Faculdade de Engenharia da Universidade do Porto

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

A Vision-based Approach Towards Robust Localization for Intelligent Wheelchairs

Marcelo Roberto Petry

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor in Informatics Engineering
of the University of Porto

2013, June

A Vision-based Approach Towards Robust Localization for Intelligent Wheelchairs

Marcelo Roberto Petry

INESC Technology and Science and
Artificial Intelligence and Computer Science Laboratory
Doctoral Program in Informatics Engineering
Faculty of Engineering, University of Porto
Porto, Portugal

Advisors:
Professor Dr. António Paulo Moreira
Professor Dr. Luís Paulo Reis

Financial support provided by the Foundation for Science and Technology (FCT) and by the European Social Fund (ESF) through the Human Potential Operating Programme (POPH) FCT/SFRH/BD/60727/2009. This work was also financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by FCT within projects «INTELLWHEELS - Intelligent Wheelchair with Flexible Multimodal Interface, RIPD/ADA/109636/2009» and «FCOMP-01-0124-FEDER-022701».

Resumo

Ao longo das últimas décadas a sociedade tem demonstrado cada vez mais preocupação em promover a inclusão social de pessoas com deficiência. Para isto, mobilidade tem uma importância fundamental uma vez que está diretamente relacionada com a independência de um indivíduo. Equipamentos tradicionais para auxílio de mobilidade como cadeiras de rodas, muletas, bengalas e membros artificias têm capacidade de auxílio limitado e em muitos casos não são capazes de prover o auxílio necessário para indivíduos que possuam combinações de deficiências físicas, cognitivas e de percepção. Neste sentido, cadeiras de rodas inteligentes são tecnologias que podem aumentar a autonomia e independência desta população e, atualmente, são objeto de estudos por vários grupos de pesquisa. Esta tese foi desenvolvida no contexto do Projeto FCT/RIPD/ADA/109636/2009 – "IntellWheels – Cadeira de Rodas Inteligente com Interface Multimodal Flexível", e concentrou-se no estudo, projeto e implementação de metodologias que auxiliem no desenvolvimento de cadeiras de rodas mais robustas e inteligentes. As principais contribuições deste trabalho podem ser divididas em três áreas distintas: robótica de assistência, visão por computador e localização robótica.

Inicialmente o trabalho descreve os principais conceitos do Projeto IntellWheels. A seguir, propõe uma metodologia de controle compartilhado baseado na ideia de que a cadeira de rodas encontra-se imersa em um campo de forças potenciais. Posteriormente, avaliam-se alguns dos simuladores robóticos mais populares com o objetivo de identificar o que possui características mais adequadas para simulação do protótipo IntellWheels. A última contribuição na área de robótica de assistência é o projeto de um *kit* de *harware* que seja capaz de mitigar o impacto visual causado pela integração de sensores e atuadores na cadeira de rodas. Os resultados experimentais demonstraram que a metodologia de controle compartilhado foi capaz de reduzir o número de colisões em mais de 75%. A avaliação dos simuladores robóticos indicou que o simulador USARSim foi o que apresentou o conjunto de características que melhor se adequa aos requisitos do projeto IntellWheels. Por fim, a análise estatística de uma pesquisa de opinião sugeriu que o protótipo proposto foi eficaz na atenuação dos impactos visuais e ergonômicos causados pelos dispositivos adicionados à cadeira de rodas.

Com respeito à área de visão computacional, este trabalho apresentou duas abordagens para aumentar a invariância à iluminação do algoritmo para detecção de *features* SURF. As abordagens propostas tiram respectivamente vantagens de normalização local e descritores baseados em *local space average color* para detectar *features* invariantes à iluminação. Demonstraram-se, através de uma análise teórica, os efeitos de diversas variações de iluminação na resposta de algoritmos para detecção de *features* populares (Harris corners, SIFT e SURF) e como as metodologias propostas podem corrigir estes efeitos. Os testes e experiências realizados demonstraram a eficácia das abordagens propostas, aumentando a repetibilidade das features detectadas em cenas com grandes variações de iluminação em 2.4% com o algoritmo LN SURF e 41.69% com o algoritmo LSAC SURF.

Finalmente, o trabalho discute como metodologias baseadas em visão podem aperfeiçoar as estimativas de localização, especialmente em robôs com características particulares como as cadeiras de rodas inteligentes. A abordagem proposta é baseada em técnicas de odometria visual e câmeras RGB-D de baixo custo. O algoritmo localiza pontos salientes na imagem e utiliza a informação de profundidade de cada pixel para estimar a translação e rotação do robô em cada frame. A análise dos resultados mostrou um erro de posicionamento inferior a 2% para os casos testados, demonstrando a aplicabilidade do algoritmo de localização em sistemas de navegação de robôs e cadeiras de rodas inteligentes.

Abstract

Over the last decades society is more and more concerned to promote the social inclusion of impaired individuals. For that, mobility plays an important role since the amount of independence that a person can achieve is closely related to how independently mobile this person is. Traditional mobility aid devices (e.g. wheelchairs, crutches, canes) are limited, and usually can not provide the assistance required by individuals with combinations of physical and cognitive or perceptual impairments. In this sense, intelligent wheelchairs (IW) are devices that can increase the autonomy and independence of this population. This thesis was developed in the context of Project "IntellWheels – Intelligent Wheelchair with Flexible Multimodal Interface", and concerned with the study, design and implementation of methodologies to support the development of more robust and intelligent wheelchairs. The main contributions of this work can be divided into three distinct areas: assistive robotics, computer vision, and robot self-localization.

First, we describe the main concepts regarding the IntellWheels project. Next, we propose a shared control methodology based on the idea that the wheelchair is immersed in a field of potential forces. Further, we evaluate some of the most popular general robotics simulators in order to identify which one is more adequate to simulate the project prototype. Our last contribution in the area of assistive robotics is a hardware design that aims at reducing the visual impact caused by the assemblage of sensor and actuators in the wheelchair. Experimental results demonstrated that the shared control methodology was able to reduce the number of collisions in more than 75%. The assessment of popular robotic simulators indicated that USARSim was the simulator whose features better matched the IntellWheels project requirements. Next, the statistical analysis of a public opinion assessment suggested that IntellWheels design was effective to mitigate the visual and ergonomic impacts caused by the addition of its sensorial and processing capabilities.

Regarding the computer vision area, we presented two approaches to increase the illumination invariance of SURF feature detection. The algorithms respectively take advantage of local normalization and local space average color descriptor to detect illumination invariant features. We performed a theoretical analysis demonstrating the effects of distinct photometric variations on the response of popular image features detectors (Harris corners, SIFT and SURF), and how our proposed methodologies can amend those effects. Experimental results demonstrated the effectiveness of the proposed approaches, improving the median repeatability of the features detected in scenes with large photometric variations in 2.4% for the LN SURF algorithm and 41.69% for the LSAC SURF algorithm.

Finally, we discussed how vision-based methodologies can improve robotic localization estimations, specially in robots with particular characteristics like intelligent wheelchairs. We also present a visual odometry approach based on inexpensive RGB-D cameras. The algorithm localizes visually salient points, and uses depth information of each pixel to estimate the robot translation and rotation updates at each frame. Experimental results showed a relative position error around 2%, demonstrating the applicability of the localization algorithm in the navigation system of intelligent wheelchairs.

Acknowledgements

First of all, I doubt I can properly express my gratitude to Prof. Dr. António Paulo Moreira. Moreira has been a great source of inspiration and I am very thankful for his continuous encouragement, support, patience, and enthusiasm. I am also greatly indebted to my co-advisor Prof. Dr. Luis Paulo Reis. His supportive attitude, vast scientific knowledge, capacity, and sympathy have been of utmost importance for finishing my PhD study.

I would like to thank the members of my steering committee, Prof. Dr. Artur Pereira and Prof. Dr. Paulo Costa, for providing me with valuable feedback on this thesis. Many thanks to the all members of the IntellWheels project, in special to Rodrigo A. M. Braga, Frederico Cunha, Sérgio Vasconcelos, Brígida Mónica Faria, João Couto Soares, Abbas Abdolmaleki and Prof. Dr. Nuno Panelas Lau. Moreover, I acknowledge the (current and former) members of the Artificial Intelligence and Computer Science Laboratory (LIACC), in particular Rosaldo Rossetti, João Almeida, Zafeiris Kokkinogenis, Lucio Passos and Nima Shafii, whose scientific discussions were certainly a plus.

Concerning the non-academic side of my life, I have to thank to a number of people for their friendship. I will not name you all, but I guess you all know who you are anyway. Yet, I have to write a special thanks to my two "sons" Leonardo Bremermann and Daniel Faria, and to the best neighbors in the block Thuane Roza and Guilherme Schmitt. I have also to acknowledge my two "brothers" Mateus Ferla and Daniel Correa for keeping our friendship updated even at 9000 Km apart.

A word (in Portuguese) of gratitude to my family: Irene, Jacob, Viviane and Rogerio. Este é certamente um dos projetos mais importantes da minha vida. Sem o vosso incondicional apoio e confiança esta tese nunca teria sido possível. As minhas desculpas por me ter ausentado por tanto tempo.

Last, but certainly not least, I would like to thank my best friend and fiancée Eluana. You made a fundamental contribution to the development of this work. If I managed to finish this thesis, it is greatly due to your support and comprehension.

Marcelo R. Petry

*“The value of things is not in the time they last, but the intensity
with which they occur. So there are unforgettable
moments, inexplicable things and
incomparable people.”*

Fernando Pessoa

Contents

List of Figures

List of Tables

Abbreviations

2D	Two-dimensions
3D	Three-dimensions
ACCoMo	Autonomous, Cooperative, COllaborative MObile robot
ACL	Agent Communication Language
AI	Artificial Intelligence
ALOI	Amsterdam Library of Object Images
ALCC	ALOI illumination color collection
API	Application Program Interface
AR	Augmented Reality
ATE	Absolute Trajectory Error
CAN	Controller Area Network
CGM	Cubical Gamut Mapping
CIE	Commission Internationale de l'Eclairage
CiP	Color in Perspective
CORBA	Common Object Request Broker Architecture
CRULE	Coefficient-Rule
DεC	Divide and Conquer
DM	Diagonal Model
DoM	Diagonal-offset Model
DoF	Degrees of freedom
DoG	Difference of Gaussians
EDA	exploratory data analysis
EKF	Extended Kalman Filter
FCT	Foundation for Science and Technology
FEUP	Faculty of Engineering of the University of Porto
FIPA	Foundations of Intelligent Physical Agents
FOV	Field of View
FRIEND	Functional Robot arm with user-frIENdly interface for Disabled people
GCIE	Gamut Constrained Illumination Estimation
GUI	Graphical User Interface
GAL	Global Agent List
GLONASS	*Globalnaya Navigatsionnaya Sputnikovaya Sistema*
GNSS	Global Navigation Satellite System
GP	Global Positioning
GPS	Global Positioning System
GW	Gray World
HMI	Human Machine Interface

HSI	Hue, Saturation, Intensity
HSL	Hue, Saturation, Lightness
HSV	Hue, Saturation, Value
ICP	Iterative closest point
IDP	Inverse Depth Parameterization
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
INESC-Porto	Institute for Systems and Computer Engineering of Porto
INESC TEC	INESC Technology and Science
IntellWheels	Intelligent Wheelchair with Flexible Multimodal Interface
IntellSim	IntellWheels Simulator
IW	Intelligent Wheelchair
JADE	Java Agent DEvelopment Framework
LAL	Local Agents List
LIACC	Artificial Intelligence and Computer Science Laboratory
LCC	Light Color Change
LCCS	Light Color Change and Shift
LICS	Light Intensity Change and Shift
LIS	Light Intensity Shift
LME	Local Motion Estimation
LSAC	Local Space Average Color
MAS	Multi-Agent System
MMI	Multi Modal Interface
MR	Mixed Reality
nRGB	Normalized RGB
PDDL	Planning Domain Definition Language
PF	Potential Field
RANSAC	Random Sample Consensus
RDS	Microsoft Robotics Developer Studio
RFID	Radio Frequency IDentifier
RMSE	Root Mean Square Error
RPE	Relative Pose Error
SBM	Scale by Max
SD	Standard Deviation
SFM	Structure From Motion
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Feature
sRGB	Standard RGB
USARSim	Unified System for Automation and Robot Simulation
VO	Visual Odometry
VSLAM	Visual Simultaneous Localization and Mapping
WP	White Patch
WWW	*World Wide Web*

Chapter 1

Introduction

"Not everything that is faced can be changed, but nothing can be changed until it is faced."

– James Baldwin

1.1 Context

Physical disability is the general term applied to a group of disabling symptoms that causes limitations over the control of voluntary muscles. The term thus refers to a broad range of impairments that can be originated at any stage of the human life cycle. During the prenatal stage, period that extends from conception to the time of birth, the causes of disabilities can involve chromosomal abnormalities (loss, gain, or exchange of genetic material from a chromosome pair), genetic abnormalities (genes that create damaging biomedical conditions), or result from the prenatal environment within the uterus (external agents, infections, toxins, and maternal health). During the perinatal stage, time period immediately before and after birth, causes of disabilities are related to prematurity, injury, prolonged oxygen deprivation, and infections contracted by the baby in the birth canal (e.g. syphilis, gonorrhea and herpes). After birth, the main causes of disabilities vary according to the age of the individuals, but are essentially related to injuries caused by accidents (amputation, traumatic brain injury, spinal cord injury, etc.), exposure to chemicals and drugs, and illness (muscular dystrophy, multiple sclerosis, cerebral palsy, etc.).

Nowadays, society is more and more concerned to promote the social inclusion of impaired individuals. For that, mobility plays an important role, since the amount of independence that a person can achieve is closely related to how independently mobile they are. In addition to independence and self-esteem, some studies reveal that mobility can have positive psychosocial and cognitive development of physically disabled children, with beneficial effects in the development and rehabilitation of children with disabilities [1, 2, 3].

Physically impaired people often rely upon assistive devices such as wheelchairs, crutches, canes, and artificial limbs to increase, maintain, or improve their functional capabilities. However, a generalization of the treatment and assistance strategies is hardly achieved since each patient shows a different combination of symptoms and levels of motor control. Another complication is that, in many cases, motor disabilities come associated with cognitive and sensorial impairment, which often lead to driving/navigational problems even when motor impairments are not severe. Therefore, there is a growing demand for intelligent and safer assistive devices. To accommodate users who find operating standard mobility devices difficult or impossible, researchers have used technologies originally developed for mobile robots to create intelligent wheelchairs [4].

In an attempt to address some of these issues, the Faculty of Engineering of the University of Porto (FEUP) in collaboration with the Artificial Intelligence and Computer Science Laboratory (LIACC), the INESC Technology and Science associated Laboratory (INESC TEC), the Institute of Electronics and Telematics Engineering of Aveiro (IEETA), the School of Allied Health Sciences of the Polytechnic Institute of Porto (ESTSP), the University of Minho (UMINHO) and the Portuguese Association of Cerebral Palsy (APPC) have developed the project IntellWheels. This thesis is inserted in the IntellWheels project and addresses problems related to the design and self-localization of intelligent wheelchairs.

1.2 Motivation

The human idea of machines capable of executing different and complicated tasks remounts from ancient mythology. In fact, stories about artificial people acting as mechanical servants can be traced back to Greeks and Romans. Since then, countless passages describe fictional robotic characters in the literature and more recently in television and films. Beyond their capacity to solve problems and to react over the environment to achieve their goals, common to these fictional stories is how robots appear to navigate with an effortless and vast accuracy.

Unlike fiction, real mobile robot's navigation is a difficult research problem, in part because it involves practically everything about robotics: sensing, acting, planning, design, etc. As described by Murphy [5], the problem of navigation can be summarized into answering three questions "What's the best way?", "Where have I been?" and "Where am I?". Answers to the first and the second questions are related with path planning, mapping and tracking, while to the last one are related to the robot localization. Indeed, answering the robot position and attitude in truly autonomous fashion is a challenge that remains nowadays in indoor non-structured environments, specially when dealing with budget and assemblage restrictions.

Self-localization is essential for mobile robots since it is required at several levels of the system. Planners usually express a set of actions in terms of localization, for example "go to that position", or "turn x degrees". Mapping algorithms usually combine the relative information provided by proximity sensors (range, bearing) with the current robot's pose estimation to build and update global maps of the environment. Controllers require the positioning feedback to correct the execution of the trajectories provided by motion planners. Robust localization is even more important in fields like autonomous transportation, assistive robotics and rehabilitation robotics, in which robots are expected to not only coexist but also to actively interact with human beings.

Traditionally, mobile robots make use of wheel odometry to assist other absolute position measurements and provide better and more reliable position estimation. However, since the fundamental idea of odometry is the integration of incremental motion information, it inevitably leads to the accumulation of errors. Particularly, the accumulation of orientation errors implies large positioning errors, which increase proportionally with the distance traveled by the robot. Wheel odometry also assumes that all wheel revolutions can be translated into linear displacement relative to the floor, which may not be entirely true due to systematic and non-systematic errors [6]. Systematic errors are predictable, and caused specifically by the vehicle due to imperfections in its design or in its mechanical implementation. Common sources of systematic errors are unequal wheel diameter, differences between the actual wheel diameter and the nominal wheel diameter. Additional systematic sources of errors include the misalignment of wheels, finite encoder resolution, finite encoder sampling rate, and difference between the actual wheelbase and the nominal wheelbase. Non-systematic errors, on the other hand, are "imposed" to the vehicle through unpredictable characteristics of the environment. These errors occur when the wheel rotates more than the predicted, for example when the wheel is forced to travel up or down some irregularity. The most common non-systematic errors are caused by traveling over unexpected objects and uneven floors, and by wheel-slippage due to slippery floors, over acceleration, fast turning, interaction with external bodies, and internal forces like the castor wheels.

In general, the literature refers that systematic errors are particularly severe because they accumulate constantly. Such idea was disseminated over the years because most robots were designed to operate in smooth indoor surfaces, scenario in which systematic errors indeed contribute much more to odometry errors than non-systematic errors. In addition to that, until the work of Borestein [6], there was not a practical method for reducing odometry errors caused by kinematic imperfections of a mobile robot. Nowadays, on the other hand, a significant part of the robots are designed to operate in harsher environments, with rough surfaces and significant irregularities. In these scenarios, though, non-systematic errors are dominant and much more severe because they cannot be corrected through calibration. This is the case of intelligent wheelchairs whose encoders are coupled directly in the motor shaft. Unlike other robots that are equipped with solid wheels, the vast majority of the tires used in wheelchairs are pneumatic inflatable structures, comprising a donut-shaped body of cords and wires encased in rubber and filled with compressed air. Air filled tires compress differently according to the weight of the users, yielding a variation in the mean diameter of the wheels. Furthermore, the position of the user over the wheelchair seat may compress the tires differently, which produces unpredictable asymmetric load distribution and unequal wheel diameters. In wheelchairs, the wheelbase is especially hard to measure because rubber tires do not present a single contact point with the floor (due to deformation and mechanical characteristics). Due to its unpredictability, such errors, that would be classified as systematic in other robots, must be considered as non-systematic in intelligent wheelchairs, and thus can not be minimized with calibration methodologies.

Therefore, in spite of its simplicity, wheeled odometry methodologies tend to be highly inaccurate. Indeed, assistive robots need a method for accurately tracking their pose in order to navigate safely over long distances, uneven surfaces and with asymmetric load distribution. Other methods based on map matching capture the world through sensors like ultrasounds and infrared, but are subjected to misreading due to concealment, possible confusion with other robots nearby, or due to reflectivity and color variations [7].

Other robots rely on Global Navigation Satellite Systems (GNSS), such as the United States Global Positioning System (GPS), the European Galileo and the Russian *Globalnaya Navigatsionnaya Sputnikovaya Sistema* (GLONASS). The accuracy of these systems are, however, highly dependent on the position and the number of satellites tracked. GNSS positional data provides low accuracy in regions where satellites are occluded by the landscape or buildings. The accuracy of GNSS systems are also dependent on the technology they rely. RTK GPS provides 2 cm accuracy, Differential GPS provides sub-meter accuracy, while regular GPS sensors provide only a weak 10m accuracy [8].

In this context, vision is an alternative. The use of cameras is recognized for its unique advantage to deliver multi-layered information. Contrary to navigational sensors such as GNSS and Inertial Measurement Units (IMU), which provide information only about the vehicle's own motion with respect to the inertial frame, vision can provide additional information relative to the environment. For example, with vision is possible to estimate how close the vehicle is to an obstacle, whether targets appear in the environment or how the vehicle is aligned with the horizon. Unlike GPS, which stops working in the shadow of satellite visibility, vision works in cluttered indoor environments as long as the captured images have sufficient texture and illumination. Additionally, different from shaft encoders, vision-based localization methodologies do not depend on wheel-terrain interactions, eliminating several sources of systematic and non-systematic errors [7, 9, 10]. At last, the inherent ability of stereo cameras to gather 3D information of the environment allows us to express the robot state in the full 6 degrees of freedom (DoF) of the Euclidean motion model [5, 11, 12].

The constant reduction in the size of robotic platforms leaded to the necessity of changing from high power consumption, heavy and bulky sensors to others with better information-to-weight ratio. Many applications that could benefit from the camera's light weight, includes people localization and environment modeling in rescue operations - cameras can be easily adapted to helmets used by rescuers, or simply worn by soldiers and fire-fighters. Furthermore, it is very attractive for tracking small flying vehicles, telepresence solutions (head motion estimation using an outward-looking camera), augmented reality (AR) environments (camera's small size facilitate its attachment into AR displays) and television (camera motion estimation for live AR). Cameras are thus well adapted for embedded systems and often pre-integrated into mobile computing devices such as PDAs, phones and laptops [9, 10, 11, 13].

On the other hand, vision captures the geometry of its surrounding environment indirectly through photometric effects – reason why it is not easy to turn the sparse sets of features from an image into reliable long-term landmarks. Camera images may be noisy and scene-dependent, and under some circumstances, the computed probabilistic models can be biased, overconfident, or subject to other numerical challenges [9, 14].

1.3 Objectives

The aim of this thesis is to contribute to the development of intelligent wheelchairs by proposing new hardware designs and computer vision methodologies. The main hypotheses addressed in this thesis are:

It is possible to design an intelligent wheelchair to assist severely handicapped individuals using low cost off-the-shelf devices without interfering with the normal operation of the power wheelchair, and with reduced visual impact.

The use and extension of current vision-based methodologies can provide robust localization for intelligent wheelchairs.

In order to verify these statements, the following intermediate objectives were defined:

- Review relevant work in the field of assistive robotics and intelligent wheelchairs.
- Propose and implement a hardware framework to provide sensing and processing capabilities to regular powered wheelchair, concerning with the user limitations and with the normal wheelchair operation.
- Assemble the proposed hardware framework in a regular powered wheelchair, perform experiments and validate the design.
- Review relevant vision-based localization methodologies and related works.
- Propose and implement novel methodologies to improve vision-based localization in real-world conditions.
- Develop a flexible interface to enable visual debugging and feedback about the local features used to estimate localization.
- Validate the proposed approaches with a controlled image set, as well as with real-world environmental conditions (i.e. in the presence of multiple illumination sources, variations in intensity and color).
- Develop a vision-based motion estimation algorithm.
- Validate the proposed motion estimation through experiments with real-world conditions.

