, "Jet-based local image descriptors," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 638–650.

[44] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Efficient gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, 1996.

[45] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[46] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[47] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod, "Compressed histogram of gradients: A low-bitrate descriptor," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 384–399, 2012.

[48] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2006.

[49] J. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[50] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[51] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 603–610.

[52] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[53] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2681–2684.

[54] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications (VISSAPP)*, 2009, pp. 331–340.

[55] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–506.

[56] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[57] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1352–1364, 2008.

[58] E. Montijano and C. Sagues, "Fast pose estimation for visual navigation using homographies," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 2704–2709.

[59] K. Okuma, J. J. Little, and D. G. Lowe, "Automatic rectification of long image sequences," in *Asian Conference on Computer Vision*, 2004.

[60] A. Zisserman and R. Hartley, "Multiple view geometry in computer vision," *Ed. Cambridge*, 2000.

[61] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. the MIT Press, 1993.

[62] G. Hughes and M. Chraibi, "Calculating ellipse overlap areas," 2011.

[63] H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," *tech report CMURITR8003 Robotics Institute Carnegie Mellon University doctoral dissertation Stanford University*, 1980.

[64] T. Lindeberg and J. Gårding, "Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure," *Image and vision computing*, vol. 15, no. 6, pp. 415–434, 1997.

[65] M. Trajković and M. Hedley, "Fast corner detection," *Image and Vision Computing*, vol. 16, no. 2, pp. 75–87, 1998.

[66] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[67] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012.

[68] ——, "Scale selection properties of generalized scale-space interest point detectors," *Journal of Mathematical Imaging and Vision*, pp. 1–34, 2012.

[69] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Medical Image Computing and Computer-Assisted Interventation*. Springer, 1998, pp. 130–137.

[70] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *International journal of computer vision*, vol. 59, no. 1, pp. 61–85, 2004.

[71] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[72] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[73] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[74] M. Brown and D. G. Lowe, "Invariant features from interest point groups," in *British Machine Vision Conference, Cardiff, Wales*, vol. 21, no. 2, 2002, pp. 656–665.

[75] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[76] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

[77] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 850–855.

[78] M. Calonder, V. Lepetit, and P. Fua, "Keypoint signatures for fast learning and recognition," *Computer Vision–ECCV 2008*, pp. 58–71, 2008.

[79] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[80] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[81] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," *Computer Vision–ECCV 2010*, pp. 183–196, 2010.

[82] R. Duits, L. Florack, J. De Graaf, and B. ter Haar Romeny, "On the axioms of scale space theory," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 3, pp. 267–298, 2004.

[83] E. J. Pauwels, L. J. Van Gool, P. Fiddelaers, and T. Moons, "An extended class of scale-invariant and recursive scale space filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 7, pp. 691–701, 1995.

[84] J. Weickert, B. T. H. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *Image Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 398–410, 1998.

[85] I. G. Olaizola, I. Barandiaran, B. Sierra, and M. Graña, "Ditec: Experimental analysis of an image characterization method based on the trace transform," in *VISAPP 2013 - International Conference on Computer Vision Theory and Applications*. Scitepress, 2013, pp. 344–352.

[86] A. Baddeley and M. Van Lieshout, "Stochastic geometry models in high-level vision," *Journal of Applied Statistics*, vol. 20, no. 5-6, pp. 231–256, 1993.

[87] N. Fedotov and A. A. Kadyrov, "Image scanning in machine vision leads to new understanding of image," in *Digital Image Processing and Computer Graphics: Fifth International Workshop*. International Society for Optics and Photonics, 1995, pp. 256–261.

[88] K. Jafari-Khouzani and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 6, pp. 1004–1008, 2005.

[89] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *The International Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, May 1987.

[90] P. Toft, "The radon transform - theory and implementation," Ph.D. dissertation, Department of Mathematical Modelling, Technical University of Denmark, June 1996.

[91] A. Kassim, T. Tan, and K. Tan, "A comparative study of efficient generalised hough transform techniques," *Image and vision computing*, vol. 17, no. 10, pp. 737–748, 1999.

[92] M. Meena, K. Pramod, and K. Linganagouda, "Optimized trace transform based feature extraction architecture for cbir," in *ACC (3)'11*, 2011, pp. 444–451.

[93] T. Durrani and D. Bisset, "The radon transform and its properties," *Geophysics*, vol. 49, no. 8, pp. 1180–1187, 1984.

[94] F. Schaffalitzky and A. Zisserman, "Viewpoint invariant texture matching and wide baseline stereo," in *Computer Vision, 2001. ICCV 2001. Proceed-*

ings. Eighth IEEE International Conference on, vol. 2. IEEE, 2001, pp. 636–643.

[95] I. Barandiaran, C. Paloc, and M. Graña, "Real-time optical markerless tracking for augmented reality applications," *Journal of Real-Time Image Processing*, vol. 5, no. 2, pp. 129–138, 2010.

[96] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25 –30, 1965.

[97] J. Wood, "Invariant pattern recognition: a review," *Pattern recognition*, vol. 29, no. 1, pp. 1–17, 1996.

[98] D. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," *Environmental Monitoring and Assessment*, vol. 32, no. 2, pp. 135–154, 1994.

[99] L. Bauer and D. Hamby, "Relative sensitivities of existing and novel model parameters in atmospheric tritium dose estimates," *Radiation protection dosimetry*, vol. 37, no. 4, pp. 253–260, 1991.

[100] J. Morel and G. Yu, "Is sift scale invariant?" *Inverse Problems and Imaging*, vol. 5, no. 1, pp. 115–136, 2011.