1.4 Contributions

This thesis makes the following contributions to the field:

- Conceptualization and development of a modular platform for the development of intelligent wheelchairs. This work led to four publications [15, 16, 17, 18].
- Definition and design of a user-centered hardware framework for intelligent wheelchairs. The contribution of this work concerns the mitigation of the visual and ergonomic impacts caused by the addition of sensing and computational capabilities. Another contribution is compatibility of the hardware framework to multiple models and brands of powered wheelchair, facilitating the conversion of regular powered wheelchair into intelligent wheelchairs. Although the concept of the Intellwheels flexible hardware framework had been published before [19], the current hardware framework was a contribution of this work. This work was submitted to one publication [20].
- Assessment of a robotic simulator to assist the selection of the general purpose simulator that better matches the requirements of the IntellWheels project. This work led to one publication [21].
- Development of a shared control methodology that is effective to avoid collisions, and yet simple enough to run in real-time in embedded systems with limited computational capability. This work led to one publication [22].
- Development of an extension of SURF feature detector with invariance to large photometric variations. Unlike other authors that perform color space mapping to deal with illumination changes, in this approach we normalize the variables used to compute filter responses. Thus, it is possible to compute invariant feature responses over regular RGB images. The contribution of this work consists on the combination of the local normalization (LN) technique with feature detection algorithms to enhance their photometric invariance properties.
- Development of a method to extend SURF feature detector invariance properties. The novelty of the method resides in the combination of the color constant working space provided by the LSAC descriptor with the SURF feature detection. The inclusion of this preprocessing step adds a small computational load to the overall algorithm, but it proved to provide a significant increase in feature detection invariance. This work led to one publication [23] and one submission [24].
- Development of a visual odometry algorithm to estimate robot's trajectory. The algorithm is based on affordable RGB-D cameras. It uses the RGB image to detect and match visual features over consecutive frames. Through depth information, the algorithm estimates the 3D position of each feature matched and computes the motion that better explains the transformation. This work was submitted to publication [25].

Besides these contributions, the work presented in this thesis also allowed the dissemination of the IntellWheels project through the mass media, as well as through other international scientific publications in the areas of Modeling and Simulation, Artificial Intelligence, Robotics and Assistive Technologies. The IntellWheels project was also distinguished by several associations with the attribution of four prizes in the areas of Social Inclusion, Assistance and Robotics.

1.5 Structure of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 presents the thesis background. Different concepts, tools and methodologies that will be frequently used in the course of this work are presented in detail.

Chapter 3 introduces the concept of intelligent wheelchairs and a comprehensive literature review of the main intelligent wheelchair projects. The chapter also presents the IntellWheels project, which was the basis for the integration of the methodologies developed in this thesis. To address some of the IntellWheels requirements, we describe our user-centered hardware design, the requirements and assessment of robotic simulators and an obstacle avoidance methodology.

Chapter 4 presents a literature review regarding the color feature detectors and constancy methodologies. A mathematical formalization demonstrate the effects of illumination changes in the response of popular feature detectors. Next, we propose and evaluate two methodologies to provide local feature detectors with invariance to large photometric variations.

Chapter 5 shows an application of the proposed photometric invariant methodology to the problem of visual odometry. We describe our motion estimation method and give two examples of robot trajectory recovery using, as only input, images provided by a camera.

Chapter 6 summarizes the main conclusions and contributions of this thesis. In addition, reflections about the limitations and some future perspectives are discussed.

Chapter 2

Background Theory

This background chapter aims at introducing the different concepts discussed in the course of this thesis. Some definitions and methodologies that will be frequently used in this work are presented in detail. First, we will introduce the image formation theory used to model the interaction of light with a surface, in terms of the properties of the surface and the nature of the incident light. Next, we review the main mathematical models used to describe the way colors can be represented. Section 2.3 presents the model used throughout this document to represent variations in the intensity and color or the scene illumination. Section 2.4 describes an intermediate image representation employed to reduce the computational cost of convolution operations with rectangular filters. Section 2.5 describes the solution adopted to compute the image partial derivatives. Section 2.6 presents a definition of the Gaussian kernels, how they can be computed and their advantages in the computer vision context. Section 2.8 discusses the computational theory used to model salient image details, and the desired properties that these models should hold. Finally, the summary and conclusions of this chapter are presented in Section 2.9.

2.1 Theory of Image Formation

Some bodies like the sun and electric light filaments are able to produce the electromagnetic radiation in the range detected by the humans eyes (luminous objects). These wavelengths typically ranges from 0.43μm (violet) to about 0.79μm (red), and are called visible light. Once most objects are not able to produce light, they are only visible through the reflection of light rays emitted from other luminous objects (light sources). Reflection is a fundamental physical phenomenon and corresponds to a change in the direction of the light propagation. It can be classified as specular or diffuse according to the nature of the surface material.

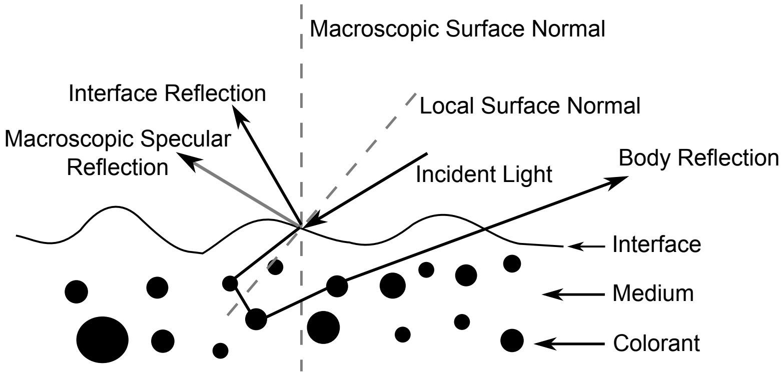


Figure 2.1: Reflection of light in inhomogeneous materials (adapted from [26]).

In the case of specular reflections, most of the incident light is reflected off the surface in a single direction. The direction of the reflected ray can be estimated through the second law of reflection, which states that the angle between the reflected ray and the surface normal equals the angle between the normal and the incident ray. Specular reflections occur more intensely in optically homogeneous materials like mirrors, metallic objects and other shiny and highly polished surfaces, because the incident ray is not able to penetrate through the surface interface.

On the other hand, diffuse reflection occurs mainly in materials that are optically inhomogeneous. When a light ray hits an inhomogeneous surface, it must pass through the interface between the air and the surface medium. Once the index of refraction of the surface medium and the index of refraction of the air are different, part of this light energy is reflected at the interface as a local specular reflection (Figure 2.1). Here we refer to local specular reflection because materials that are optically rough do not present a macroscopic or reference surface normal, but instead they present a number of different microscopic surface normals that significantly varies from point to point. The remaining light energy penetrates the interface, passes through the medium and is dispersed by the colorant. At this point, the light energy can be either: absorbed by the colorant; re-emitted through the interface, producing body reflection; or transmitted through the material, when this material is translucent.

The geometric distribution of the body reflection is sometimes assumed to reflect light evenly in all directions. Such isotropic surfaces are known as Lambertian surfaces, because they preserves the Lambert's cosine law, which states that the reflected light intensity in any direction of a perfectly diffusing surface varies as the cosine of the angle between that direction and the normal vector of the surface. Therefore, the luminance of that surface is the same regardless of the viewing angle. Although very few natural surfaces are truly Lambertian, many common materials can be described as near-Lambertian at moderate angles, including concrete, asphalt, varnishes, paper, ceramics, plastics and most paints.

Assuming that a scene contains surfaces which exhibits Lambertian reflectance properties, its resulting image I can be modeled as a function of the irradiance $F(\lambda, x_i)$ falling onto an infinitesimal small patch on the sensor array, as given by the equation:

$$I(x_i) = \int F(\lambda, x_i) p(\lambda) d\lambda \tag{2.1}$$

where $p(\lambda)$ is the camera spectral sensitivity of wavelength λ, and x_i is the object location x_{obj} expressed in the image coordinate frame. For a Lambertian surface, which reflects light equally in all directions, the irradiance can be described in terms of the light spectral power distribution $E(\lambda, x_{obj})$ and the surface reflectance $S(\lambda, x_{obj})$.

$$F(\lambda, x_i) = E(\lambda, x_{obj}) S(\lambda, x_{obj}) \tag{2.2}$$

Thus, the intensity I measured by the sensor in the location x_i is given by

$$I(x_i) = \int E(\lambda, x_{obj}) S(\lambda, x_{obj}) p(\lambda) d\lambda \tag{2.3}$$

Although each sensor responds to a range of wavelengths, the sensor is often assumed to respond to the light of a single wavelength. Thus, one can approximate the sensor response characteristics by Dirac's delta functions, as given by:

$$p_k(\lambda) = \delta(\lambda - \lambda_k) \tag{2.4}$$

where $k \in \{R, G, B\}$. Through the former assumption, it is possible to simplify the Equation (2.3) and express the intensity $I_k(x_i)$ measured by the sensor k in the position x_i as:

$$I_k(x_i) = E(\lambda_k, x_{obj}) S(\lambda_k, x_{obj}) \tag{2.5}$$

In [26], Shafer argues that the model described by the Equation (2.3) presents severe limitations because it assumes that the illumination at any point comes from a single light source. Shaffer states that a more realistic model may consider the illumination as a combination of the light source and the ambient light. This ambient light can be defined as an isotropic light of lower intensity than the light source (and possibly with a different spectral power distribution), that can come from the scattering of the white light source, objects highlights, infra-red sensitivity of the camera sensor and inter-reflections of walls and other objects. Thus, taking advantage of the linear properties of the spectral projection, Shaffer extends the model by adding a diffuse term A(λ):

$$I_k(x_i) = \int E(\lambda, x_{obj}) S(\lambda, x_{obj}) p_k(\lambda) d\lambda + \int A(\lambda, x_{obj}) p_k(\lambda) d\lambda \tag{2.6}$$

However, the assumption that the ambient light is equal in all directions implies that the diffuse term is independent of the surface and of the spatial coordinate x_{obj}. When computing the partial derivative of I, the effect of A(λ) is canceled. Therefore, the reflection model of the spatial derivative of the image I at x_{obj} on the scale σ can be described as:

$$I_k(x_i) = \int E(\lambda, x_{obj}) S_\sigma(\lambda, x_{obj}) p_k(\lambda) d\lambda \tag{2.7}$$

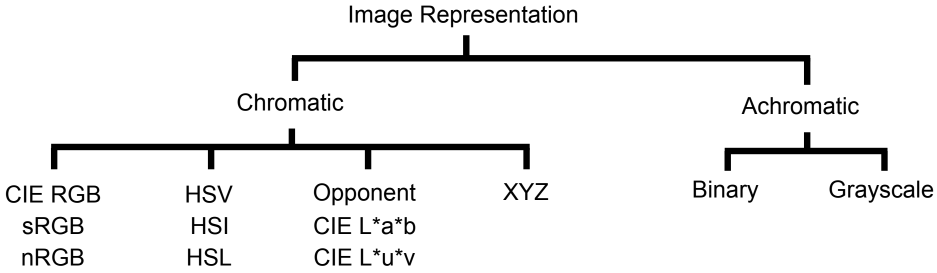


Figure 2.2: Taxonomy of color spaces.

2.2 Color Spaces

Color is a subjective human sensation produced when the electro-magnetic radiation in the range of 430 nm (violet) to about 790 nm (red) reaches the human eye. Extensive experimental studies provide evidence that human eyes have around 6 to 7 million photoreceptors (called cone cells) responsible for color perception, which can be classified in three groups according to the wavelength range they are capable to sense.

The L cones have sensitivity in the range of long optical wavelengths (500–700 nm), and are responsible for the sensation that humans call red. The M cones have sensitivity in the range of middle optical wavelengths (450–630 nm), and are responsible for the sensation we call green. Finally, the S cones have sensitivity in the range of short optical wavelengths (400–500 nm), and are responsible for the sensation of blue.

The ability to perceive colors through the interaction of three types of color-sensing cone cells is the physiological basis for the trichromatic theory of color vision, which suggests that any color sensation in the visible band of the electromagnetic spectrum can be created by mixing three primary spectra (tristimulus values).

For this reason, describing colors accurately is utmost important when working with colors in computer vision methodologies. It requires a standard system to convert the physical stimulus (spectral radiance) into a mathematical representation. Any methodology used to associate the tristimulus values with each color is called a color space. Several different color spaces were proposed in the literature in order to deal with different purposes, some aiming to mimic the human vision system, others to comply with the way electronic monitors reproduce colors and others to describe how humans understand the colors (Figure 2.2).

CIE RGB

The CIE RGB color space was created to model the way humans perceive colors [27]. In this model, each color appears in its primary spectral components of red (R), green (G) and blue (B). Each of these three components corresponds to a filtered spectral mapping from image space to a 3-D sensor space [28]. The model that describes this transformation was discussed in Section 2.1,

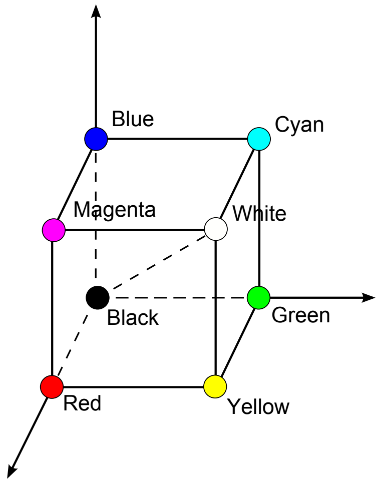


Figure 2.3: RGB Color space: points along the main diagonal have gray values.

and can be summarized through the Equation (2.3). All colors that can be created can therefore be represented within a cubic volume, positioned in the positive octant of a three dimensional Cartesian system whose axes are the RGB primaries (Figure 2.3). The different colors in this model are points inside the cube, and are defined by vectors extending from the origin.

Grayscale

Grayscale images are monochromatic images that carry only intensity information. Distinctly of the binary images, they can represent many shadows of gray in-between the black and white. Grayscale images often result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.). For those cases where grayscale images are synthesized from full color image, the luminance is calculated as a weighted sum of the three linear-intensity values given by:

$$I_G(x,y) = 0.299R + 0.587G + 0.114B \tag{2.8}$$

Binary

Images that are void of color are called achromatic/monochromatic, because their only attribute is intensity. Binary images are digital images that can have only two values $0,1$ to describe the intensity of each pixel. For this reason, they are stored in a single bit and are commonly called black-and-white images. Such images often occur after a thresholding operation of grayscale images.

$$I_B(x,y) = \begin{cases} 1 & \text{if } I_G(x,y) > T \\ 0 & \text{else} \end{cases} \tag{2.9}$$

sRGB

sRGB is a RGB color space proposed by a group of private companies to approximates the color gamut of the most common computer display devices. This specification allowed sRGB to be directly displayed on typical CRT monitors of the time, which greatly aided its acceptance. The sRGB color system is defined from the XYZ color system, so that the sRGB tristimulus values for an illuminated object of a scene are simply linear combinations of the CIE XYZ values: (0.64, 0.33, 0.03) for the R light, (0.30, 0.60, 0.10) for the G light, (0.15, 0.06, 0.79) for the B light. The (x,y,z) coordinates of its reference white are those of the CIE standard illuminant D65, namely (0.3127, 0.3290, 0.3583). The transformation from the XYZ space into sRGB can be described by:

$$\begin{bmatrix} sR \\ sG \\ sB \end{bmatrix} = \begin{bmatrix} 3.2419 & -1.5374 & -0.4986 \\ -0.9692 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.10}$$

Reversely, the conversion from the sRGB space into XYZ may be expressed through:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.35758 & 0.180423 \\ 0.212671 & 0.71516 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} sR \\ sG \\ sB \end{bmatrix} \tag{2.11}$$

XYZ

One of the main limitations of the RGB color space is that it can not represent all colors the average human can see. In order to encompass these colors the RGB components would have to, for instance, assume negative values. To address this problem, the Commission Internationale de l'Eclairage (CIE) proposed in 1931 the CIE XYZ color space. This model is based on three artificial primaries, XYZ, that do not correspond to any real light wavelength, but that can represent all visible colors by using only positive values. The Y primary is intentionally defined to match closely to luminance, while X and Z primaries give color information. CIE XYZ is a commonly used standard, and serves as the basis from which many other color spaces are defined. Its main advantage when compared to other color systems consists in the fact that it is completely device-independent. The transformation from CIE RGB to CIE XYZ can be mathematically expressed through:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.769 & 1.752 & 1.130 \\ 1.000 & 4.591 & 0.060 \\ 0.000 & 0.057 & 5.594 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.12}$$

nRGB

The normalized RGB color space (nRGB) is an attempt to separate the light intensity to process color information independently from varying lighting levels. The red, green, and blue components of nRGB can be obtained by dividing each component of the RGB space by the sum of the three components. Note that as r+g+b=1 only two components are enough to define a color. Such space is independent of uniformly varying lighting levels and provides a relative robustness in certain types of illumination changes. The transformation from the RGB space into nRGB can be described by:

$$
\begin{aligned}
r &= \frac{R}{R+G+B} \\
g &= \frac{G}{R+G+B} \\
b &= \frac{B}{R+G+B}
\end{aligned}
\tag{2.13}
$$

HSV

In addition to the high correlation of the three primaries, RGB, XYZ and other linear color spaces are not well suited to describe colors in terms that are practical for humans to interpret. In fact, when required to describe colors, humans make use of other attributes like hue, saturation and intensity. Historically, in the beginning of the 19th century the professor and artist Albert Henry Munsell [29] proposed the use of three subjective parameters (hue, value and chroma) to describe colors according to what he felt that most closely reflect the perceptions of human observers. Later, based on Munsell's work, related color models were proposed to decouple the intensity component from the color carrying information, like the HSI (hue, saturation, intensity), the HSV (hue, saturation, value) introduced by Smith in [30], and the HSL (hue, saturation, lightness) presented by Joblove and Greenberg in [31]. Although these color spaces have a different mathematical definition for each attribute, they share a very close conceptual definition. In Munsell's color space and in its variants, the hue component defines the color itself. It is the attribute associated with the dominant wavelength in a mixture of light waves, and represents the dominant color perceived by the observer. The values for the hue axis vary from 0 to 360 beginning and ending with red and running through green, blue and all intermediary colors. Saturation refers to the relative purity of a light wave. It is the amount of white light mixed with a hue, and indicates its colorfulness (difference between the hue against gray) in the color space. The pure spectrum of colors is fully saturated. The degree of saturation is inversely proportional to the amount of white light added. The values for the saturation range from 0 to 1, indicating respectively no color saturation and maximum saturation of a given hue at a given illumination. The value component - lightness (HLS) or intensity (HSI), shows the amplitude of light, indicating the illumination level. It is a subjective, non-quantitative reference to physiological sensations and perceptions of light, which relates to the achromatic notion of intensity. Both vary from 0 (black, no light) to 1 (white, full illumination).

HSV and its related models can be derived from the RGB space through geometric strategies. This becomes clear when we tilt the RGB color cube on the Cartesian coordinate system, positioning the black vertex at the origin (0,0,0) and the white vertex above it along the vertical axis (0,0,1)(Figure 2.4A). Representing the achromatic diagonal of the RGB cube coincident with the vertical axis of the Cartesian space it is possible to determine the illumination level of any point through the plane that is perpendicular to the vertical axis and that contains the point. Furthermore, it is possible to describe the colors decoupling the illumination component by projecting the RGB color cube into the xy plane, which presents intensity value zero and is therefore referred to as chromatic plane. The position on the plane gives information about the chromaticity of a pixel. In the chromaticity plane, primary colors are separated by 120°, while secondary colors are 60° apart from the primaries. Through the chromaticity plane, both hue and saturation can be defined with respect to the hexagonal shape of the projection. Mathematically, the transformation from the RGB to the HSV space is given by:

$$
\begin{aligned}
&M = max(R,G,B)\\
&m = min(R,G,B)\\
&V = M\\
&S = \begin{cases} (M-m)/_M & \text{if } M \neq 0\\ 0 & \text{otherwise}\end{cases}\\
&H = \begin{cases} Undefined & \text{if } s = 0\\ 60\,(G-B)/_S & \text{if } M = R\\ 120+60\,(B-R)/_S & \text{if } M = G\\ 240+60\,(R-G)/_S & \text{if } M = B\end{cases}
\end{aligned}
\tag{2.14}
$$

The hue of a point is considered as the angle from the reference line, which usually is designated by the red axis as indicated in the Figure 2.4B. The hue increases counterclockwise from the reference and ranges from 0° to 360° (i.e. the hue of the blue color is 240°, of the yellow is 60° and of the green is 120°). The saturation component is the radial distance to the achromatic axis (the length of the vector to a given point in the color space). Finally, the intensity is the height in the vertical axis direction, and describes the gray levels, from zero (no illumination – black) to one (maximum illumination – white). Colors in the HSV model are defined with respect to normalized the red, green, and blue values, so that each component may be divided by 255 to convert them to the range $[0,1]$. The conversion may also include a reverse gamma correction to first yield these intensities.

The HSI color system has a good capability of representing the colors of human perception, because human vision system can distinguish different hues easily, whereas the perception of different intensity or saturation does not imply the recognition of different colors [27]. The transformation to the HSL cylindrical space is defined through the RGB space

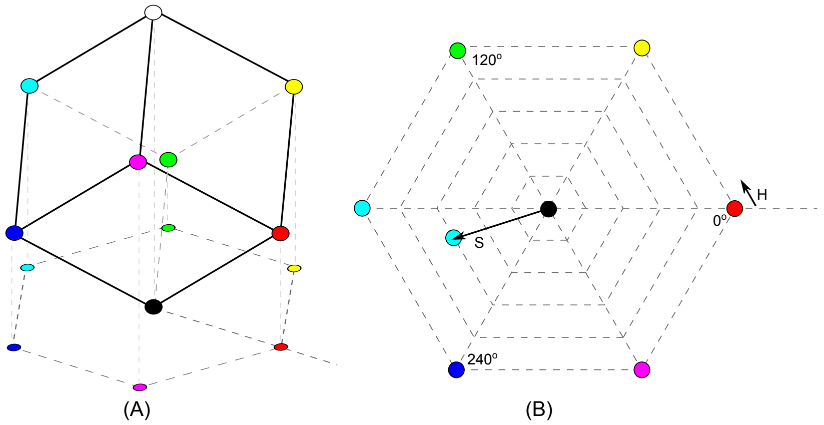


Figure 2.4: Conceptual relation between the RGB and the HSV model: the tilted RGB cube (A) and the projection of the colors into the xy plane (B).

$$
\begin{aligned}
M &= max(R,G,B)\\
m &= min(R,G,B)\\
I &= \frac{R+G+B}{3}\\
S &= 1-\frac{m}{I}\\
H &= \begin{cases} 60\,(G-B)\,/_{S} & \text{if } M = R\\ 120+60\,(B-R)\,/_{S} & \text{if } M = G\\ 240+60\,(R-G)\,/_{S} & \text{if } M = B \end{cases}
\end{aligned}
\tag{2.15}
$$

When considering all the colors that can be represented, the HSV color space may be geometrically represented as a cone, but once the RGB space has a more limited subset of colors the geometry that better bounds this color space is the hexagonal cone (Figure 2.5). Each slice of the cylinder perpendicular to the intensity axis is a plane with the same intensity.

Similar to the HSV, when considering all the possible colors that can be represented, the HSL color space may be described as a cylinder, where the coordinates r, σ, z correspond, respectively, to the values of saturation, hue and intensity. However, when considering just the colors that can be represented in the RGB cube, this color space is better represented by a double hexcone (Figure 2.5). Just as the previous color spaces, the HSI space may be geometrically represented by a cylinder, but as the HSL, it is better represented by a double hexcone when only the colors represented at the RGB color space are considered.

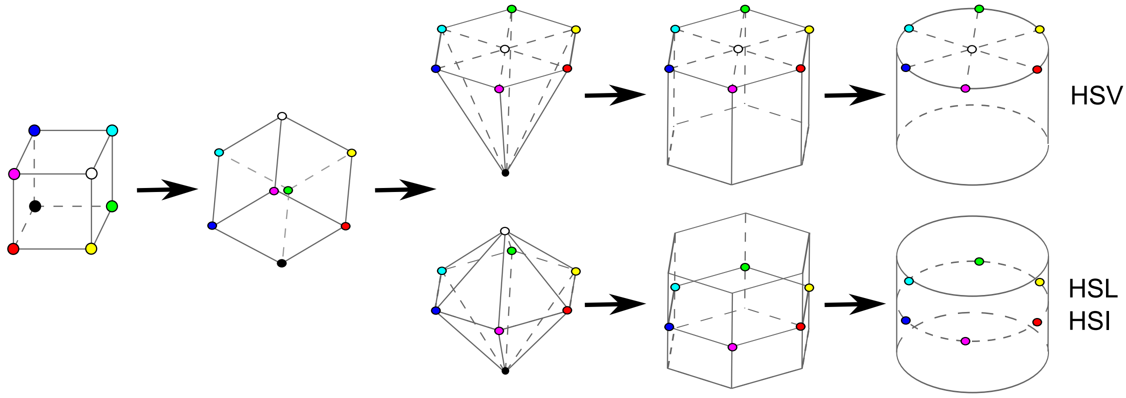


Figure 2.5: Geometric representation of HSV, HSL and HSI color spaces.

$$
\begin{aligned}
M &= max(R,G,B)\\
m &= min(R,G,B)\\
L &= \frac{M+m}{2}\\
S &= \begin{cases} \dfrac{M-m}{M+m} & \text{if } L < 0.5\\ \dfrac{M-m}{2-M+m} & \text{if } L \geq 0.5 \end{cases}\\
H &= \begin{cases} 60\,(G-B)\,/_S & \text{if } M = R\\ 120+60\,(B-R)\,/_S & \text{if } M = G\\ 240+60\,(R-G)\,/_S & \text{if } M = B \end{cases}
\end{aligned}
\tag{2.16}
$$

Opponent

The opponent color space theory states that the three signals produced by the cone cells present in the retina are converted into three channels (O_1, O_2, O_3) before been transmitted to the brain. O_1 is the chromatic red/green component, and is derived by differencing data from the red and green cones. O_2 is the chromatic yellow/blue component, and is derived by differencing the values from the luminance channel (yellow = red + green) and the blue cones. Finally, the intensity information is represented by the O_3 component, which is an achromatic channel created by summing the excitation from red and green cones [32], Equation (2.17).

$$
\begin{aligned}
O_1 &= \frac{R-G}{\sqrt{2}} \\
O_2 &= \frac{R+G-2B}{\sqrt{6}} \\
O_3 &= \frac{R+G+B}{\sqrt{3}}
\end{aligned}
\tag{2.17}
$$

The chromaticity coordinates O_1, O_2 can be represented in polar coordinates by the saturation, and corresponds to the Euclidean distance from the lightness axis. The hue angle h, expressed in degrees starting from the positive O_1 axis (red) and turning in an anti-clockwise direction, can be described by the Equation (2.18).

$$
\begin{aligned}
H &= \arctan \frac{O_1}{O_2} \\
S &= \sqrt{O_1^2 + O_2^2} \\
I &= O_3
\end{aligned}
\tag{2.18}
$$

CIE L*a*b* and CIE L*u*v*

The ability to express color difference in a uniform scale is a significant characteristic that is not achieved even by popular spaces like RGB, sRGB, Nrgb, XYZ, HSV, HSL, HSI. Thus, it is not possible to evaluate the similarity of two colors from their Euclidean distance. In order to match the sensitivity of human eyes with computer processing CIE introduced the CIE L*a*b* [33] (also called CIELAB) and CIE L*u*v* (also called CIELUV) color spaces [27]. These color models are both uniform derivations from the standard CIE XYZ space, and can be represented through the uniform chromaticity scale (UCS). The UCS is a diagram that uses a non-linear transform and weighting of the XYZ values to derive a two-dimensional model that approximates the perceptual uniformity property.

CIELAB and CIELUV are opponent color spaces described in three dimensions, which represent lightness of the color L*, the difference between red and green (a* for CIELAB or u* for CIELUV), and the difference between yellow and blue (b* for CIELAB or v* for CIELUV). The Y component from the XYZ color space is a linear scale of lightness with equal steps between each value. Since humans have more ability to differentiate variations when they occur in higher intensities than in lower intensities, this kind of scale is not adequate to represent differences in lightness that are visually equivalent. For example, a difference between values of 10 and 15 on the Y lightness scale differ by the same magnitude as values of 70 and 75. Thus, Y values can be translated to other values that are approximately uniformly spaced, but more indicative of the actual visual differences.

The resulting scale L* represents lightness and closely models the Munsell system's scale of Value. The central vertical axis L* ranges from 0 (black) to 100 (white) and is used for both

CIELAB and CIELUV uniform color spaces. On the chromaticity plane, the a* axis indicate the transition between red (positive values) and green (negative values), while the b* axis indicates the transition between yellow (positive values) and blue (negative values). It is important to note that neither a* nor b* corresponds to known psychophysical properties of visual perception. The non-linear transformation from the XYZ to the CIELAB and CIELUV uniform color spaces in terms of the CIE XYZ tristimulus values are given by the Equations (2.19) and (2.20), respectively.

$$
\begin{aligned}
L &= \begin{cases} 116\frac{Y}{Y_n}^{1/3} & \text{if } Y/_{Y_n} > 0.008856 \\ 903.3\frac{Y}{Y_n} & \text{otherwise} \end{cases} \\
f(x) &= \begin{cases} x^{1/3} & \text{if } x > 0.008856 \\ 7.787x + \frac{4}{29} & \text{otherwise} \end{cases} \\
a* &= 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \\
b* &= 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]
\end{aligned}
\tag{2.19}
$$

$$
\begin{aligned}
L &= \begin{cases} 116\frac{Y}{Y_n}^{1/3} & \text{if } Y/_{Y_n} > 0.008856 \\ 903.3\frac{Y}{Y_n} & \text{otherwise} \end{cases} \\
u' &= \frac{4X}{X + 15Y + 3Z} \\
u_n &= \frac{4X_n}{X_n + 15Y_n + 3Z_n} \\
u* &= 13L(u' - u_n) \\
v' &= \frac{9Y}{X + 15Y + 3Z} \\
v_n &= \frac{9Y_n}{X_n + 15Y_n + 3Z_n} \\
v* &= 13L(v' - v_n)
\end{aligned}
\tag{2.20}
$$

where (X_n, Y_n, Z_n) are the tristimulus values corresponding to "the illuminant" (reference white), which according to the CIE Standard illuminant corresponds to (0.950155, 1.0000, 1.088259). An approximate measure of the magnitude of the difference between colors (relative perceptual distance) can be derived from the Euclidean distance (D_{ab}) between the color components specified in CIELAB/CIELUV coordinates [32].

$$
D_{ab} = \sqrt{\Delta L*^2 + \Delta a*^2 + \Delta b*^2} \tag{2.21}
$$

From CIELAB and CIELUV spaces, it is possible to derive the perceptual color attributes such as intensity, hue and saturation conveniently. One may use one of the two CIE color spaces and the associated color difference formulas to map the (L* a* b*) or (L* u* v*) coordinates to the

HSI cylindrical coordinates [28]. The chromaticity coordinates - (a*,b*) for CIELAB model and (u*,v*) for CIELUV model - are represented in polar coordinates by the saturation, and corresponds to the Euclidean distance from the lightness axis [34].

$$S = \sqrt{\Delta u*^2 + \Delta v*^2} \tag{2.22}$$

The hue angle H_{uv}, expressed in degrees starting from the positive a* axis (red) and turning in an anti-clockwise direction, can be described by:

$$H_{uv} = atan2(v*, u*) \tag{2.23}$$

2.3 Illumination Changes

One of the most difficult problems of working with colors is that the object's apparent color varies unpredictably with variations in the intensity and temperature of the light source. For instance, a well-known example occurs in outdoor environments with daylight variations, the color shift between sunny and cloudy days is simply not well modeled as Gaussian noise in RGB [35].

Fortunately, there exist models that are able to describe those kind of variations. One of these models is given by a diagonal transform of the color space, which corresponds to the so-called von-Kries model [36], or Diagonal model (DM). According to Diagonal Model, it is possible to map an observed image I^o taken under an unknown illuminant to a corresponding image I^c under a canonical illuminant through a proper transformation in order to render images color constant. The Diagonal model can be mathematically described by the following relation:

$$I^c = D^{u,c} I^u \tag{2.24}$$

where $D^{u,c}$ is a diagonal matrix that maps the corresponding images from the unknown light source u into the canonical light source c. To exploit this observation, Forsyth [37] modeled the illumination change using a set of three scale factors $[a, b, c]$ such that an observed RGB^o image $[R^o, G^o, B^o]$ is mapped into its corresponding RGB^c under a reference light $[R^c, G^c, B^c]$ according to:

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} R^o \\ G^o \\ B^o \end{bmatrix} \tag{2.25}$$

Like the Reflection Model, Equation (2.3), the DM is strictly valid under the assumption of narrow-band camera sensors. Despite the fact that camera sensors are not narrow-band, Finlayson *et al.* [38] note that this model can still be used for many surfaces and light sources. However, they point that the DM really presents its shortcomings when trying to map saturated and near saturated colors. For this reason, they propose an extention of the DM that includes the "diffuse" light term (λ) by adding an offset to the Equation (2.25). Such model is known as the Diagonal-offset Model (DOM), and is given by:

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} R^o \\ G^o \\ B^o \end{bmatrix} + \begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} \tag{2.26}$$

Since the offset term $(o_1, o_2, o_3)^T$ is expected to be relatively smaller than the diagonal term $[a, b, c]$, the illumination change can still be modeled using the DM, but with some alternative to avoid null solutions. According to Sande *et al.* [39], the Diagonal-offset Model corresponds to (2.6). Thus, assuming the wavelength λ_k at the position X and a surface reflectance $S(\lambda_k, X)$ for $k \in \{R, G, B\}$, the Diagonal-offset Model can be described in terms of the light source and the surface reflectance as:

$$\begin{bmatrix} E^c(\lambda_R)S(X,\lambda_R) \\ E^c(\lambda_G)S(X,\lambda_G) \\ E^c(\lambda_B)S(X,\lambda_B) \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} E^0(\lambda_R)S(X,\lambda_R) \\ E^0(\lambda_G)S(X,\lambda_G) \\ E^0(\lambda_B)S(X,\lambda_B) \end{bmatrix} + \begin{bmatrix} A(\lambda_R) \\ A(\lambda_G) \\ A(\lambda_B) \end{bmatrix} \tag{2.27}$$

Finally, as the surface reflectance $S(\lambda_k, X)$ remains equal under the canonical and the observed images and A(λ_R) does not depend on the surface reflectance because it represents the ambient light, the Equation (2.27) can be simplified as follows:

$$\begin{bmatrix} E^c(\lambda_R) \\ E^c(\lambda_G) \\ E^c(\lambda_B) \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} E^0(\lambda_R) \\ E^0(\lambda_G) \\ E^0(\lambda_B) \end{bmatrix} + \begin{bmatrix} A(\lambda_R) \\ A(\lambda_G) \\ A(\lambda_B) \end{bmatrix} \tag{2.28}$$

On the basis of the Diagonal Model and the Diagonal-offset Model, illumination variations can be classified into five categories: light intensity change (LIC), light intensity shift (LIS), light intensity change and shift (LICS), light color change (LCC) and finally light color change and shift (LCCS) [39].

In the first category (LIC), the three RGB components of a given image vary equally by a constant factor, such that a = b = c and $o_1 = o_2 = o_3 = 0$. Hence, when a function is invariant to light intensity changes, it is scale-invariant with respect to light intensity. Therefore, we can rewrite (2.26) as:

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} R^o \\ G^o \\ B^o \end{bmatrix} \tag{2.29}$$

In the second category (LIS), a constant shift affects equally all the RGB channels of a given image, such that a = b = c = 1 and $o_1 = o_2 = o_3 \neq 0$. Therefore, when a function is invariant to light intensity shift, it is shift-invariant with respect to light intensity, as defined by:

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} R^o \\ G^o \\ B^o \end{bmatrix} + \begin{bmatrix} o_1 \\ o_1 \\ o_1 \end{bmatrix} \tag{2.30}$$

The third category (LICS), is a combination of the two above mentioned categories, and also affects all three RGB channels equally, in such a way that a = b = c and $o_1 = o_2 = o_3 \neq 0$. Thus, when a function is invariant to light intensity changes and to light intensity shift, it is known as scale-invariant and shift-invariant with respect to light intensity. According to this, we can write:

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} R^o \\ G^o \\ B^o \end{bmatrix} + \begin{bmatrix} o_1 \\ o_1 \\ o_1 \end{bmatrix} \tag{2.31}$$

The two remaining categories do not assume that RGB channels are equally affected by variations in the light source. The fourth category (LCC) is the Diagonal Model, which assumes that $a \neq b \neq c$ and $o_1 = o_2 = o_3 = 0$. Since images are able to vary differently in each channel, this category can model changes in the illuminant color temperature and light scattering. Following these assumptions, the light color change category can be described by (2.25).

The last category (LCCS) is the full Diagonal-offset Model, and takes into consideration independent scales $a \neq b \neq c$ and offsets $o_1 \neq o_2 \neq o_3$ for each image channel. The light color change and shift is the most complete model, and can be described through the full Diagonal-offset Model (2.26).

2.4 Integral Images

Integral image is an intermediate image representation proposed by Viola and Jones [40]. This representation was designed to reduce the computational effort regarding the convolution of rectangular filters by pre-processing the input images. An integral image $I_{\Sigma}(x,y)$ can be defined as a table that, at each location $X = (x,y)^T$, contains the sum of all pixels of $I(x,y)$ within the rectangular region formed by the origin O and X (all pixels above and to the left of X). Table 2.1 shows the values of a given gray-scale image (A) and the respective integral image representation (B).

$$I_{\Sigma}(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{2.32}$$

Table 2.1: Example of Integral Image Computation: intensity matrix of a given gray scale image (A) and its respective integral image (B).

57	57	66	72	76
64	65	66	70	78
65	61	64	69	69
109	76	84	109	107
115	69	101	132	124

(A)

57	114	180	252	328
121	243	375	517	671
186	369	565	776	999
295	554	834	1154	1484
410	738	1119	1571	2025

(B)

Note that the implementation of the Equation (2.32) in its current form is not computationally efficient, since at least two passes over each pixel X are needed to compute $I_{\sum}(x,y)$. The integral image can be computed in one-pass over the image using the following recurrence relation:

$$s(y,x) = s(y,x-1) + I(y,x) \tag{2.33}$$

$$I_{\sum}(y,x) = I_{\sum}(y-1,x) + s(y,x) \tag{2.34}$$

where $s(x,y)$ is the cumulative row sum. The cost for deriving this intermediate representation in computer vision applications is totally justified by the significant reduction in the computational complexity of box filters convolution. Once the integral image has been computed, the sum of intensities of any rectangular region is reduced to three additions and four memory accesses. An example is the convolution of a box type filter (dark gray rectangle) with a given image (Figure 2.6A). The first step consists in deriving the sum of all image pixels within the rectangle formed by the origin and filter's bottom-right corner (Figure 2.6B). Next, one may subtract the regions that do not belong to the filter, like the gray areas represented in (Figure 2.6C) and (Figure 2.6D). Finally, it is needed to balance the equation with the region (Figure 2.6E) that was subtracted twice. Using the data provided by the input image, the filter response F_r could be calculated as:

$$F_r = I_{\sum}(x_0-1,y_0-1) + I_{\sum}(x_e,y_e) - I_{\sum}(x_e,y_0-1) - I_{\sum}(x_0-1,y_e) \tag{2.35}$$

This approach is particularly interesting in algorithms that require the convolution of several filters and with filters of big size. Since the number of memory access and arithmetic operations do not vary with the filter size, the time to compute the filter response remains constant.

2.5 Image Derivatives

The derivative of a continuous function $f(x)$ at a given point X is defined by the function whose value at X is the limit

$$\frac{\partial f(x)}{\partial x} = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h} \tag{2.36}$$

However, since the input image is represented as a set of discrete pixels of intensity $I(x,y)$, and not by a continuous function, the problem we have to address is how to compute the image derivatives. One solution to replace partial derivatives is to make use of finite differences of the intensity of consecutive pixels. The partial differential equations are converted into difference equations, thus the resultant system of algebraic equations can be solved using any direct or iterative method.

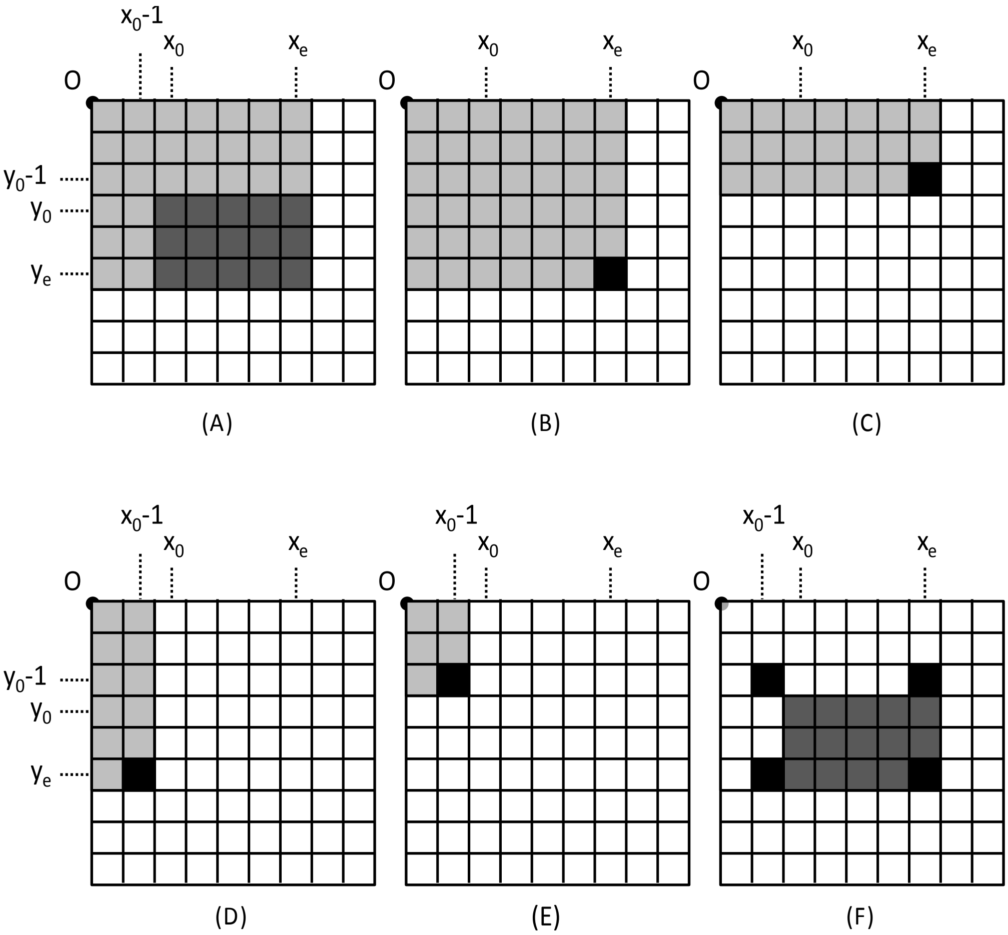


Figure 2.6: Example of box type filter convolution.

If instead of approaching zero, h has a fixed, positive, non-zero value, it is possible to re-write the Equation (2.36) as a forward difference.

$$\frac{\partial I(x)}{\partial x} \approx \frac{I(x+h)-I(x)}{h} \tag{2.37}$$

On the other hand, if h has a fixed, negative, non-zero value, it is possible to approximate the Equation (2.36) as a backward difference, Equation (2.38).

$$\frac{\partial I(x)}{\partial x} \approx \frac{I(x)-I(x+h)}{h} \tag{2.38}$$

Since the definitions of both forward and backward differences are not symmetric, they would compute the derivative at the "half-pixel" position $\partial I(x+h/2)$ and $\partial I(x-h/2)$, respectively. One way to cope with this issue is to use the method of the central difference, Equation (2.39), which fits a parabola through three consecutive points of the profile in order to compute the derivative of the parabola at the center point.

$$\frac{\partial I(x)}{\partial x} \approx \frac{I(x+h)-I(x-h)}{2h} \tag{2.39}$$

With the same mechanism, it is possible to use the finite differences to approximate the partial derivatives of discrete functions of two variables

$$\frac{\partial I(x,y)}{\partial x} \approx \frac{I(x+h,y)-I(x-h,y)}{2h} \tag{2.40}$$

$$\frac{\partial I(x,y)}{\partial y} \approx \frac{I(x,y+k)-I(x,y-k)}{2k} \tag{2.41}$$

By definition, second derivatives may be zero in flat areas, non zero at the onset and end of a gray level step or ramp and zero along ramps of constant slope. Thus, we can approximate partial second derivatives through the Equations (2.42) to (2.44).

$$\frac{\partial^2 I(x,y)}{\partial x^2} \approx \frac{I(x+h,y)-2I(x,y)+I(x-h,y)}{2h} \tag{2.42}$$

$$\frac{\partial^2 I(x,y)}{\partial y^2} \approx \frac{I(x,y+k)-2I(x,y)+I(x,y-k)}{2k} \tag{2.43}$$

$$\frac{\partial^2 I(x,y)}{\partial xy} \approx \frac{I(x+h,y+k)-I(x+h,y-k)-I(x-h,y+k)+I(x-,y-k)}{4hk} \tag{2.44}$$

From the equations above one may note that the finite derivatives approximated through the central difference method are linear filters, thus, can be represented through convolution masks, respectively Equations (2.45) to (2.49). The process of applying the filter will be referred to as convolution, and the pattern of weights used in the filter will be referred to as the kernel of the filter.

$$Mask_{\frac{\partial I(x,y)}{\partial x}} = \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \tag{2.45}$$

$$Mask_{\frac{\partial I(x,y)}{\partial y}} = \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tag{2.46}$$

$$Mask_{\frac{\partial^2 I(x,y)}{\partial x^2}} = \frac{1}{2}\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \tag{2.47}$$

$$Mask_{\frac{\partial^2 I(x,y)}{\partial y^2}} = \frac{1}{2}\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \tag{2.48}$$

$$Mask_{\frac{\partial^2 I(x,y)}{\partial xy}} = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.49}$$

2.6 Gaussian Filtering

Gaussian kernels consist of a weighted sum of pixels using different patterns. Gaussians take advantage of a basic image property: the value of a pixel is usually similar to that of its neighborhood. Thus, assuming that the noise that affects the image preserves this property, it is possible to reduce the effects of the noise by replacing the intensity of each pixel with a weighted average of its neighbors, attenuating high-frequency components (a process often referred to as smoothing or blurring). Using a set of weights that are large at the center and fell off sharply as the distance to the center increases, it is possible to model the kind of smoothing that occurs in a defocused lens system. A good formal model for this fuzzy blob is the symmetric Gaussian kernel described by the Equation (2.50), and depicted in the Figure 2.7-A.

$$G_\sigma(x,y) = \frac{1}{2\Pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2.50}$$

σ is referred to as the standard deviation of the Gaussian, and (x,y) are inter-pixel spaces referred to as pixels. The constant term makes the integral over the whole plane equal to one and is often ignored in smoothing applications.