[101] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 9, pp. 1465–1479, 2006.

[102] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[103] B. A. Olshausen and D. J. Field, "What is the other 85% of v1 doing," *Problems in Systems Neuroscience*, pp. 182–211, 2004.

[104] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International journal of computer vision*, pp. 1–26, 2011.

[105] J. Fischer, A. Ruppel, F. Weißhardt, and A. Verl, "A rotation invariant feature descriptor o-daisy and its fpga implementation," in *Intelligent Robots and*

Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, 2011, pp. 2365–2370.

[106] *Intel® 64 and IA-32 Architectures Software Developer Manual.*

[107] V. Lepetit and P. Fua, *Monocular-Based 3D Tracking of Rigid Objects*. Now Pub, 2005.

[108] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*. IEEE, 1999, pp. 85–94.

[109] B. Williams, G. Klein, and I. Reid, "Real-time slam relocalisation," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. Ieee, 2007, pp. 1–8.

[110] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3d camera tracking," in *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*. IEEE, 2004, pp. 48–56.

[111] V. Lepetit, "On computer vision for augmented reality," in *Ubiquitous Virtual Reality, 2008. ISUVR 2008. International Symposium on*. IEEE, 2008, pp. 13–16.

[112] A. Boffy, Y. Tsin, and Y. Genc, "Real-time feature matching using adaptive and spatially distributed classification trees," in *British Machine Vision Conference*, 2006.

[113] M. Ozuysal, V. Lepetit, F. Fleuret, and P. Fua, "Feature harvesting for tracking-by-detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 592–605.

[114] V. Lepetit, J. Pilet, and P. Fua, "Point matching as a classification problem for fast and robust object pose estimation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–244.

[115] I. Barandiaran, C. Cottez, C. Paloc, and M. Grana, "Random forest classifier for real-time optical markerless tracking," *Proc. VISAPP*, vol. 2, pp. 559–564, 2008.

[116] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[117] ——, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[118] X. Hu, "Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 233–240.

[119] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1619–1630, 2006.

[120] I. Barandiaran, C. Cottez, C. Paloc, and M. Graña, "Comparative evaluation of random forest and fern classifiers for real-time feature matching," in *Proceedings of WSCG*, 2008, pp. 59–166.

[121] G. Reitmayr and D. Schmalstieg, "Opentracker: A flexible software design for three-dimensional interaction," *Virtual Reality*, vol. 9, no. 1, pp. 79–92, 2005.

[122] A. Davison, W. Mayol, and D. Murray, "Real-time localization and mapping with wearable active vision," in *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*. IEEE, 2003, pp. 18–27.

[123] D. Marimon, A. Bonnin, T. Adamek, and R. Gimeno, "Darts: Efficient scale-space extraction of daisy keypoints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2416–2423.

[124] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[125] K. Mikolajczyk, T. Tuytelaars, C. Schmid *et al.*, "Affine covariant features," *Collaborative work between: the Visual Geometry Group, Katholieke Universiteit Leuven, Inria Rhone-Alpes and the Center for Machine Perception*, 2007.

[126] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[127] F. Bellavia, D. Tegolo, and E. Trucco, "Improving sift-based descriptors stability to rotations," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*. IEEE Computer Society, 2010, pp. 3460–3463.

[128] F. Fraundorfer and H. Bischof, "A novel performance evaluation method of local detectors on non-planar scenes," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*. IEEE, 2005, pp. 33–33.

[129] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International journal of computer vision*, pp. 1–26, 2011.

[130] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[131] J. A. Noble, "Descriptions of image surfaces," Ph.D. dissertation, University of Oxford, 1989.

[132] R. Gendron, "An improved objective method for rating picture sharpness: Cmt acutance," *Journal of the SMPTE*, vol. 82, no. 12, pp. 1009–1012, 1973.

[133] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[134] M. Zuliani, "Ransac for dummies," Tech. Rep., Nov. 2008.

[135] S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," in *Proceedings–2009 British Machine Vision Conference*, 2009, pp. 1–12.

[136] P. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[137] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," *Pattern Recognition*, pp. 236–243, 2003.

[138] P. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002.

[139] B. Tordoff and D. Murray, "Guided-mlesac: Faster image transform estimation by using matching priors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1523–1535, 2005.

[140] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. Ieee, 2005, pp. 220–226.

[141] K. Ni, H. Jin, and F. Dellaert, "Groupsac: Efficient consensus in the presence of groupings," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2193–2200.

[142] T. Sattler, B. Leibe, and L. Kobbelt, "Scramsac: Improving ransac's efficiency with a spatial consistency filter," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2090–2097.

[143] P. H. Torr, A. Zisserman, and S. J. Maybank, "Robust detection of degenerate configurations while estimating the fundamental matrix," *Computer Vision and Image Understanding*, vol. 71, no. 3, pp. 312–333, 1998.

[144] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British machine vision conference*, vol. 1, 2002, pp. 384–393.

[145] R. I. Hartley, "In defense of the eight-point algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 6, pp. 580–593, 1997.

[146] O. Chum and J. Matas, "Randomized ransac with td, d test," in *Proc. British Machine Vision Conference*, vol. 2, 2002, pp. 448–457.

[147] T. K. Moon, "The expectation-maximization algorithm," *Signal Processing Magazine, IEEE*, vol. 13, no. 6, pp. 47–60, 1996.

[148] S. Choi and J. Kim, "Robust regression to varying data distribution and its application to landmark-based localization," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2008, pp. 3465–3470.