In this approach, smoothing suppress the noise by enforcing the requirement that pixels look like its neighbors. By down weighting distant neighbors in average, we can assure that the requirement that a pixel looks like its neighbors is less imposed for distant neighbors.

From (2.50), three important practical considerations can be highlighted:

- If σ is small, the smoothing will have little effect because the weights for all pixels of the center will be very small.

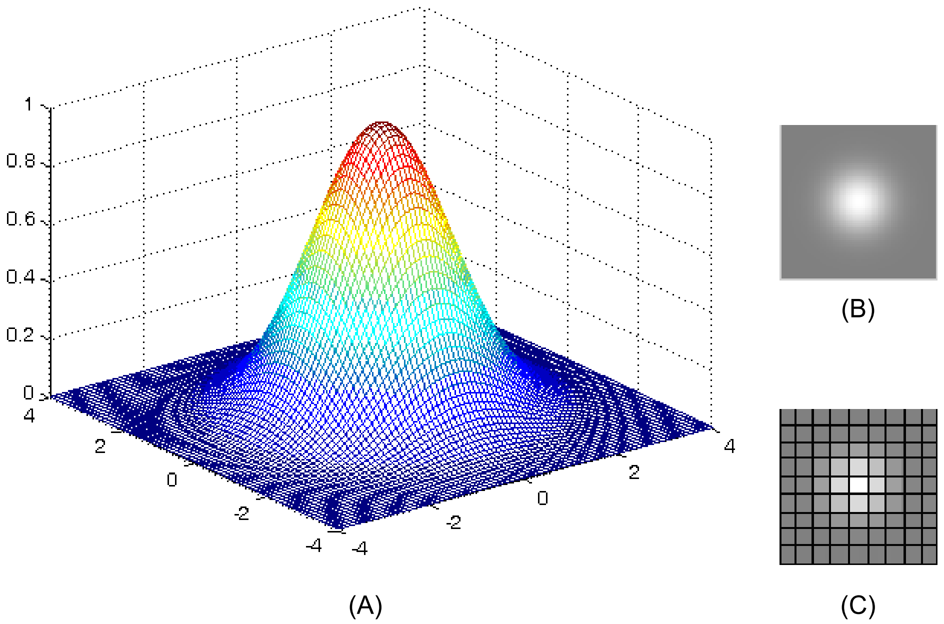


Figure 2.7: Gaussian Filter. (A) 3D plot of the continuous Gaussian filter. (B) 2D continuous Gaussian kernel. (C) 9x9 2D discrete Gaussian kernel.

- For larger standard deviation, the neighboring pixels will have larger weights in the weighting average, which means that the average will be strongly biased toward a consensus of the neighbors (noise large disappears, but at the cost of some blurring).

- A kernel that has large standard deviation will cause much of the image detail disappears along with the noise.

Gaussians are convenient blurring kernels due to a number of important properties they present. The first is that the convolution of a Gaussian $(G_{\sigma 1})$ with another Gaussian $(G_{\sigma 2})$ results in a third Gaussian, Equation (2.51).

$$G_{\sigma 1} \otimes G_{\sigma 2} = G_{\sqrt{\sigma_1^2+\sigma_2^2}} \tag{2.51}$$

Therefore, it is possible to obtain heavily smoothed images by re-smoothing smoothed images. This is an important property because discrete convolution can be an expensive operation (particularly for kernels of large size), and it is common to want versions of an image smoothed by different amounts.

Another important property is the efficiency. For a Gaussian kernel of standard deviation 1 pixel, points outside a $5x5$ grid centered at the origin have values smaller than $e^{-4} = 0.0184$. This means that, for several applications, we can ignore their contribution and represent the discrete Gaussian as a small array. However, if the standard deviation is 10 pixels, we may need at least a $50x50$ array. Accounting the number of operations necessary to perform the convolution of a

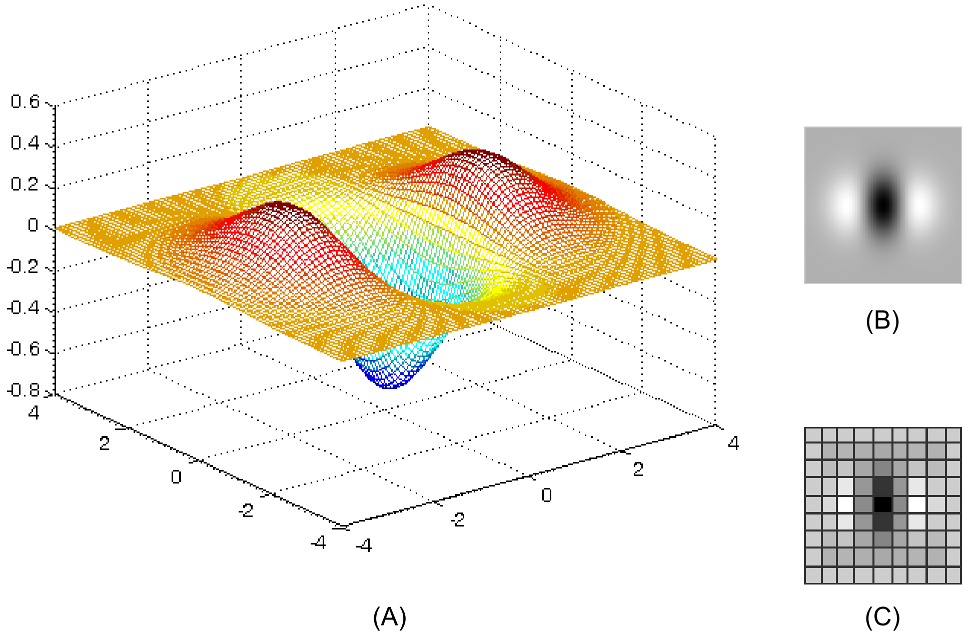


Figure 2.8: Gaussian derivative in x-direction. (A) 3D plot of the continuous filter. (B) 2D continuous kernel. (C) 9x9 2D discrete kernel.

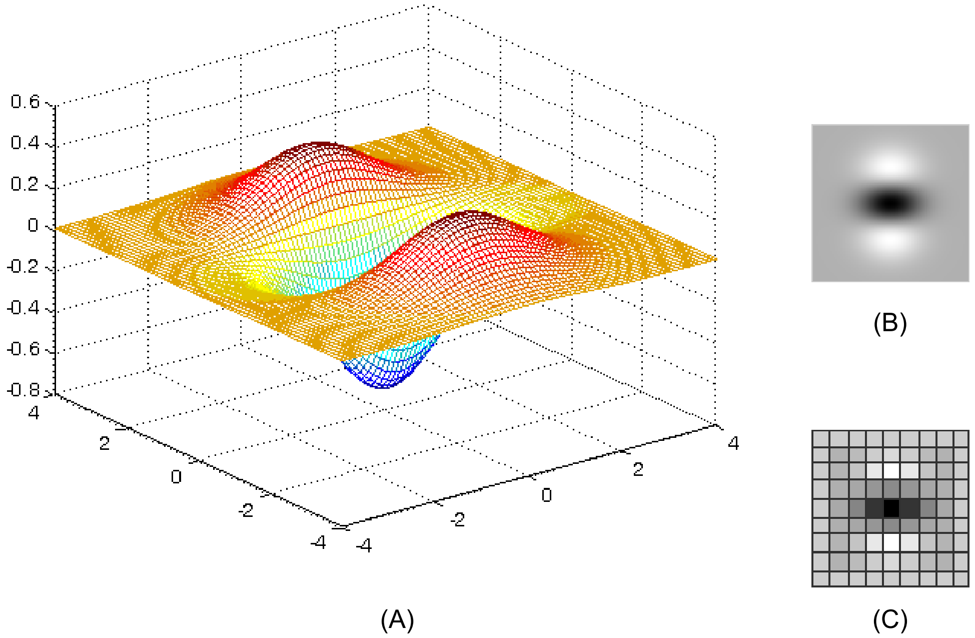
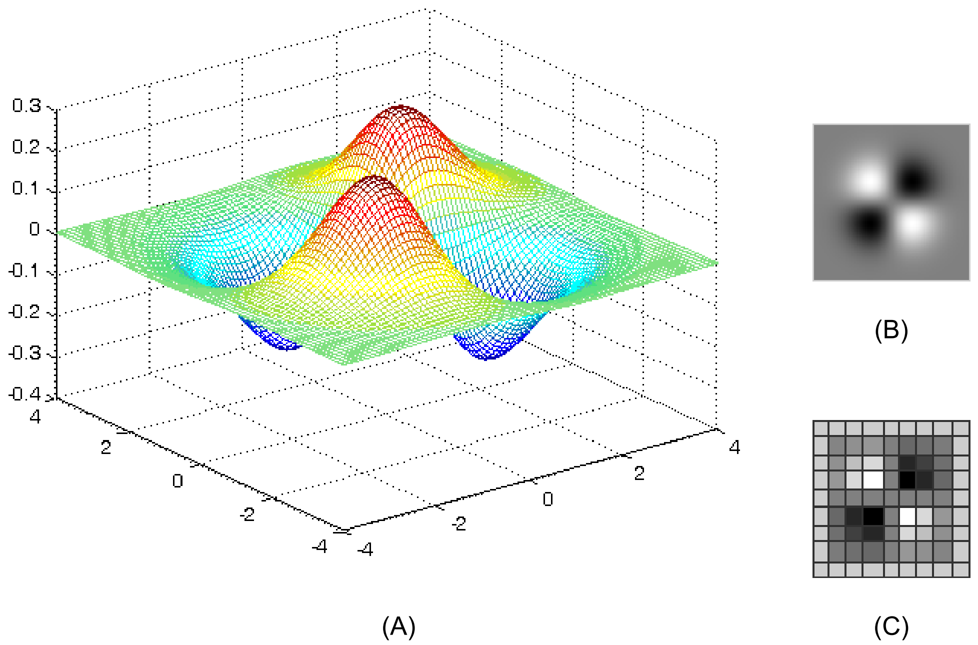


Figure 2.9: Gaussian derivative in y-direction. (A) 3D plot of the continuous filter. (B) 2D continuous kernel. (C) 9x9 2D discrete kernel.



(B)

(A)

(C)

Figure 2.10: Gaussian derivative in xy-direction. (A) 3D plot of the continuous filter. (B) 2D continuous kernel. (C) 9x9 2D discrete kernel.

reasonably size image with a 50x50 array may demonstrate that it is an unattractive procedure. A possible alternative is to perform repeatedly the convolution with much smaller filters, which is much more efficient because it is not necessary to keep a large amount of pixels in the memory [41].

Gaussians are not the only low-pass filter used for smoothing images and constructing a scale-space representation. However, several mathematical results, such as those described by Koenderink [42] and Babaud *et al.* [43] demonstrate that, within the class of linear transformations, the Gaussian kernel is the unique kernel for generating a scale-space. According to Linderberg [44], the conditions that specify the uniqueness are essentially linearity and shift invariance:

- Linearity: $G_\sigma(aF + bH) = aG_\sigma F + bG_\sigma H$, where a and b are constants and F and H are signals;
- Shift invariance: $G_\sigma S_{(\partial x, \partial y)} F = S_{(\partial x, \partial y)} G_\sigma F$, where $S_{(\partial x, \partial y)}$ denotes the translator operator.

But the Gaussian kernel also satisfies a number of other properties (scale-space axioms) that make it a special form of multi-scale representation, like:

- Non-creation of local extrema (zero-crossings) in one dimension;
- Non-enhancement of local extrema in any number of dimensions
- Rotational symmetry

- Scale invariance
- Positivity
- Normalization

The Figures 2.8, 2.9 and 2.10 plot the 3D and the 2D Gaussian second order derivatives in x, y and xy directions. Gaussians may be optimal for scale-space analysis, but its continuous nature is not practical in computer vision applications. Thus, the intensity of each pixel in the discretized Gaussian kernel. The Figures 2.8 C, 2.9 C, and 2.10 C define the filter to approximate a Gaussian second order partial derivative.

2.7 Random Sample Consensus

The RANdom SAmple Consensus (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers [45]. First, RANSAC selects a random subset of the original data (hypothetical inliers). Next, the algorithm estimates the parameters of the model which can explain the observation. All points are tested against the fitted model, and those which fit well are also considered hypothetical inliers. Later, this extended set of points is used to re-estimate the parameters. The model is then evaluated by computing the error relative to the hypothetical inliers. This procedure is repeated a fixed number of times. Finally, the model hypothesis with lowest error is selected. The number of hypothesis (iterations) N necessary to guarantee, with probability p, that a correct solution is found can be computed by:

$$N = \frac{log(1-p)}{log\,(1-\omega^s)} \tag{2.52}$$

where ω is the assumed inlier ratio (number of inliers in data / number of points in data) and s the minimal number data points needed to estimate the model.

2.8 Image Features

Current models of human visual system suggest that our visual attention is a bottom-up process [46]. It starts with the unconscious detection of all salient image details, which are image patterns whose low-level features (i.e size, shape, luminance, color, direction, texture, binocular disparity) differs significantly from its immediate neighborhood [46]. The visual focus is then sequentially shifted to each of these regions, so they can be analyzed in detail. Computational models based on information theory have been shown to successfully model human salience. In computer vision and image processing, such image details are usually referred to as local image features, which can be points, edges, T-junctions, lines, contours, blobs or image patches. The use of high-level information, though, is usually avoided due to the fragility of segmentation algorithms.

Local image features provide a limited set of well localized and individually identifiable anchor points. What the features actually represent is not really relevant, as long as their location can be determined accurately and in a stable manner over time. For this reason, it is extremely important for extracted features to be robust to noise and invariant with respect to geometrical (i.e. changes in scale, translation, rotation, affine/projective transformation) and photometric variations (illumination direction, intensity, color, and highlights) [47, 48]. According to Tuytelaars *et al.* [49], salient features should hold the following properties:

- Repeatability: is a property related to the stability of the detected interest points. An interest point is repeated if it is accurately detected in the different images of the same scene. The repeatability rate is a measure of the stability, and corresponds to the percentage of the total observed points that are detected in both images. Repeatability is considered the most important property of a feature detector, and can be achieved either by invariance or by robustness. Invariance is provided by the mathematical model used to compute the detector response, and consists in the capability to yield a constant response in the presence of large variations in scale, translation, rotation, illumination and distortion. Robustness, on the other hand, is concerned with the ability to make feature detection methods less sensitive to small deformations. Typical deformations that are tackled using robustness are image noise, discretization effects, compression and blur.

- Distinctiveness: is the capacity of the detected features to present intensity patterns with significant variations. This property is important in the sense that it allows a single feature to be correctly matched with high probability against a large database of features.

- Locality: features may be defined in terms of its local neighborhood. The detection of features through local operations reduces the probability of occlusion due to camera motion. In addition, it allows to use simpler models to approximate the geometric and photometric deformations between two images.

- Quantity: the optimal number of features depends on the application and the scene conditions, and should be adaptable over a large range by a threshold. The number of detected features should be large enough to describe even small objects, but small enough to not compromise the algorithm efficiency.

- Distribution: the density of features should reflect the information content of the image to provide a compact image representation. However, image transformations can not be properly estimated when features are concentrated in small image regions.

- Accuracy: the detected features should be accurately localized, both in image location and scale.

- Efficiency: regards to the computational simplicity of the algorithm. This property can be a determinant factor in the choice of the proper feature detector. Preferably, the detection of features in a new image should allow for time-critical applications.

Although these theoretical properties make the Local image features suitable to be used as visual landmarks, in practice, such stability is not always achieved. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. The four most popular feature detectors used in vision-based localization methodologies will be summarized throughout the next subsections.

2.8.1 Harris Corner

The Harris corner detector was proposed by Harris and Stephens [50], and relies on the principle that at a corner the image intensity will change largely in multiple directions. The algorithm is based on the second moment matrix M, also called the autocorrelation matrix, which describes the gradient distribution in a local neighborhood of a pixel located at (x,y):

$$M(x,y) = \begin{bmatrix} \left(\frac{\partial I(x,y)}{\partial x}\right)^2 & \frac{\partial I(x,y)}{\partial x}\frac{\partial I(x,y)}{\partial y} \\ \frac{\partial I(x,y)}{\partial x}\frac{\partial I(x,y)}{\partial y} & \left(\frac{\partial I(x,y)}{\partial y}\right)^2 \end{bmatrix} \tag{2.53}$$

The local image derivatives are computed with Gaussian kernels (Equation 2.50) of scale σ_D (the differentiation scale). The derivatives are then averaged in the neighborhood of the point by smoothing with a Gaussian window of scale σ_I (the integration scale).

$$M = {\sigma_D}^2 g(\sigma_I) * \begin{bmatrix} {I_x}^2(x,y,\sigma_D) & I_x(x,y,\sigma_D)I_y(x,y,\sigma_D) \\ I_x(x,y,\sigma_D)I_y(x,y,\sigma_D) & {I_y}^2(x,y,\sigma_D) \end{bmatrix} \tag{2.54}$$

where $g(\sigma)$ is the Gaussian kernel, and

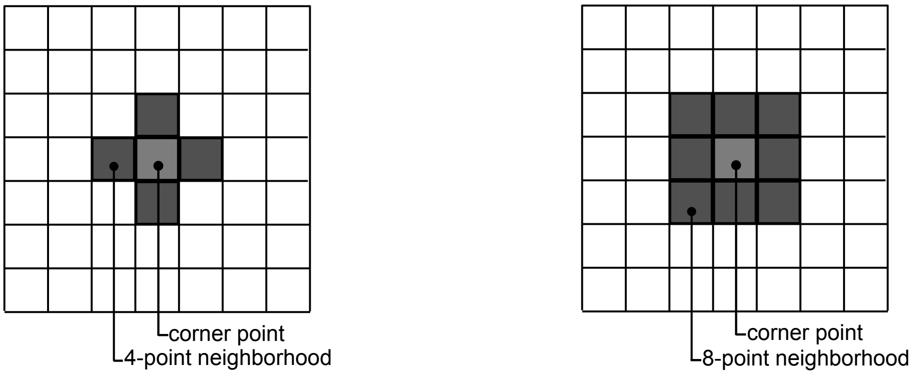
$$I_x(x,y,\sigma_D) = \frac{\partial}{\partial x,y} g(\sigma_D) * I(x,y) \tag{2.55}$$

The eigenvalues of this matrix represent the principal signal changes in two orthogonal directions in a neighborhood around the point defined by σ_I. Let λ_1 and λ_2 be the eigenvalues of the matrix $M(x,y)$, the Harris corner detector defines the autocorrelation function R as:

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_1)^2 \tag{2.56}$$

This function will be sharply peaked if both of the eigenvalues are high, which means that shifts in any direction will produce a significant increase, indicating that it is a corner. Since the explicit decomposition of the eigenvalues is computationally expensive, Harris and Stephens suggested to approximate the autocorrelation function using the determinant and the trace of the second moment matrix:

$$R(x,y) = det(M) - k\, trace(M) \tag{2.57}$$



(a) (b)

Figure 2.11: Definition of pixel connectivity: (a) 4-neighbor pixels are those who share an edge; (b) 8-neighbor pixels are those who either share an edge or a vertex.

with $det(M)$ the determinant and $trace(M)$ the trace of the matrix M. The value of k is determined empirically, and usually set to 0.04. Adding the trace reduces the response of the operator on strong straight contours. When used as an interest point detector, local maxima of the cornerness function are extracted through non-maximum suppression of the pixel direct (Figure 2.11a) or indirect neighborhood (Figure 2.11b).

The algorithm describes the local neighborhood of a point by directly storing the raw image intensity values from a small square window around the detected point.

2.8.2 Harris-Laplace

Harris corner detector has shown its ability to identify interest points of the image with rotational invariance. One severe limitation of the algorithm, though, is the lack of invariance to changes in scale. Thus, Mikolajczyk *et al.* [51] proposed the Harris-Laplace detector, which is a twofold process that combines the traditional 2D Harris corner with a Laplacian scale-space operator.

The theory of scale-space shows that in addition to the two-dimensions of the (x,y) image position, a third dimension (scale) can be constructed to estimate the appearance of the image as if seen from further away. To address such multi-scale analysis, Burt and Adelson [52] have proposed a simple but yet powerful representation, in which the scale-space can be represented by a collection of decreasing resolution images arranged in the shape of a pyramid (Laplacian pyramid). The main advantage of this representation is that the image size decreases exponentially with the scale level, and hence also the amount of computations required to process the data. As represented in the Figure 2.12, the base of the pyramid contains a high resolution representation of the image under analysis (first level). Then, each subsequent level is obtained by successively reducing the image size by smoothing and sub-sampling the previous image. Thus, as you move up the pyramid apex, both image size and resolution decrease.

Harris-Laplace locates local features with the Harris corner detector in all scales of the image pyramid. Next, the algorithm selects the points for which the Laplacian attains a maximum over

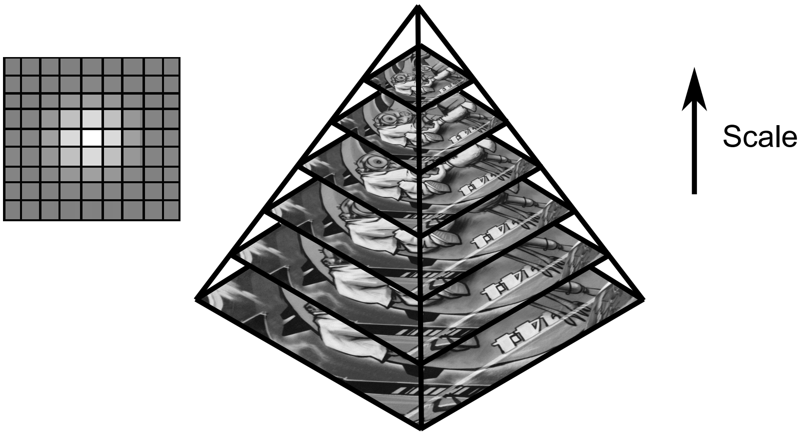


Figure 2.12: Typical scale-space representation arranged in the shape of pyramid.

the scale.

2.8.3 Scale-Invariant Feature Transform

The Scale-Invariant Feature Transform (SIFT) is a feature detection and description algorithm presented by Lowe [53]. The interest points extracted are said to be invariant to image scale, rotation, and partially invariant to changes in viewpoint and illumination. SIFT features are located at maxima and minima of a difference of Gaussians (DoG) function applied in scale space. Finding these principal curvatures amounts to solving for the eigenvalues of the second-order Hessian matrix H. The Hessian matrix can be described as the square matrix of second-order partial derivatives of a function. Considering a continuous function f(x,y) of two variables, the Hessian matrix H can be defined as:

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 I(x,y)}{\partial x^2} & \frac{\partial^2 I(x,y)}{\partial xy} \\ \frac{\partial^2 I(x,y)}{\partial xy} & \frac{\partial^2 I(x,y)}{\partial y^2} \end{bmatrix} \tag{2.58}$$

The trace of H, Equation (2.59), gives us the sum of the two eigenvalues, while its determinant, Equation (2.60), yields the product. The SIFT response R, Equation (2.61), depends only on the ratio of the eigenvalues rather than their individual values. Therefore the higher the absolute difference between the two eigenvalues, the higher the value of R.

$$trace(H(x,y)) = \left(\frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \right) \tag{2.59}$$

$$det(H(x,y)) = \frac{\partial^2 I(x,y)}{\partial x^2} \frac{\partial^2 I(x,y)}{\partial y^2} - \left(\frac{\partial^2 I(x,y)}{\partial xy} \right)^2 \tag{2.60}$$

$$R(x,y) = \frac{(trace\,(H(x,y)))^2}{det\,(H(x,y))} \tag{2.61}$$

The description of SIFT key points is performed based on the local image information at the key point's characteristic scale. First, to provide rotation invariance, SIFT assigns a global orientation to each point based on local image gradient directions. Next, each point is used to generate a feature vector that describes the local image region sampled. The descriptor is computed based on a set of orientation histograms at a 4x4 subregion around the interest point. Since there are 4 x 4 = 16 histograms, each with 8 bins, the final descriptor vector presents 128 elements. Illumination invariance comes at the cost of additional computation, in which the descriptor is normalized by the square root of the sum of the squared components.

2.8.4 Speeded Up Robust Feature

Speeded Up Robust Feature (SURF) is a detector and descriptor algorithm that was designed to deal with scale and rotation invariant features over variations in the camera's point of view [54, 55]. It is based on sums of bidimentional Haar wavelet responses and makes an efficient use of integral images. The algorithm can be divided into two main steps: interest point detection and interest point description.

The detection step of the SURF algorithm takes advantage of the good performance and accuracy of the Hessian matrix, Equation (2.58), to detect blob-like structures. Informally, the eigenvectors of the Hessian matrix represent the dominant edge orientations of the window, and the eigenvalues represent the amount of energy along these orientations. The SURF detector looks for blob-like structures that have significant amounts of energy in all directions. Since the determinant of the Hessian matrix is the product of its eigenvalues, it is possible to classify the points based on the sign of the result. A negative determinant means that the eigenvalues do not have the same signal, and thus the point is not local maximum. On the other hand, a positive determinant means that the signal is the same for both eigenvalues, and so the point is a local maximum. The determinant of the Hessian matrix can be expressed according to:

$$det(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 \tag{2.62}$$

As previously discussed in Section 2.5, the derivatives of the Hessian matrix can be calculated using standard convolution methods. However, once the kernels derived from the second central difference are approximating a second derivative measurement on the image, they are very sensitive to noise. For this reason, Bay *et al.* [55] propose to substitute the second central difference for a Gaussian kernel of second order.

This way, the Hessian matrix H(X_p) of a given point X_p=(x,y) from the image I(x,y), can be re-written as function of both space X_p=(x,y) and scale σ:

$$H(G(x,y,\sigma)) = \begin{bmatrix} \frac{\partial^2 g(\sigma)}{\partial x^2} & \frac{\partial^2 g(\sigma)}{\partial x \partial y} \\ \frac{\partial^2 g(\sigma)}{\partial x \partial y} & \frac{\partial^2 g(\sigma)}{\partial y^2} \end{bmatrix} \tag{2.63}$$

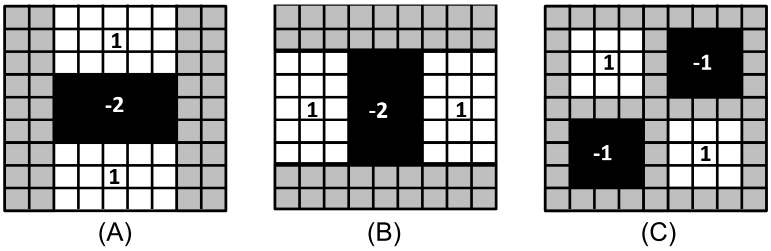


Figure 2.13: Box plot filters: approximations of the second order Gaussian partial derivative in the (A) y-direction (D_{yy}), (B) x-direction (D_{xx}) and (C) xy-direction (D_{xy}).

where, $\frac{\partial^2 g(\sigma)}{\partial x^2}$ is the convolution of the Gaussian second order derivative in the x-direction (Figure 2.8) at the point X, $\frac{\partial^2 g(\sigma)}{\partial y^2}$ the convolution of the Gaussian second order derivative in the y-direction (Figure 2.9) and $\frac{\partial^2 g(\sigma)}{\partial x \partial y}$ the convolution of the Gaussian second order derivative in the xy-direction (Figure 2.10).

The drawback of this approach is the number of operations necessary to perform the convolution, which is equal to the number of pixels within the filter. For example, in a small 9x9 filter, 81 memory accesses are required, but this number can easily grow to 2500 (50x50 pixels) and 9801 (99x99 pixels) as the filter side length increases.

In order to improve the efficiency of the algorithm, SURF pushes the discretization further and approximates the second order Gaussian derivatives (and thus the Hessian matrix determinant) using box type filters (Figure 2.13). Box filters are spatial averaging filters in which all coefficients are equal. Thus, each second order Gaussian derivative filters can be approximated by using of three (x-direction and y-direction) or four boxes (xy-direction) with different weights. The approximation of Gaussians by box filters provides SURF the ability to use integral images, and thus perform fast convolutions with filters of several sizes at constant time. To illustrate the benefits of integral images, we can recover the previous example about number of memory access required to perform the convolution of a 9x9 Dyy filter. While 81 memory accesses are needed to compute the response of the discretized and cropped version of the second order Gaussian filter, only 8 memory accesses are demanded by the respective box type filter. Furthermore, while for the original Gaussian filter the number of memory accesses increases with the filter size, it remains constant for box like filters of any size.

For both D_{xx} and D_{yy} filters, white regions are weighted 1, black regions weighted -2 and gray regions zero. For the D_xy filter, white regions are weighted 1, black regions weighted -1 and gray regions zero. The weights applied to the rectangular regions are kept simple for computational efficiency. This way, the determinant of the Hessian matrix can be approximated as:

$$det(H_{approx}) = D_{xx}D_{yy} - (0.912D_{xy})^2 \tag{2.64}$$

where the constant 0.912 represents a relative weight necessary for the energy conservation between real Gaussian kernels and the approximated Gaussian kernels. The determinant is referred

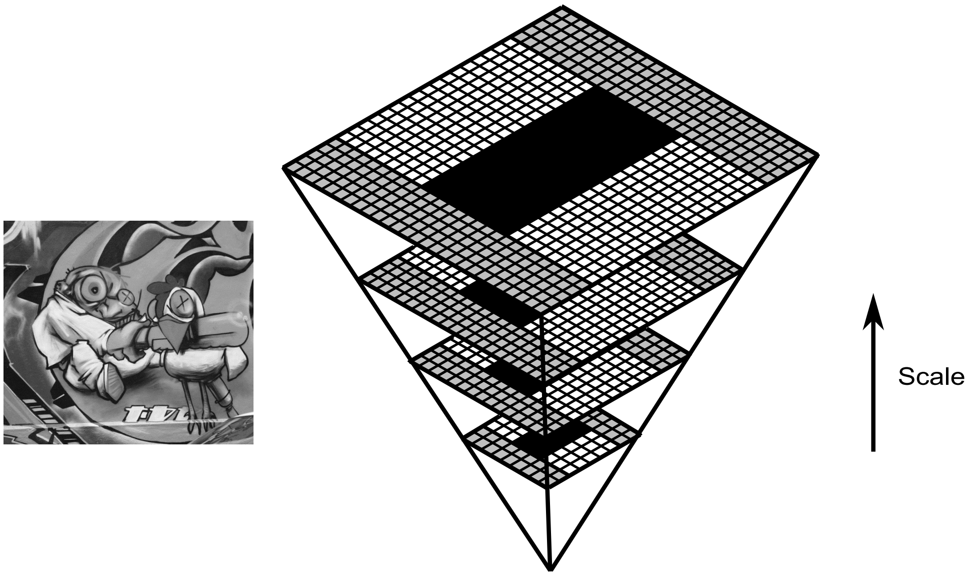


Figure 2.14: SURF scale-space: instead of smoothing and sub-sampling the input image, SURF leaves the image size unchanged and varies only the size of the filters (adapted from [55]).

to as the blob response at the location $X = (x, y, \sigma)$.

The notion of scale is extremely important for feature detection, and the explanation is very simple. Features in a given image, just like any objects in the real world, are only meaningful at certain ranges of scale. In fact, features appear differently according to the scale they are observed due to surface textures and perspective effects. Thus, the selection of the right scale is not only utmost important for the detection, but also for its description, allowing features to be matched across images taken at different zoom levels or distances.

Based on classical scale-space representation, Bay *et al.* [55] proposes an alternative structure. In their innovative approach, the scale-space is represented through a collection of filters with increasing sizes, arranged in the shape of an inverted pyramid (Figure 2.14). The inverse pyramid approach is very advantageous over its counterpart once that it is possible to eliminate the computational overhead of smoothing and up-scaling the input image.

A second advantage consists in the fact that the image is not subjected to the aliasing effect, unlike those sub-sampled images of the usual scale-space implementation. Furthermore, once the pyramid layers do not depend from its previous, they can be processed in parallel (which could have a significant impact in the overall computational efficiency). On the other hand, the box filter approach presents a disadvantage concerning the scale-invariance of the detected interest points. In fact, once box type filters preserve high frequency components (sharp transitions between regions) they can get lost in zoomed-out variants of the same scene (thus limiting the feature's scale-invariance).

In SURF, the scale-space is subdivided into a number of structures called octaves. Each octave represents a series of filter responses obtained by the convolution of the input image with filters of increasing sizes. The number of filters in each octave is kept constant, and called scale levels. The value of the pair octave-level determines the size length of the filter that will be convolved with

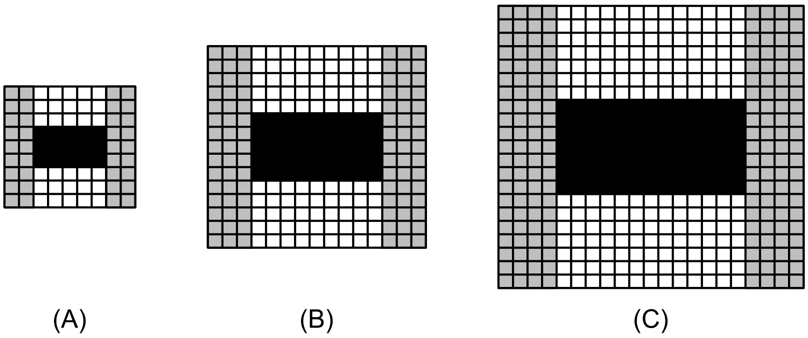


Figure 2.15: Sequence of the first three D_{yy} filters of the first octave.

the image at that given layer in the scale-space.

$$FilterSize = 3(2^{octave} level + 1) \tag{2.65}$$

The lowest scale is obtained from the output of the $9x9$ filter, which approximates a Gaussian derivative with $\sigma = 1.2$. When constructing the octaves, the increase in the filter's size is restricted by the layout ratio of the second order Gaussian derivative. Due to the presence of a central pixel in each lobe its dimensions must increase equally around this location, so the minimum increase in the lobe size is 2 pixels. Since each filter presents three lobes, the minimum increase in the filter size is of 6 pixels. Thus the filter sequence in the first octave is 9, 15, 21, 27 (Figure 2.15). For each new octave, the filter size increase is doubled. Therefore, the second octave, that starts with a filter of $15x15$ pixels and presents a filter size increase of 12 pixels per level, has a filter sequence of 15, 27, 39, 51. The third octave, which starts with a filter of size 27 and presents a filter size increase of 24, has a sequence of 27, 51, 75, 99 and so on. As the filter increases so does the corresponding scale of the Gaussian derivative. An approximation of the Gaussian scale can be estimated by:

$$\sigma_{approx} = FilterSize \frac{Basefilterscale}{BasefilterSize} \tag{2.66}$$

The position of the upper-left and bottom-right corner of each region of the filter can be determined according to the position (x,y) of the mask's central pixel and the length l_0 of the positive or negative lobe of the partial second order derivative (shortest side of the weighted black and white regions), in the direction of the derivation (x or y):

$$l_0 = \frac{FilterSize}{3} \tag{2.67}$$

Typically, the approach used to detect features consists on applying the filter over different scales and detect those with maximum response (Figure 2.16).

After a threshold operation, a non-maximum suppression is applied both spatially and over the neighboring scales (Figure 2.17). In this step, each pixel is compared in a $3x3x3$ neighborhood, and

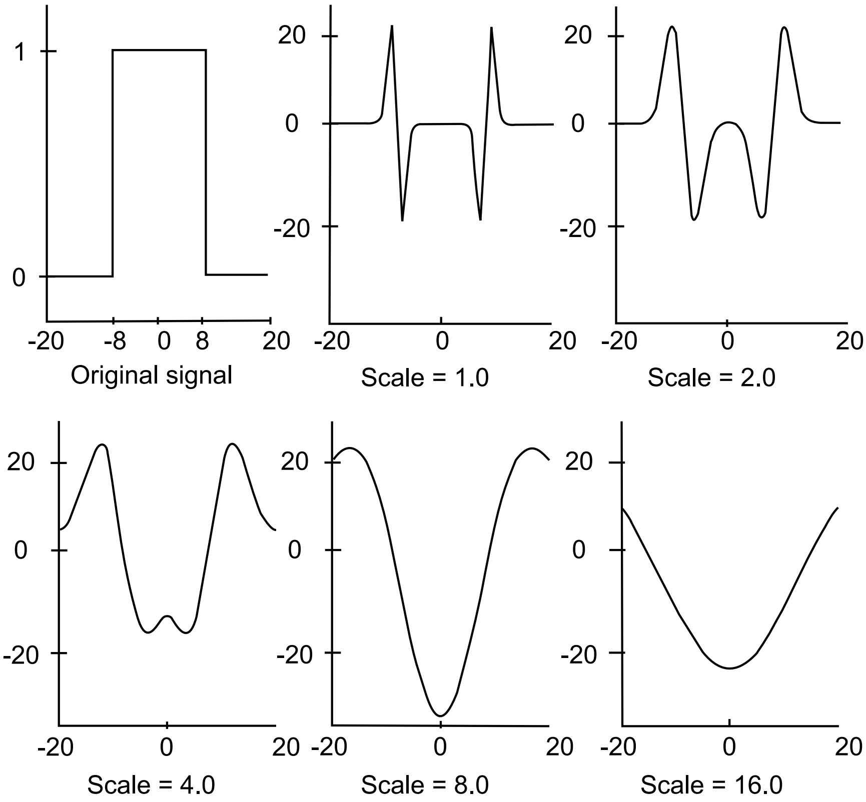


Figure 2.16: Left to right: the original signal and its respective blob response for several scales. The feature is recognized at the scale which provides the maxim blob response ($\sigma = 8.00$ in this example).

classified as interest points if its blob-response is greater than the blob-response of its 26 neighbours (8 pixels in the native scale, 9 pixels in the immediately lower and 9 pixels in immediately higher scale). The scale at which a maximum response over scales is attained will be assumed to give information about how large a feature is. Hence, the first and the last Hessian response map of each octave cannot contain such maxima themselves, as they are used for comparison purposes only.

Finally, the interest point localization uses 3D interpolation method proposed by Brown and Lowe [56]. This methodology fits a 3D quadratic function to the local sample points to determine the interpolated location of the maximum. The interpolation of the features blob-response in scale and space substantially improve to feature's matching and stability by yielding a location with sub-pixel/sub-scale accuracy. Brown expresses determinant of the Hessian function $H(x,y,\sigma)$ as a Taylor expansion up to quadratic terms.

$$H(X) = H + \frac{\partial H^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 H}{\partial X^2} \tag{2.68}$$

where H and its derivatives are evaluated at the sample point and $X = (x,y,\sigma)$ is the offset from this point. The location of the interpolated interest point $\hat{X} = (x,y,\sigma)$ is taken as the extremum of this 3D quadratic, and can be determined by setting to zero the derivative of the Equation 2.68.

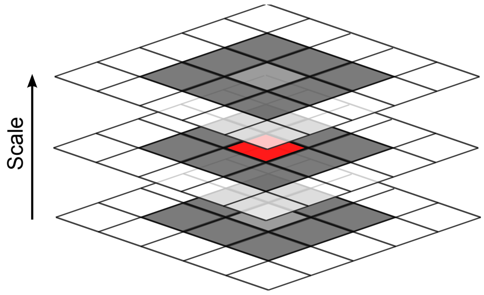


Figure 2.17: Non-maximum suppression in a 3x3x3 neighbourhood. The central pixel is only considered a feature point if its blob-response is greater than its 26 neighbours.

$$\hat{X} = -\left(\frac{\partial^2 H}{\partial X^2}\right)^{-1} \frac{\partial H}{\partial X} \tag{2.69}$$

As suggested by Brown, the Hessian and derivative of H are approximated by using differences of neighboring sample points. The resulting 3x3 linear system can be solved with minimal cost, computing the entries of the $3x3$ matrix $\frac{\partial^2 H}{\partial X^2}$ and the 3x1 vector $\frac{H}{\partial X}$:

$$\frac{\partial^2 H}{\partial X^2} = \begin{bmatrix} d_{xx} & d_{yx} & d_{\sigma x} \\ d_{xy} & d_{yy} & d_{\sigma y} \\ d_{x\sigma} & d_{y\sigma} & d_{\sigma\sigma} \end{bmatrix} \tag{2.70}$$

$$\frac{\partial H^T}{\partial X} = \begin{bmatrix} d_x \\ d_y \\ d_\sigma \end{bmatrix} \tag{2.71}$$

where d_x refers to $\frac{\partial I}{\partial X}$, d_{xx} refers to $\frac{\partial^2 I}{\partial X^2}$, and so on. If the offset $\hat{X}$ is larger than 0.5 in any dimension, then it means that the extremum lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed instead about that point. The final offset $\hat{X}$ is added to the location of its sample point to get the interpolated estimate for the location of the extremum.

A SURF feature descriptor is a vector of 64 elements that describes how the intensity of the pixels in the neighbourhood of an interest point is distributed. In order to build the descriptor, the algorithm first determines the feature orientation in the scale-space that the feature has been detected. Then, the algorithm constructs a square window aligned to the selected orientation to extract the descriptors. Thus, once the descriptors are always computed relatively to the predominant direction, it is possible to compare the descriptors of interest points detected in frames with different orientation, and so providing orientation invariance to detected features.

The first step to characterize an interest point consists on finding its predominant orientation of the intensities in its neighborhood. The predominant orientation is important in order to achieve invariance to image rotation, and is built on the response distribution of Haar wavelet filters. First order Haar wavelets (Figure 2.18) are simple filters which can be used to find gradients in the x

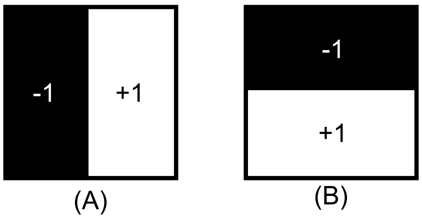


Figure 2.18: Haar wavelet filters: gradient in the x-direction (A) and y-direction (B).

(A) and y (B) directions. In these filters, white regions are weighted 1, while black regions are weighted -1.

To determine the interest point orientation, the algorithm computes the response of both Haar filters (x and y directions) for a set of sampled pixels within a circular neighborhood around the interest point. It is important to note that in order to keep scale invariance, the sampling step size, the side length of the Haar filters and the radius size of the circular neighborhood depends on the scale σ in which the feature was detected, and are respectively σ, 4σ and 6σ. One more time, the use of integral images contribute to the algorithm efficiency, once only six operations are needed to compute the Haar wavelet response in x or y direction at any scale.

Once calculated, the Haar responses are weighted with a Gaussian function $(SD = 2\sigma)$ centered at the interest point. The Gaussian function is extremely important in many areas due to its significance as the probability density function for the normal distribution. As a weighting function, it expresses the idea that points close to the center have more relevance than the distant points. The weighted responses are then represented as points in the space, with the x-responses along the horizontal axis and y-response along the vertical axis of the Cartesian coordinate system. Finally, the dominant orientation is estimated by rotating a sliding window (typically a circular arc with central angle of $\frac{\Pi}{3} radians$ with a step size of 0.1 radians, Figure 2.19. In each step, the algorithm computes the sum of all x-responses and of all y-responses within the window to determine a local orientation vector. The longest vector of all windows defines the orientation Θ of the interest point.

The statistics of the gradient in an image neighborhood yields quite a useful description of the neighborhood.The extraction of SURF descriptors starts with the construction of one squared window for each interest point detected (descriptor window). This window contains the pixels that will be analyzed to compute the feature's descriptors. Each descriptor window is centered at the interest point, and oriented along the feature's orientation. In order to keep scale invariance, the side length l_w of each window varies linearly with the scale in which the feature was detected, and corresponds:

$$l_w = 20\sigma \tag{2.72}$$

Descriptor windows are regularly sub-divided into $4x4$ smaller sub-regions. Each sub-region contains local spatial information that provides to the feature four descriptors. In order to determine these descriptors, the response of Haar wavelets of size 2σ for both x and y-directions (d_x and d_y respectively), are computed for 25 regularly spaced sample points (Figure 2.20). The

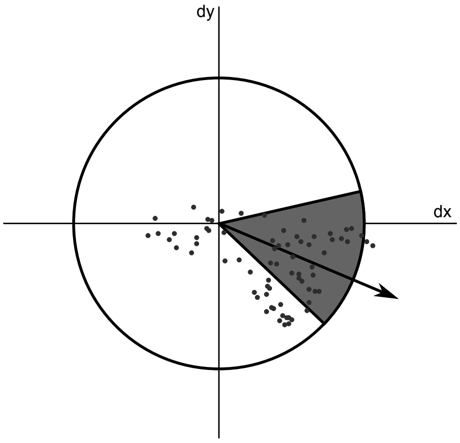


Figure 2.19: Predominant interest point orientation (adapted from [55]).

use of such wavelets provides the descriptor with important characteristics, like the invariance to bias in illumination (offset), and invariance to contrast (scale factor), which is achieved by simply normalizing the descriptor vector.

It is important to note that the horizontal wavelet response d_x and the vertical wavelet response d_y are defined in relation to the selected interest point orientation, preventing the use of integral images, once integral images only retrieves upright rectangular areas. This way, the algorithm first extracts the descriptor window W(y,x) containing all $20\sigma^2$ pixels around the keypoint:

$$x_s = x + \frac{l_w}{2}\cos\theta + \frac{l_w}{2}\cos\theta \tag{2.73}$$

$$y_s = y - \frac{l_w}{2}\sin\Theta + \frac{l_w}{2}\cos\Theta \tag{2.74}$$

$$W(y,x) = I(y_s, x_s) \tag{2.75}$$

Later, the descriptor window is then scaled to the P of size 20, so that each pixel's size is σ, facilitating the computation of the gradients in x and y:

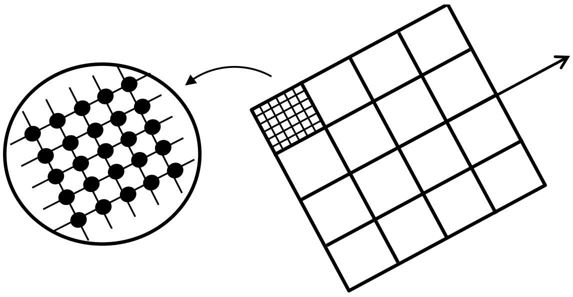


Figure 2.20: Feature's orientation: in detail the 25 regularly sampled points used to compute the first four-dimensional descriptor vector.

$$D_x(i,j) = P(i,j+1) - P(i,j) + P(i+1,j+1) - P(i+1,j) \tag{2.76}$$

$$D_y i,j) = P(i+1,j) - P(i,j) + P(i+1,j+1) - P(i,j+1) \tag{2.77}$$

Finally, the responses are weighted with a 2D Gaussian function, of standard deviation $SD = 3.3\sigma$, to increase robustness towards geometric deformations. The sum of the weighted wavelets responses d_x and d_y over each sub-region yields the first set of four entries in the feature vector:

$$V = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_x|\right) \tag{2.78}$$

2.9 Conclusions

Color information can be used in image processing to simplify the identification and extraction of objects from a scene. Color images carry more information than gray level images, and thus provide a broader class of discrimination between material boundaries [57]. In addition, color information enables one to distinguish between true color variation and photometric distortions [58]. Thus, when colored images are represented only through their intensity value, a very important source of information is lost [48].

Linear color spaces such as RGB and XYZ, presents high correlation on its color components. This correlation makes the three components dependent upon each other and strongly associated with the light intensity. Because of such association, all three color components change according to variations in the illumination of the scene, causing of severe instability in color matching methodologies. Hence, when working with linear color spaces it is very difficult to discriminate highlights, shadows and shading [27]. Thus, all the gain of information provided by the chromatic dimensions might be useless once varying lighting conditions affects the observed colors. Indeed, photometric invariance is less trivial to achieve, but utmost important when dealing with such as a changes in illumination color, illumination direction, and camera viewpoint.

One possible way to deal with varying illumination conditions is to work in those color spaces that decouple lightness, thus, applying the algorithms that were originally developed to gray scale images in hue component only. However, decoupling the illumination through a non-linear transformation (like in the nRGB color space) has some drawbacks: the normalization reduces the sensitivity of the distribution to the color variability and introduces noise in pixels with low intensities [27, 28]. Hue is particularly useful in the cases where the illumination level varies from point-to-point or image-to-image because when the white reference holds, hue is invariant to certain types of highlights, shading, and shadows [27]. Furthermore, material boundaries correlate more strongly with hue than with intensity differences. Shadow boundaries, highlight boundaries and transparency boundaries are strongly associated with intensity edges, and less with hue boundaries.

But while the image definition in terms of hue, saturation, and value is mathematically valid, it is important to note that they are only approximations of the human color perception [32]. One well-known problem of the HSV and related color spaces relies on the non-removable singularity at the achromatic axis, where R = G = B = 0. A small change in any of the R, G, or B components may cause a large variation in the transformed values, creating discontinuities in the representation of colors. When the saturation is low, the hue component gets unstable, and thus, unreliable to describe pixels [27]. Take for example, pixels whose HSV values are $(x, x, 0)$ where $x \geq 0$. For 8-bit digital implementation (i.e., hue in the range of $[0, 255]$) a minimal digital perturbation gives $(x+1, x, 0)$, which can result in changes up to $^{\Pi}/_{3}$.

Hue based models present another significant shortcoming with respect to the definition of saturation. Mathematically, saturation is defined as a percentage of the maximum saturation in a given intensity. Through the hexconic representation we may note that, according to the intensity, the maximum value for the saturation varies from 0 (apex) to 1 (base) in the HSV model, while varies from 0 (lower apex) to 1 (center) to 0 (upper apex) in the HSL and HSI. It means that in the HSV the intensities close to the black vertex present a saturation range very small. In the same way, it means that for the HSL and HSI models, not only very low intensities but also very high intensities present such small saturation range. Thus, slight variations of intensity in those regions may imply in a significant variations in the saturation component. In practice, such variations are responsible to the introduction of noise for dark regions in the HSV (i.e. shadows), and for dark and light regions in the HSL and HIS color models (i.e. reflections) [34].

Despite of the perceptual uniformity, CIE hue only approximates additive/shift invariance due to its non-linear cube-root transformation and normalization [28]. Moreover, the assumption that CIE-Lab and CIE-Luv colors are uniformly distributed is only partially valid. Actually, the equivalence between Euclidean and perceptual distances holds only for small distances. For larger distances, the most we can say about a pair of colors is that they are different, thus simply measuring Euclidean distance in CIE-Lab is insufficient to accurately describe a color [59].

Throughout this thesis it is assumed that the relationship between the energy measured by the sensor and the image data is linear. In fact, several of the methods, deductions and formalities discussed are only valid in scenarios were this premise is true. Since the sRGB color space is device independent and allows to exchange images between different machines, it is reasonable to assume that the image was stored using this representation [60] in situations were nothing is known about the process on how the image was created. Therefore, before any other kind of image processing, one needs to restore the linear relationship between image data and measured intensities.

Regarding the Diagonal Model, several works [61, 62, 63, 64, 65, 66, 67] point out that the efficiency of the model is intrinsically related to the sensors of the vision system. It was demonstrated that, whether or not the sensors are narrow band and whether or not their sensibility functions overlap, has a significant impact on the accuracy of von Kries adaptation. Sensors that do not behave as delta functions result in non-zero off-diagonal elements in the transformation matrix (relating object reflectance to receptor stimulation), which prevents the Diagonal Model to hold properly.

It has been shown in the literature that it is possible to overcome such camera issues through the use of narrow band illumination. In [61], Worthey demonstrates that narrow band illumination in "not so narrow band" sensors produces similar effects as broad band illumination in narrow band sensors. Nevertheless, this method is not feasible in practice, since most of the time it is not possible to control the illuminant of real world images. One method, however, that improves the results in real world images is the spectral sharpening. When sensors are not completely narrow band it is possible to simulate this behavior by artificially sharpening the camera sensor response through linear transformations of the sensor functions [68, 69, 70].

Concerning the local image feature detectors, Harris corners demonstrate itself as an attractive option since it is of simple implementation, and invariant to translations and rotations. The lack of invariance to scale, though, is one limitation of the original algorithm. Harris-Laplace approach, on the other hand, solves this problem by selecting the points in the multi-scale representation which are present at characteristic scales. Moreover, the algorithm handles the problem of affine invariance by estimating the affine shape of a point neighborhood.

One drawback of both algorithms consists in the way local feature points are described. Harris describes the local neighborhood of a point by directly storing the raw image intensity values from a small square window around the point. This has the advantage of simplicity of computation, but it lacks invariance to lighting, rotation, and viewpoint changes. Another shortcoming of the multi-scale Harris detector is that the algorithm extracts many points which are repeated at the neighboring scale levels. This in turn increases the matching complexity and the probability of mismatches. [49].

SIFT and SURF demonstrated to be very robust detectors, with invariance to several image transformations. When concerning only with robustness and repeatability, SIFT seems to outperform SURF in several scenarios. However, there are other factors that should be considered in the election of a detector and descriptor for visual odometry. An important parameter in this selection is the computational cost of the detection and description methods. The SIFT algorithm is, therefore, less suited to continuous tracking of real-time computer vision tasks due to the high computational cost of its extraction, description and matching stages. SURF, on the other hand, has a lower computational cost compared to SIFT [55], and its simplified assumptions allows the online extraction of visual landmarks.

Chapter 3

Intelligent Wheelchairs

Intelligent wheelchairs can become an important solution to assist physically impaired individuals who find it difficult or impossible to drive regular powered wheelchairs. In this chapter we describe the main concepts and the design of the IntellWheels intelligent wheelchair, and propose a shared control methodology based on the idea that the wheelchair is immersed in a field of potential forces. Due to the lack of realism of the first version of the IntellWheels simulator, we develop a new wheelchair simulator taking advantage of one general robotics simulator. We also propose a hardware design that aims to reduce the visual impact caused by the assemblage of sensor and actuators in the wheelchair. Experimental results demonstrate that the shared control methodology was able to reduce the number of collisions in more than 75%. The assessment of popular robotic simulators indicated that USARSim was the simulator whose features better matched the IntellWheels project requirements, and a new IntellWheels simulator was built using Usarsim as its base. Finally, a public opinion assessment suggested that IntellWheels design was effective to mitigate the visual and ergonomic impacts caused by the addition of its sensorial and processing capabilities.

3.1 Introduction

An intelligent wheelchair (IW) can be defined as a motorized device with a chair, in which an artificial control system augments or replaces the user control in order to assist physically impaired individuals [71]. The main difference from regular powered wheelchairs is that IWs are provided with a sensorial system and processing capabilities to reduce or eliminate the user's task of driving. According to Braga *et al.* [15, 16] and Jia *et al.* [72], the main capabilities of an IW are:

- Extended human-machine interaction through distinct types of devices such as joysticks, touch-sensitive display, voice, facial expressions vision and other sensors based control like pressure sensor;
- Perception of the environment;
- Obstacle avoidance capabilities;
- Autonomous navigation;
- Cooperation with the user (shared control);
- Communication and collaboration with others devices, such as automatic doors, elevators and others wheelchairs.

According to Fehr *et al.* [73], 91% of the clinicians believe that robotic wheelchairs with automated navigation systems can be useful at least for a few users, and 23% believe the systems can be useful for many of them. Another recent study, conducted by Simpson *et al.* [4], estimated that between 61% to 91% of all the wheelchair users would benefit somehow from the features of intelligent wheelchairs. Therefore, investment in research and commercialization of intelligent wheelchair have much greater potential impact than previously thought.

This chapter presents three contributions to the development of intelligent wheelchairs. The first deals with the development of a generic hardware framework, which design concerns with minimizing the visual and ergonomic impacts of the addition of sensing and computational capabilities. The second is the assessment of robotic simulators, which considered seven criteria to compare and select the simulator that better matches the requirements of the IntellWheels project. Finally, the third contribution proposes a shared control methodology that is effective to avoid collisions, and yet simple enough to run in real-time in embedded systems with limited computational capability.

The outline of the chapter is the following. Section 3.2 presents relevant related works in the areas of intelligent wheelchairs and obstacle avoidance. Section 3.3 presents a description of the IntellWheels project. Section 3.6 address the problem of obstacle avoidance. Section 3.7 presents the results of the local obstacle avoidance experiments, as well as the assessment of robotic simulators and of the visual appearance of the IntellWheels prototype. Finally, the summary and conclusions of this chapter are presented in Section 3.8.

3.2 Literature Review

3.2.1 Intelligent Wheelchairs

One of the first projects of an autonomous wheelchair for the physically handicapped was proposed by Madarasz *et al.* [74]. In 1986, they presented a wheelchair equipped with a microcomputer, a digital camera and an ultra-sound sensor. Their objective was to develop a vehicle capable of operating without human intervention in populated environments, with little or no collisions with objects or people. In this project, the camera was used to recognize moving objects, artificial landmarks, previously identified objects (such as the number of the rooms and elevators) as well as drive the wheelchair in the center of the corridors. Ultra-sound sensors were used to determine the relative distances towards objects, and to orient the wheelchair regarding walls and corridors. In some situations, the information from both sensors were combined, such as to verify whether the elevator's door is open or closed.

From 1987, The University of Edinburgh's CALL Centre developed a intelligent wheelchair prototype (CALL Centre Smart Wheelchair) for children with severe and multiple disabilities who could not use ordinary mobility aids [75]. The project performed a qualitative evaluation about the effective use of intelligent wheelchairs as means to increase the mobility, communication and education of the children that used the wheelchair. For validation, twelve prototypes were tested in three schools for children with special needs. The CALL Centre Smart Wheelchair is equipped with bumpers to protect the pilot and the environment from collisions, ultra-sound sensors to reduce the wheelchair speed in the proximity of objects and a track follower to enable the wheelchair to follow lines on the floor, and a computer. Regarding the human-machine interface, the wheelchair can be driven by single or multiple switches, a scanning direction selector, and proportional joystick. In [76] Nisbet emphasized that the focus of the project was to design an intelligent wheelchair to complement the user skills, since a fully autonomous vehicle would provide low therapeutic effects. The wheelchair control is seen as symbiotic partnership between the user and the wheelchair. The CALL Centre Smart Wheelchair is manufactured by Smile Rehab Limited [77] since 2000 and is one of the few intelligent wheelchairs currently available on the market.

Hoyer and Holper [78] first presented the OMNI (Office Wheelchair with High Maneuverability and Navigational Intelligence for People with Severe Handicap) project in 1993. The name of the project comes from the omnidirectional robotic base used to facilitate the navigation in cluttered environments. The wheelchair architecture is modular, composed of different locally intelligent units, which enables a flexible reaction and increases the reliability because of a mutual verification of the transferred data. The control system was divided into low level control (including motion control, sensors module and robotic arm), high level control (trajectory planning and task planning) and interface (voice control, keyboard and joystick) [79]. In its initial architecture the project presented the robotic arm MANUS, and interesting functionalities like autonomous and semi-autonomous navigation, obstacle avoidance, and some specific functions like driving along a wall and driving through a door

NavChair started in 1991 with US$ 330,000.00 dollars in funds and a three year duration. The prototype is based on a commercial powered wheelchair assembled with a computer, ultra-sound sensors, motor controller and joystick [80, 81]. The functionalities developed in the wheelchair are obstacle avoidance, wall following, and doorway navigation. The project implemented and tested a shared control system, in which user and wheelchair share the control of the wheelchair in order to yield safer navigation.

From 1995, Miller and Slack [82] designed Tin Man I prototype based on a low cost powered wheelchair assembled with encoders, bumpers, infra-red sensors, ultra-sound sensors, a digital compass and a microprocessor. Initially, the system had three modes of operation: manual control with obstacle avoidance; autonomous driving through a pre-defined trajectory; and autonomous driving to a specific point (x, y). Later, the project evolved to Tin Man II in order to re-design its user-machine interface, increase its operation speed and reduce its dependency to bumpers. In addition, new navigation capabilities was also designed, such as the ability to store travelled information, return to the starting point, follow walls, navigate through doors and charge the battery autonomously. By including some of Tin Man's capabilities, the Maid project [83] is designed to navigate in two particularly difficult and tiresome situations, respectively narrow cluttered environments and through wide crowded areas. Wellman *et al.* [84] proposed a hybrid wheelchair equipped with two legs, in addition to its regular four wheels - enabling the wheelchair to climb over steps and to move through rough terrain.

The project FRIEND (Functional Robot arm with user-frIENdly interface for Disabled people), developed by the University of Bremen, is a robot for rehabilitation whose main goal is to assist impaired individuals with limited locomotion. In its first version, FRIEND was composed by a powered wheelchair and the robotic manipulator MANUS with six degrees of freedom [85, 86]. Both wheelchair and manipulator were controlled trough a touch-sensitive screen and voice commands. In 2005, the system evolved to a second version (FRIEND II) that extended the hardware [87] and implemented a new multiple layer software architecture [88]. The goal of the new software architecture was to facilitate the interaction with smart devices of a household environment by adapting or generating new sequences of actions. The framework was divided into three parts (hardware, skills and sequential control), in a modular fashion that used the Common Object Request Broker Architecture (CORBA) to communicate between each other.

Smartchair is an intelligent wheelchair designed in the GRASP lab of the University of Pensilvânia. The prototype has one monocular and one omnidirectional camera, one video projector, infra-red sensors, encoders, one laser scanner, one GPS and one embedded processing board. Among the modes of operation, the user can choose to navigate autonomously to a given destination, navigate in the hallway, navigate through doors and manual navigation with obstacle avoidance [89, 90, 91]. The cameras were used both for recognition of artificial landmarks and well as for estimation of the wheelchair orientation in indoor environments. Preliminary experiments were also performed in order to evaluate the use of GPS in outdoor environments,

SENA is currently under development in the University of Málaga, Spain. It is one of the few IW projects that concerns with the communication system between wheelchair modules. The prototype is based on a commercial powered wheelchair equipped with laser scanner, infra-red sensors, sonars, encoders and a CCD camera, and controlled by a computer connected to a microcontroller through a USB connection. SENA architecture was initially based on a 3 layer structure named Architecture for Cognitive Human-Robot Integration (ACHRIN), developed to facilitate the participation of the user in the wheelchair tasks, including deliberation and plan execution. The architecture later evolved to a multi-agent system called MARCA (Multi-Agent-based Robotic Control Architecture), due to some deficiencies of the previous model, like its rigid client-server communication and the lack of mechanisms that allow redundancy. The communication between agents was designed to transmit messages using the FIPA-ACL protocol [92], and a reinforcement learning methodology was incorporated in each agent [93].

RoboChair aims to develop a high performance low cost intelligent wheelchairs to assist elderly and physically impaired people. The project is focused on two levers of complexity: one is an intelligent control system to achieve good control stability and fast image processing capability. Another is a friendly user interface for voice control, emotion and gesture detection, as well as a wireless vision system for carers or relatives to monitor and communicate remotely. The combination of Adaboost Face Detection and Camshift object tracking algorithms resulted in a real-time hands-free wheelchair control through face detection and gesture recognition. RoboChair is also able to identify localization landmarks, follow walls, avoid obstacles and navigate autonomously [72].

ACCoMo (intelligent wheelchair as Autonomous, Cooperative, COllaborative MObile robot) [94] is a prototype of an IW that allows handicapped individuals to move safely in indoor environments. The prototype is based on a powered wheelchair with infra-red sensors, camera, a computer and a touch-screen display. Through its multi-agent system, ACCoMo claims to be able to provide an autonomous navigation with obstacle avoidance, a cooperative behaviour with other wheelchair and a collaborative behaviour with the user. The system intelligence is given by reinforcement learning, neuronal networks and genetic algorithms. The navigation is based on metric maps of indoor environments and localization is performed through Radio Frequency IDentifier (RFID).

The Mobile Internet Connected Assistant (MICA) project from the EISLAB, focused on finding assistive technology solutions to help wheelchair users in their daily life [95]. The prototype is equipped with a computer, a fiber optic rate gyroscope, a laser scanner and two encoders, and spans in the area from navigation techniques to the design aspects of intelligent wheelchairs [96].

The MIT intelligent wheelchair project (Wheelesley) proposes to enhance ordinary powered wheelchairs equipping it with distance sensors to perceive the surroundings, a wireless device for room-level location determination, and motor-control software to effect the wheelchair's motion. However, the main focus of MIT research has been the development of a speech interface to interpret and follow natural language directions [97].

The University of Leuven presented the project Sharioto, which is based on a powered wheelchair equipped with different ultrasonic sensors, infrared sensors, one laser scanner and with a gyroscope. Sharioto proposes shared control that attempts to estimate the user's intent from user's noisy input signal (a joystick) and the interaction with the perceived environment to generate navigational behaviors [98].

The intelligent wheelchair prototype from the University of Shiga [99] is based on a six wheeled powered wheelchair equipped with infra-red sensors and a computer. They propose an obstacle avoidance algorithm based on neural networks to provide aid for people who find it difficult or impossible to drive a conventional wheelchair. For that Shiga's prototype vary the connection weights of the neural network according to the distance to obstacles in the vicinity of the wheelchair, and thus, improve the obstacle avoidance function.

In addition, other important projects present solutions to some common issues faced by individuals with limited mobility, such as the intelligent navigation system discussed in SENARIO [100]; the autonomous and semi-autonomous movements of VAHM [101]; the obstacle avoidance and shared-control system of Rolland [102]; the motion control of Vulcan from the University of Texas Austin [103]. Other projects present alternatives for human-machine interaction, like the interface designed in SIAMO [104], the recognition of facial expressions developed in [105] and [106], and the use of electromagnetic waves of the brain [107, 108].

Currently there are several active international projects. RADHAR [109, 110] that proposes a framework to fuse uncertain information from both environment perception and the driver's steering signals in order to estimate a safer trajectory to the wheelchair. LURCH [111], that aims to extend the user command interface as well as perform autonomous and semi-autonomous navigation. ARTY project [112], which focuses in developing an intelligent pediatric wheelchair and the project from the University of Zaragoza [113, 114] that focus on mobile robot navigation and brain-computer interfaces.

In Portugal four other projects proposed solutions to assist physically impaired individuals. ENIGMA [115] is a robotic omnidirectional base developed at the University of Minho. Recently it is being used for the study of some applications of gestures commands. Magic Wheelchair is a gaze driven IW, which is part of the MagicKey Project from the Polytechnic Institute of Guarda [116]. RobChair is the intelligent wheelchair project developed in the University of Coimbra. The wheelchair is steered with voice commands and assisted by a reactive fuzzy logic controller [117]. The prototype is based on a powered wheelchair with twelve infra-red sensors, four ultrasound sensors, a front bumper and two optical encoders [118, 117]. The prototype aims to assist the locomotion of impaired people providing autonomous and semi-autonomous navigation with obstacle avoidance [119]. Robchair presents a distributed modular architecture interconnected through a CAN bus. The control module follows the classic subsumption architecture proposed by Rodney Brooks [120]. Table 3.1 is an extension of [121], and summarize details of the most relevant intelligent wheelchair projects.

Table 3.1: A simple longtable example

Project	RGB Camera	Sonar	Infrared	Bumper	Encoder	2D Laser Scanner	Compass	Line Tracker	RFID	Accelerometer	Inclinometer	Giroscope	IMU	GPS	RGB-D Camera
ACCoMo [94]	✓		✓						✓						
Arty [112]					✓	✓									
ASU [74]	✓	✓			✓										
A.G.W. [122]			✓					✓							
CALL Centre [75]		✓		✓				✓							
CHARHM [123]	✓														
COACH [124]		✓	✓												
CPWNS [125]	✓				✓										
CUHK [126]		✓													
CWA [127]					✓										
Enigma [115]															
FRIEND [88]	✓				✓										
Hephaestus [128]		✓		✓											
INCH [129]	✓				✓										
INRO [130]		✓			✓	✓					✓			✓	
I.W.S [131]	✓		✓		✓										
IntellWheels [16]	✓	✓			✓	✓									✓
KU [132]					✓										
LOUSON III [133]	✓	✓			✓		✓								
LURCH [111]	✓				✓	✓									
Magic [116]		✓			✓										
MAid [83]		✓	✓		✓	✓						✓			
Mister Ed [134]		✓	✓	✓	✓										
Mr. HURI [135]	✓	✓													
NavChair [80]		✓			✓										
NLPR [136]	✓	✓			✓										
OMNI [78]		✓	✓	✓	✓										
Orpheus [137]		✓			✓										
Phaeton [138]	✓	✓		✓	✓										
RADHAR [110]	✓				✓	✓							✓		✓
RobChair [118]		✓	✓	✓											

Continued on next page

Table 3.1 – *Continued from previous page*

Project	RGB Camera	Sonar	Infrared	Bumper	Encoder	2D Laser Scanner	Compass	Line Tracker	RFID	Accelerometer	Inclinometer	Giroscope	IMU	GPS	RGB-D Camera
RoboChair [72]	✓	✓													
Robotic Wheelc. [139]	✓	✓			✓										
Rolland [102]	✓	✓	✓	✓	✓										
SENA [140]	✓		✓			✓									
SENARIO [100]		✓			✓										
Siamo [104]	✓	✓	✓	✓	✓	✓									
SIRIUS [141]		✓			✓										
Smart Alec [142]		✓			✓										
SmartChair [90]	✓		✓		✓	✓									
SPAM [143]		✓	✓	✓	✓										
SWCS [144]		✓	✓	✓											
TAO [145]	✓		✓	✓											
TetraNauta [146]	✓	✓	✓	✓											
The Wheelchair [147]	✓	✓	✓												
Tin Man II [82]		✓	✓	✓	✓		✓								
TUT [148]		✓	✓	✓											
UNIVPM [149]		✓			✓										
UOP [150]		✓													
UP [151]	✓	✓			✓							✓			
VAHM P2 [101]		✓	✓	✓	✓										
V-c-A [152]			✓		✓										
Vulcan [103]	✓	✓	✓		✓										
WAD [153]			✓	✓	✓										
Watson [154]	✓					✓									
Wheelesley [155]	✓	✓	✓	✓	✓										

3.2.2 Obstacle Avoidance Methodologies

Considered one of the first methodologies proposed to avoid obstacles during robot navigation, edge-detection became popular in late eighties. The algorithm starts with the robot in a stationary state. Through ultrasonic-sensor readings, it performs a panoramic scan of the environment. With the distance measures, an edge-detection algorithm tries to map the position of the vertical edges of the obstacles in the robot surroundings. Once new edges are found, a temporary map is updated and an optimum path planning algorithm is applied to plan the robots subsequent path [156]. The robot moves through the path while taking alternate measures of its ultra-sound sensors[1]. When an obstacle is detected under a certain safety distance, the robot stops and restarts the cycle all over again [157, 158]. Edge-detection methodology is not itself an obstacle avoidance technique. Actually, it can be better described as an approach to represent the environment based on geometrical primitive line segments. Therefore, off-line path planners are still needed in order to yield obstacle-free paths, limiting its implementation in low-resource embedded systems.

Certainty grid (CG) method is a probabilistic representation of obstacles in a grid based world model. This world model has been developed for mobile robots in Stanford and CMU for more than ten years, and was originally designed to handle sonar's inaccuracies shortcomings [159]. In this method, the robot's work area is modeled as a 2-D array of square elements, called cells. Each cell of the grid contains a likelihood estimate (certainty value) that indicates confidence that an obstacle is placed within the corresponding region of space. Once readings are more likely to detect objects closer to the acoustic axis of the sonar, a probabilistic function updates more the certainty value in this region than in the other areas enclosed by the sensor [159, 160]. In spite of some improvements presented by CG methodology, some drawbacks can compromise its implementation in real-time applications. Firstly, the accuracy provided is too much dependent of the cell size. Secondly, as the robot moves over large areas, lots of memory and processing power are required, restricting the application of CG especially in some embedded systems. Finally, the subsequent robot's path is computed by a global path-planning, usually off-line.

Introduced by Borenstein and Korem [156], the Vector Field Histogram (VFH) uses a polar histogram instead of a 2-D Cartesian grid to avoid collisions and steer the mobile robot to the target. This method employs a two-stage data reduction process in order to compute the control command to the robot. In the highest level of data, VFH stores a detailed 2-D histogram grid map of the robot's neighborhood. As just only one cell in the histogram is updated for each range reading, it takes just a small computational overhead. Thus, a probabilistic distribution is obtained by continuously and quickly sampling each sensor while the robot moves [161]. At the second level, data is mapped onto a one-dimensional polar histogram that comprises n angular sections each with width α. Each sector in the polar histogram contains a value representing the polar obstacle density in that direction. Finally, based on the obstacle polar density (1-D histogram),

[1] To avoid the interference of the sound waves of one sonar into another, one alternative is not to range all ultra-sound sensors at the same time. Since each range takes around 80ms, a full panoramic scan of the environment is only performed with the robot stopped

VFH selects the best steering direction for the robot and computes the reference values for driving the robot (third level of data representation) [156].

As can be observed, the VFH overcomes some issues shown by the other methods described above. In fact, the influence of low accuracy distance measures is minimized through the histogram representation. In addition, the world representation is restricted to the robot's surrounding trough a bi-dimensional sliding window, reducing the computational overhead. On the other hand, local minima problems are still not solved by the algorithm itself, which has to invoke a global path planner when these situations are flagged. Finally, like edge-detection and CG methodologies, VFH depends not only from the data gathered by the sonars, but from an accurate localization system. Otherwise, inaccurate robot's position can introduce more errors and disturb the object mapping.

First suggested by Andrews and Hogan [162] and Khatib [163], the Potential Fields methodologies (PF) relies on a simple and powerful principle, the artificial potential field concept. In this method, the robot is considered immersed in a potential field generated by the target and by obstacles. In this field, obstacles generate imaginary repulsive forces, while the target generates an attractive force to the robot. The resultant robot behavior is obtained by the sum of all attractive and repulsive forces at a robot's given position.

After the original work, a number of improvements and extensions have been published. Krogh [164] has computed forces not only to steer the robot around objects, but to set its speed as well and Seiki [165] has introduced the consideration about the nonholonomic motion constrains and the robots shape into the PF. Khatib and Chatila [166], considered, besides distance, the robot's relative orientation to the obstacle in order to compute forces. Bicho [167] implemented a dynamic approach using low level sensory information, in which each sensor generates a repulsive force that drives the direction and the speed of the robot.

In its original version the PF methodology exhibit many shortcomings, in particular the sensitivity to local minima that arises mostly due to the symmetry of the environment. Furthermore, it tends to be very susceptible to misreading (since it takes into account just one set of data) and to the sonar most common issues. Some versions still assume a known and prescribed world model to evaluate off-line the potential field. Finally, some implementations present significant problems related to oscillations in narrow passages and in the presence of obstacles [156, 168]. The potential field concept is a specially interesting approach because it can be easily adapted to avoid the map-based obstacle representation, and thus to run on embedded systems with limited computational capability.

3.3 Overview of the IntellWheels Project

Despite the several projects under development, there is not a generic model for transform a regular powered wheelchair into an intelligent wheelchair. Usually, these projects have hardware and software architectures specific to the model of the wheelchair used in the project, are cost prohibitive for most potential users and typically requires a very difficult configuration.

In an attempt to address some of these issues, the Faculty of Engineering of the University of Porto (FEUP) in collaboration with the Artificial Intelligence and Computer Science Laboratory (LIACC), the INESC Technology and Science associated Laboratory (INESC TEC), the Institute of Electronics and Telematics Engineering of Aveiro (IEETA), the School of Allied Health Sciences of the Polytechnic Institute of Porto (ESTSP), the University of Minho (UMINHO) and the Portuguese Association of Cerebral Palsy (APPC) developed the project IntellWheels. The focus of the project is to develop an intelligent wheelchair with flexible multimodal interface that facilitates the development and test of new methodologies and techniques, and whose integration into commercially available powered wheelchairs may be performed with only minor modifications [16].

IntellWheels is modeled using the multi-agent system (MAS) architecture in order to facilitate the integration of new abilities to the wheelchair. Another advantage of such an approach is that agents can show self-organization behavior, which can emerge through simple individual strategies. The IntellWheels MAS architecture was designed to follow the standards of the Foundations of Intelligent Physical Agents (FIPA) [92] in order to promote the interoperation of heterogeneous agents and the services that they can represent [169]. The MAS is better explained when subdivided into macro and micro perspectives. In a micro perspective, each wheelchair is composed of several micro agents (i.e. intelligence, control, interface, etc.). In a macro perspective, each wheelchair is represented by a single macro agent, just as other agents present in the system (i.e. door agent, logger agent, etc.). Figure 3.1 depicts IntellWheels software architecture, with the wheelchair agents modeled in the platform. The tasks performed by each of the micro agent that comprise an intelligent wheelchair macro agent are briefly described below:

- Cognitive Agent: responsible for the Strategic layer. This agent defines the IW global goal and the sequence of intermediate high level objectives required to fulfill the global task. Once intermediate objectives are defined, the agent can then generate a plan with the sequences of basic actions.

- Control Agent: responsible to provide the wheelchair with the Tactical control (middle level control). The middle level performs the control of basic actions, like follow line, spin, follow wall, goto XY. In addition, it also computes the reference speed (wheelchair's linear and angular speeds) and communicate with the wheelchair.

- Interface Agent: The Interface Agent is responsible for collecting user inputs (through the Multimodal Interface module), and to display the most relevant information (e.g. sensor readings, speed, position) through a graphical user interface (GUI). In addition, it is also responsible for making the interaction between the user and the other agents of the system.

- Perception Agent: This agent represents the perception system of mobile robots. Its tasks are read the sensors, update the world representation, perform the wheelchair localization and map the environment.

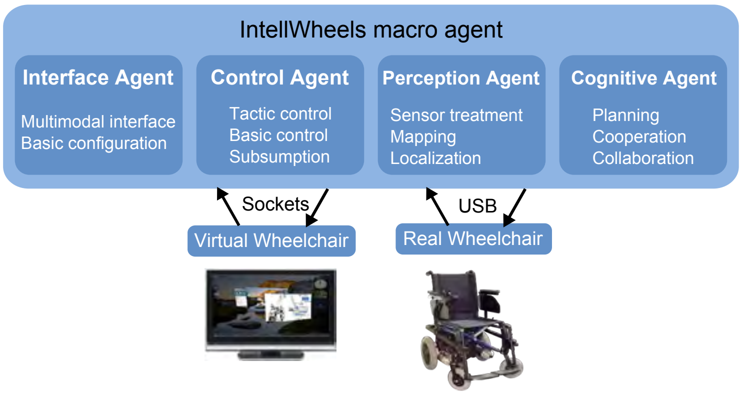


Figure 3.1: IntellWheels Software Architecture.

Other agents, designated as Services Agents, were created to assist the IW system to achieve its global goals. Services Agents can cooperate and collaborate with the agents embedded in the mobile robot. The Door Agent is responsible for controlling the doors and gates in the IW environment, opening and closing doors to allow or inhibit access in restricted areas. The Logger Agent is responsible for creating permanent log files about the messages exchanged between agents, in order to assist the debugging process and system analysis. The Wheelchair Actions Watcher Agent is responsible for centralizing the control of all traffic in the IW environment, thus avoiding traffic conflicts. The role of this agent is to monitor all activities and actions when necessary so as to avoid potential conflicts and to solve possible deadlocks. The Assistant Agent is responsible for system-wide human interaction, as well as for receiving and handling global goals. This agent is the interface between nurses, doctors, therapists and assistants with the IW system.

In this system, an IW can assume bodily form in three different modes, real, virtual and mixed reality. To instantiate the body of the wheelchair, it is necessary to use the hardware for the real robot, the simulator for the virtual robot or both for the mixed reality. In face of that, one of the most innovative features of the platform is that it allows interactions between real and virtual IWs. These interactions make high complexity tests possible, with a substantial number of objects, devices and other wheelchairs. Furthermore, it implies a large reduction in project costs, once it is not necessary to build a large number of real IW to perform interaction tests [19].

3.3.1 User Inputs

The project has a deep concern in providing assistance to individuals with distinct impairments. Therefore, eight types of user inputs are currently supported. By providing a broader range of inputs, we aim to give to the user the capability to choose the most comfortable and suitable way to drive the wheelchair. In addition to the traditional joystick, the user has the option to use a game joystick, which has many configurable buttons that can be customized to trigger high level actions

[17, 18]. Through the keyboard and touch-screen display, users and technical staff can setup the IW parameters.

Another form of interaction includes the head gestures input, which allows elderly and disabled people to steer the IW according to the position of their head. The head gestures input is a device designed in the framework of the IntellWheels project, and consists of a cap fitted with a 3-axis accelerometer/inclinometer that communicates via Bluetooth with the computer [15, 170]. Through the multimodal interface, the wheelchair user can choose to use the cap to trigger high level commands, or use it as a proportional controller where the position of his head is translated into the wheelchair linear and angular speed. The Facial expressions input allows the wheelchair to recognize some simple facial expressions, which can be associated with middle (i.e. go forward, turn right, turn left) and high level actions (i.e. go to the dining-hall, go to the bedroom) [171, 172]. IntellWheels also support voice control by capturing spoken commands through a microphone, converting the speech into text (through the Windows speech API) and triggering an action. Finally, the integration of a commercial brain computer interface that recognizes facial expressions and thoughts is been tested, but due to its low accuracy it is still very difficult to use this device to enable safe and robust commands to the intelligent wheelchair [173]. An evaluation of the distinct input methods available to control the wheelchair is presented in [174].

3.3.2 Navigation System

The Navigation System is responsible for performing the wheelchair's sensors treatment, localization and driving the wheelchair between different locations. The user control module is the application in which the user defines the type and parameters that the controller will use for automatic mode. After choosing one of seven types of actions (following the line, point, the angle, following the left wall, the right wall, wait, stop) several parameters and configuration fields become available to the user. There is also the possibility of creating a sequence of actions (each one with its individual configuration and objective). Once the objective of the current action is completed, this action is deleted and the next action is performed [175, 176].

The localization system of IntellWheels is based on dead reckoning techniques [19, 15]. It estimates the state of the robot at the current time-step k, given previous knowledge about the initial state and all measurements up to the current time. Typically, a three-dimensional state vector $p = [x; y; \theta]^T$ is used to represent the position and orientation of the robot. The position estimation of a robot based on sensor data is one of the fundamental problems of mobile robotics. The probabilistic robotics paradigm was used in ours odometry motion model. This paradigm pays tribute to the inherent uncertainty in robot perception, relying on explicit representations of uncertainty when determining what to do. Viewed probabilistically, perception is a statistical state estimation problem, where information deduced from sensor data is represented by probability distributions [177]. The wheelchair has a typical differential drive configuration, where the two incremental encoders are mounted to count the wheel revolutions. Relating the pulse increment with the sampling interval and the nominal wheel diameter it is possible to express the wheelchair displacement and rotation in the Cartesian frame [6]. Through the Odometry Motion Model presented in [178],

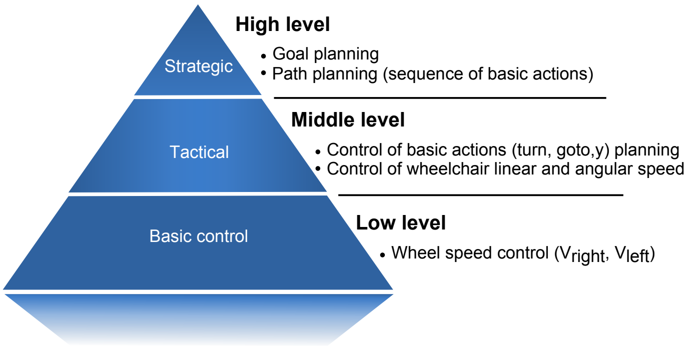

High level
Strategic
• Goal planning
• Path planning (sequence of basic actions)
Middle level
Tactical
• Control of basic actions (turn, goto,y) planning
• Control of wheelchair linear and angular speed
Low level
Basic control
• Wheel speed control (V_{right}, V_{left})

Figure 3.2: IntellWheels Multi-level Control Architecture.

the computed uncertainty was used to estimate odometry errors and plan the wheelchair path to cross artificial landmarks.

Figure 3.2 depicts IntellWheels multi-level control architecture, subdivided into three layers: strategic layer (goal planning and path planning), tactical layer (control of basic actions, and linear and angular speeds) and basic control layer (control of wheel speeds) [179]. A goal planner was implemented with Planning Domain Definition Language - PDDL [180]. The planning graph is a powerful data structure that encodes information about which states may be reachable, and provide a sequence the intermediate objectives required to achieve the global goal (high level action given by the user). The system than generate a path in order to achieve the objectives proposed by the planner, taking into account information from the world model. To find a path from a given initial point to a given goal point, the system has an adapted A* Algorithm implemented [176]. Later, the tactical layer of the control module subdivide the path into basic forms (lines, circles, points), and computing the wheelchair's linear and angular speeds to put the wheelchair into motion [176]. Finally, the lowest level of control (Basic Control Layer) converts linear and angular speeds into wheel speeds send them through serial communication to the interface board.

3.3.3 Communication System

Safe communications in open transmission systems, safe navigation, obstacle avoidance, and others, are some constraints applicable to mobile robots and IWs. With the proliferation of Wi-Fi technologies and devices, the current way in which communications occur is evolving. While these new technologies present advantages, they also have some disadvantages, specifically in the field of safety-related systems or safety-critical systems (a system that in the event of a failure can damage individuals, properties or the environment) [181].

If a mobile robot is a safety-related system or part of one, the communication system must prevent failures and prove to be safe for unauthorized access, while maintaining the desired level of compatibility with the system's available physical media transmission layers. To address and solve these issues, the standard EN 50159-2 [182] must be followed. It describes the known threats to communications and their defensive methods applicable for safety critical systems that use open transmission media layers.

Usually, a multi-agent platform as the Java Agent DEvelopment Framework (JADE) [183] would be used to enable communications and organize the different agents. However, with common multi-agent platforms, it is not possible to customize and enhance its functionality to better adapt the system to safety-critical problems. The solution to this problem, in this project, was to develop new methods in a new multi-agent platform.

The IntellWheels communication system was implemented in Object-Pascal, following the FIPA guidelines for the ACL [92], and a set of services, such as an Agent Management System, a Message Transport System and a Directory Facilitator. The system's architecture was designed as five separate layers, with their respective receiving and sending handling methods, and interfaces running in parallel. This way, it becomes possible for the user to choose which layers should be applied to the application, without compromising the agent's functionality, while following the OSI Reference Model and implementing fault tolerant methods.

The Communications layer is responsible for receiving and sending messages from and to the message transport layer. This layer prevents the interpretation of repeated messages, present in the physical media, and enables the retransmission of messages, thus preventing packet loss at the network level. It also prevents the application from receiving messages with a size that is larger than the one specified by the user during agent implementation. The Security layer is responsible for the message's security, preventing the interception and modification of messages. The Encryption method is chosen according to the message's destination and the platform knowledge at that moment. The possible encryption methods involve the use of a private and public key pair or an AES pre-shared key. It also performs message integrity checking by cross-referencing the message with the transmitted message's hash. The Temporal layer is responsible for adding time restrictions to the messages. These restrictions can be seen as a defensive measure. By adding a timestamp to the message's data, it would be possible to filtrate outdated messages. Finally, the Parser layer is responsible for the construction of the message according to the FIPA-ACL standard and represented using the normative constant FIPA-SL. It also selects the messages that are accepted by the application according to their correct structure configuration and to the sender's presence in the platform, thus stopping any communication from an unauthenticated application.

Crucial to this architecture is the election of a Container entity, similar to JADE, and the distribution of a Local Agents List (LAL), as well as a Global Agent List (GAL), using a message-oriented paradigm. These lists contain the application's configurations that enable communications and distribution of the public encryption key between agents. The Container was designed to be responsible for the lists maintenance operations that include creation, update and deletion. However, and contrary to other systems, the Container was not designed as a separate entity or as the base for agent's creation and their activity. The idea behind this is that it is admissible and probable for a wheelchair to lose network connectivity or to change its network configuration, but it is not acceptable for these changes to cause a system malfunction.

3.3.4 User Interface

An interface is an element that establishes boundaries between two entities. Currently, most traditional human-machine interfaces are based on a single and not customizable input/output correlation. An evolution to this paradigm and a way to create a more natural interaction with the user is to establish a multimodal interaction, which contemplates a broader range of modes and channels of communication, such as video, voice, pen, etc. According Oviatt [184], a Multimodal Interface (MMI) aims to naturally recognize occurring forms of human languages, and incorporate one or more recognition-based technologies (i.e. speech, pen, touch, manual gestures, gaze, body movements, etc.).

Since the physical disability is very wide and specific to each individual, it is important to provide the largest possible number of input methods in order to try to cover the largest possible number of individuals with different characteristics. Therefore, IntellWheels MMI is designed to allow the simultaneous connection of several input devices, and assist users with a wider range of symptoms and physical capabilities [185]. Since this is a system that aims to be used by disabled people, safety is of extreme importance. In order to avoid the potential accidents caused by the false recognition of user commands, the proposed methodology allows the user to define sequences of inputs, which are subjected to a reliability test [186]. Such sequences are composed of inputs from the same input device (homogeneous inputs), as well as inputs from different input devices (heterogeneous inputs). Thus, users can define the most suitable input sequences taking into account their limitations [170]. Each input sequence can than be associated to one of the actions that the wheelchair is capable to perform. The unique combination between the heterogeneous input sequences and their flexible association with wheelchair actions provide the user the capability to create its own communication language with the wheelchair.

3.4 Hardware Framework Design

Powered wheelchairs are typically composed by a metal frame with four wheels and a seat, batteries, two motors, one motor controller and joysticks. Such configuration is adequate to act in the environment with the constant supervision of a human operator, however, it does not allow the wheelchair to perform higher level tasks. To be considered minimally intelligent, a wheelchair needs to sense its surroundings and react according to changes in the environment, user commands and goals. Therefore, the standard wheelchair configuration needs to be complemented with additional sensors, control electronics and computational hardware. In IntellWheels, this additional set of metal frames and electronic devices are referred to as hardware framework.

When designing the hardware architecture, most projects of intelligent wheelchair concerns, in fact, with solutions to robotics problems. Such solution-centered designs tend to disconsider the typical wheelchair users and their limitation. These wheelchair indeed present a desired feature, or perform better in some situations, but may also create inoperable sophisticated wheeled devices (at least for individuals with limited mobility). It is not hard to find designs that assemble laser

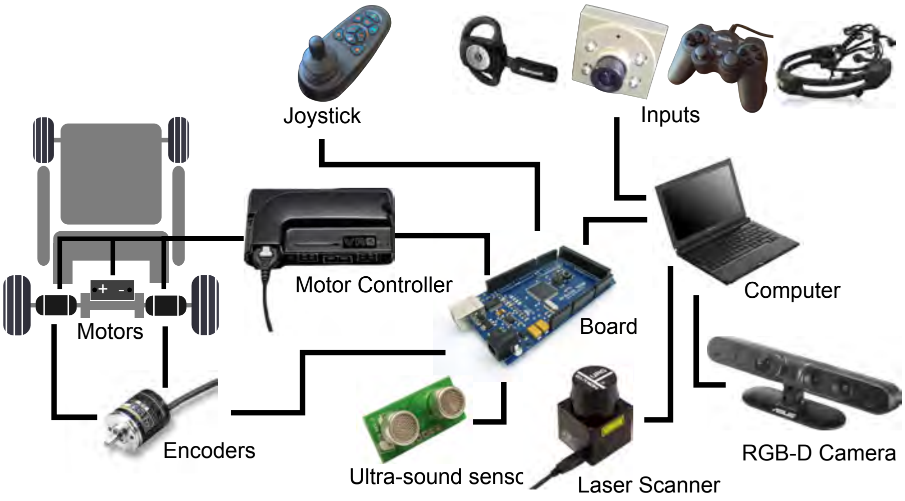


Figure 3.3: Architecture of the IntellWheels hardware framework.

scanners in the region between the user legs, bumpers close their feet, sonar rings in from of seat and a pole over the head.

In this thesis we propose a user-centered hardware design, in which the needs and limitations of physically impaired users are given attention. Impaired individuals spends a significant part of their life on their wheelchair, thus user comfort is regarded as a main priority. Since the addition of any element to the wheelchair may become a nuisance, the proposed design avoided bulky and heavy sensors, and paid special attention to place all components out of the user workspace. Only solutions that does not interfere with the normal wheelchair operation are implemented. The proposed design also seeks to reduce the visual impact of the hardware framework, and maintain its compatibility with multiple wheelchair brands and models. An intelligent wheelchair system that requires substantial modification may be impractical for installation in many of the wheelchair models currently available on the market, interfere with normal service of the wheelchair, and prevent potential users from obtaining wheelchairs that could provide mobility assistance. For this reason, we propose a modular system that can be added to a variety of commercial power wheelchairs with minimal modifications.

The architecture of the framework and its connections with the original wheelchair system are depicted in the Figure 3.3. A computational unit (Intel core 2, 1.2GHz, 2GB RAM) runs the multi-agent system that controls the intelligent wheelchair. A laser scanner, with a field of view (FOV) of 270^{o}, provides to the computer unit distance measures with high accuracy, which in the future can be used for mapping and localization. Two encoders coupled directly in the motor shaft are connected to the interface board and provide information about the wheel revolutions, that in turn are used to estimate the wheelchair displacement and relative localization. Sixteen ultrasound transducers, with a field of view of 45^{o}, are connected to the interface board and provide raw distance information that is used by the obstacle avoidance algorithm to prevent collisions. A RGB-D camera provides a wide 3D view of the environment with a horizontal FOV of 58^{o},

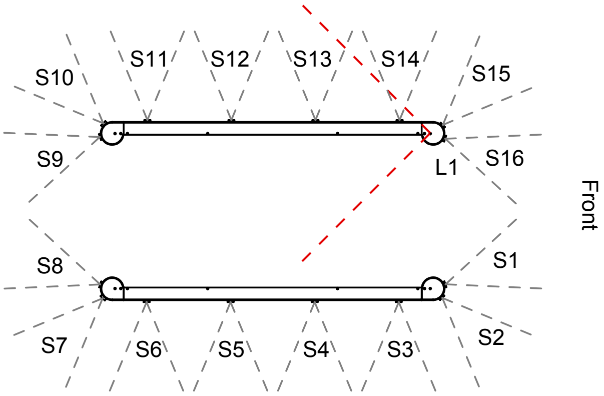


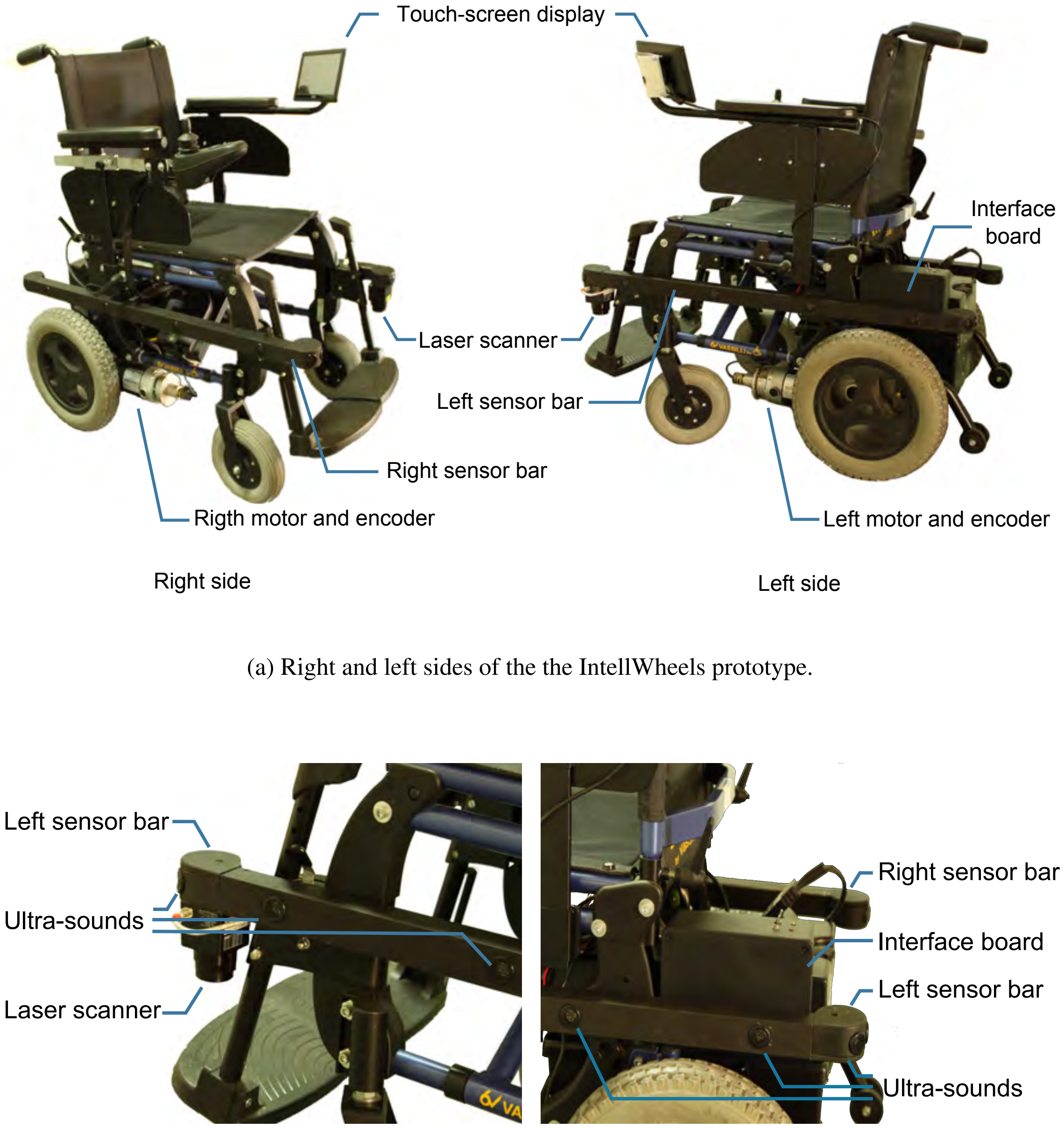
Figure 3.4: Placement and field of view of the distance sensors in the sensor bars. This configuration allows a safety perimeter that extends from 27cm until 80cm.

vertical field of view of 45^{o} and distance ranges from 0.5 to 3.5 meters. An interface board process information from the encoders and the ultra-sound sensors and send them to the computer unit. In addition, it is also capable of providing a reactive obstacle avoidance behavior, that will be detailed in the Section 3.6. The interface board also receives the reference speed of each wheel from the computer unit and generates the corresponding analogical signal that is sent to the motor controller. Power for the sensors and interface board is drawn from the wheelchair batteries though a voltage regulator.

As shown in Figure 3.3, the framework is "inserted" into a power wheelchair control system between the user's input device and the wheelchair motor controller. Since most of wheelchair motor control uses a proprieatry version of CAN bus, intercepting the joystick signal requires opening the joystick module, reading the wires that carry the joystick signal, and altering the signal to those wires. To avoid the need to open the joystick module, it would be necessary to have acess to the bus protocol, or use specific motor controllers that accept signals from external devices.

Normally, the input device is plugged directly into the motor controller. When the framework is installed, however, both the input device and the motor controller are connected to the interface board. The interface board reads the signal from the input device and sends a revised signal to the wheelchair motor controller. The motor controller then treats the revised signal as if it came directly from the input device. Under normal circumstances in which the user operates the wheelchair manually, the revised joystick signal is identical to the original signal. But if an obstacle is detected, the collision avoidance algorithm alters the joystick signal to avoid collisions.

Two lateral sensor bars hold the ultra-sound transducers, the laser scanner, wires and a plastic box containing the interface board. Thus, the IntellWheels hardware framework can be easily attached to standard power wheelchairs from several different manufacturers to convert them into intelligent wheelchairs. Sensors bars are made of aluminium, which provides a good compromise between weight and robustness. Its black color makes the set more discreet, and is consistent with original lines of the wheelchair. The design in two separate bars yields both visual and



(a) Right and left sides of the the IntellWheels prototype.

(b) Close-up view of the front of the left sensor bar. In detail a 2D laser scanner (FOV of 270°) and ultra-sound sensors (FOV of 30°).

(c) Close-up view of the back side of the left sensor bar. In detail the plastic box containing the interface board.

Figure 3.5: IntellWheels prototype. A special attention was given to the design of the sensor bars to minimize its interference with the normal service of the wheelchair. Note that because the two sensor bars are not physically connected, the framework does not interfere in normal operations required for transportation purposes (like battery removal and wheelchair folding). Objects located in the front and in the back of the wheelchair are detected by the ultra-sound sensors assembled in the rounded tip that fits in both extremities of the sensor bars.

operational advantages. The assemblage of the sensors bars do not interfere with the normal battery removal or wheelchair folding, operations usually required to facilitate the transportation of several wheelchair models.

The general configuration of sensors is shown in Figure 3.4. Lateral ultra-sound sensors (S3-S6 and S11-S14) are located 22 cm apart from each other and were assembled directly in the sensor bar. Front and rear ultra-sound sensors (S1, S2, S7, S8, S9, S10, S15 and S16) were assembled in a a special rounded tip designed to fit in both extremities of the sensor bars, and are headed with a 45^{o} difference. This configuration allows a safety perimeter that extends from 27cm until 80cm, in which surrounding objects are in the wheelchair field of view. The figure also depicts the positioning of the laser scanner (L1) in the left sensor bar. Note that Figure 3.4 represents only the position and field of view of the sensors, and not their detection range.

Figure 3.5 identifies the location of the components of the hardware framework in the Intell-Wheels prototype. In addition to the ultra-sound transducers, the left sensor bar also supports the laser scanner (Figure 3.5b) and the plastic container that holds the interface board (Figure 3.5c).

3.5 Simulation

Up to a recent past the use of simulations for simulating IWs (as any robot in general) was quite restricted due to the lack of general simulators. Usually, the existing simulators were developed to deal with some quite specific situations and environments. The development of a new tool for the simulation of IWs is time and resource consuming, and frequently is out of the project's scope. However, this reality started changing due to the release of general simulators.

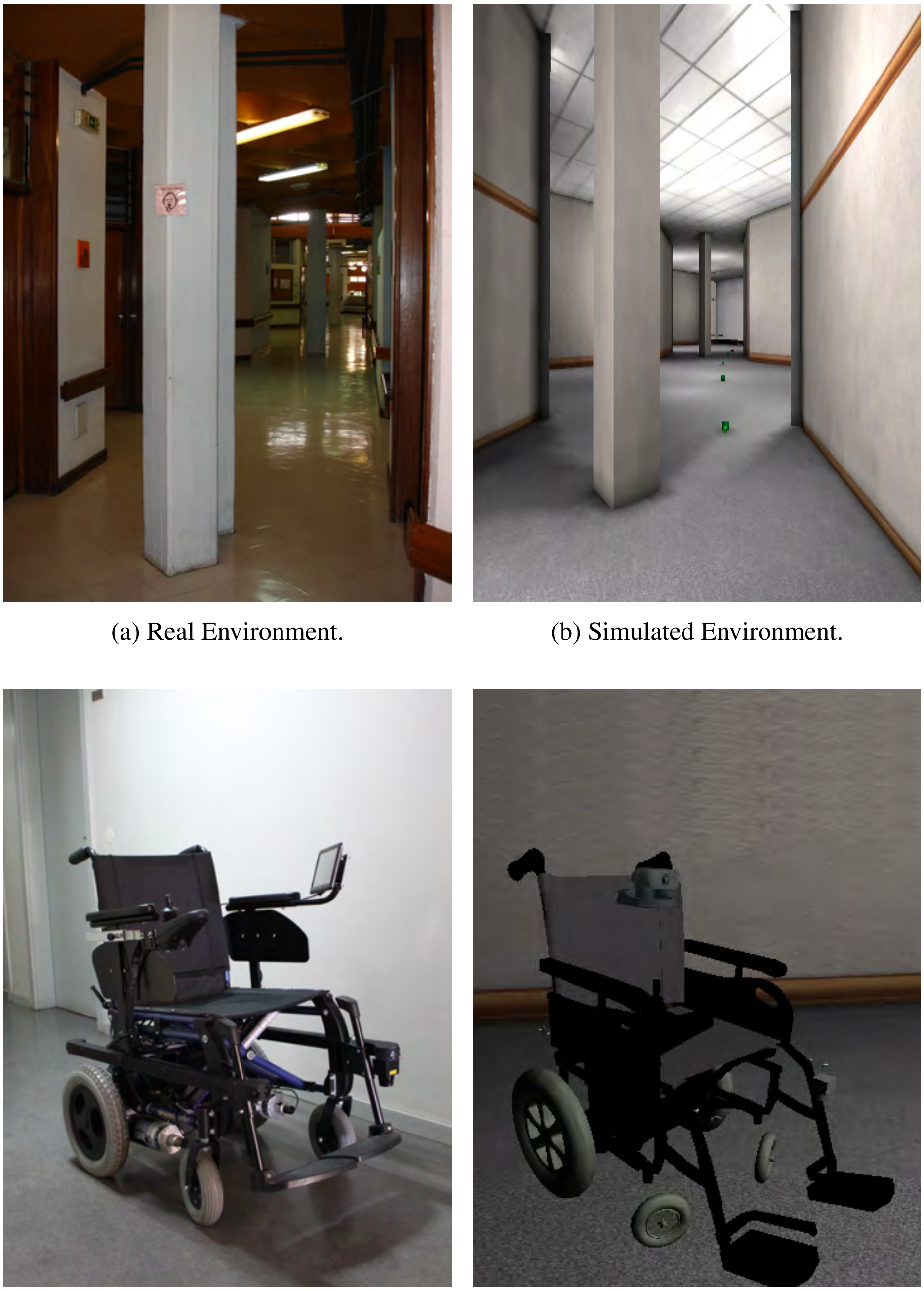
Simulations have a great potential for low cost analysis, since it is able to give researchers access to cost-prohibitive sensors and robotic platforms. In addition, simulators provide the ability to compress time, and so, to evaluate the results of time-consuming experiments much faster. They are pedagogically proven technique for training [187], so they can be used to drill people in safe environments. They allow testing under repeatable and controllable conditions, simplifying debugging (e.g. the same scenario can be precisely generated to trigger a known error).

Unlike real testing environments, which may not be accessible, or may only be accessible at certain times, simulated environments have unlimited availability [188]. For example, experiments that require special natural illumination (i.e. sun light) may be accessible for just some hours a day, and experiments requiring special weather conditions (like fog, rain, etc.) may be accessible just a few times a year. Simulations also provide researchers virtual access to different testing environments, making these virtual testing very cost effective. Actually, with the right modelling, the behavior of the robot can be tested in any environment (from the reconstruction of a laboratory up to urban environments, desserts, catastrophes, lakes, oceans, others planets, etc). Finally, the extensive use of simulators allows researchers to safely refine their algorithms before testing the robot behavior in real environments.

The simulator's involvement in the IW project is even greater as the notion of mixed reality (MR) is introduced (e.g. real wheelchair agents, virtual wheelchair agents, virtual door agents, viewer agents, medical agents). Such types of interactions, between the real and virtual worlds, create a mixed reality environment. The MR support stretches the IntellWheels simulator's capabilities beyond merely testing algorithms. Thus, it is possible to evaluate the reaction of a real IW in a more dynamic scenario - with moving obstacles, complex maps and other intelligent agents moving around. In other words, a real IW connected to the simulator is capable of interacting with virtual objects. The perception agent uses the data gathered from the real encoders to compute the wheelchair's position and then send it to the simulation server. Once the data is received, the simulator places the IW virtual body onto its respective position and returns the perception of the virtual proximity sensor's perception to the real wheelchair agent. Next, the real wheelchair agent combines the data from real and virtual proximity sensors, computes the motor power and sends it to the real wheelchair.

The first version of the IntellWheels simulator was a customization of the Cyber-Mouse simulator [189]. The "Cyber-Mouse" simulator presented several useful characteristics for IW simulation, such as the simulation of different environments, differential robots with two wheels and some sensors (e.g. compass and proximity sensors, GPS). In addition to the simulation server, Cyber-Mouse also contains a 2D simulation viewer specific to the "Cyber-Mouse" competition [190]. IntellWheels simulation module preserved the Cyber-Mouse conceptual architecture, but applied significant adjustments to the robot model and to the collision detection polices [191], as well as in the addition of a 3D visualization module [192, 16].

The first experiments, though, demonstrated that the Cyber-Mouse based simulator lacked the capacity to perform realistic simulations, which in turn are required for testing the intelligent wheelchair subsystems and training users. In fact, simulation results just reflect the reality when the simulation requirements are considered and when the appropriate models are introduced. The requirements for simulating mobile robots may differ according to the purpose of the simulation. For testing motion control, a higher level of detail in multi-body may be important. On the other hand, for testing sensor data processing, a higher fidelity in the sensors measures is desirable. If the simulation aims to evaluate higher level of abstractions, like global localization, ground truth data should be provided. If machine vision is used by the robot, a good rendering is required. Essentially, simulation requirements can be classified into physical fidelity and functional fidelity. The first concerns with how the simulation looks, sounds and feels. In other words, it is the ability of the simulator to render high resolution textures, shades, lighting and reflection. The second concerns with the simulation of most of the forces acting on robots and on its actuators, including not only but gravity, dragging, accelerations and collisions [193]. Carpin *et al.* [194] claim that the simulation of robotic platforms should not consider a robot as an isolated entity, but as an entity which interact and is affected by the environment where it is situated. With that in mind, a detailed assessment of six 3D robotic simulators was performed: Unified System for Automation and Robot Simulation (USARSim) [195], Microsoft Robotics Developer Studio (RDS) [196], Webots [197], SimTwo [198], V-REP [199] and Gazebo [200]. The results of such analysis are presented in the Section 3.7.2. Due to its unique combination of features, USARSim was selected as the basis to the new IntellWheels simulator: IntellSim [21].



(a) Real Environment.

(b) Simulated Environment.

(c) Real Wheelchair.

(d) Simulated Wheelchair.

Figure 3.6: IntellSim: the new IntellWheels simulator. Increased realism with the simulation of textures, lightings, shadows and physics. (a) and (c) depict the real environment and wheelchair, while (b) and (d) their simulated counterpart.

As described by Carpin *et al.* [194], "USARsim is a general-purpose multi-robot simulator that can be extended to model arbitrary application scenarios". It was designed to create physically accurate simulations of robots for research in fields like the human-robot interaction and multi-robot coordination. The simulator is built upon a commercial game engine thanks to the architecture of the Unreal Tournament 3 [201], which separates the game logic and rules from simulation dynamics and environmental data. This way the game core code was reused and applied to a more comprehensive simulation, providing USARsim with high realistic visual rendering and high performance physics simulation. A further advantage relies in the fact that every improvement driven by the gaming industry translates directly into simulation advantages, which is particularly true for hyper realistic rendering and physical simulation [202].

The simulator is open source under the GPL licensing, and platform independent, running under operating systems like Windows, Linux and MacOS. USARSim is highly configurable and extensible, allowing users to develop new sensors, to model new robots and to create and re-create virtually any desired environment. As a consequence, USARSim has become quite widespread within the scientific community, which has released a number of improvements. Simultaneously, researchers have published several papers with quantitative evaluations that demonstrate a very close similarity between the real and simulated systems[195].

To create a realistic environment a model of the APPC building, similar to the place where the patients are used to move around, was built using the Unreal Editor [203]. Several components in the map were modeled using 3DStudioMax [204] and imported into USARSim. Figure 3.6 presents a picture of the APPC building (Figure 3.6a), and its simulated model (Figure 3.6b). The virtual wheelchair was modeled using the program 3DStudioMax [205] and imported to the Unreal Editor as static meshes (*.usx). The model was then added to USARSim by writing appropriate Unreal Script classes and modifying the USARSim configuration file. Figure 3.6c presents a picture of the real wheelchair, and Figure 3.6d its simulated counterpart.

3.6 Local Obstacle Avoidance

Intelligent wheelchairs operating in dynamic environments need to sense its neighborhood and adapt the control signal, in real-time, to avoid collisions and protect the user. In this section a robust, real-time obstacle avoidance extension of the classic potential field methodology is proposed. The algorithm is specially adapted to share the wheelchair's control with the user avoiding risky situations. This method relies on the idea of virtual forces, generated by the user command (attractive force) and by the objects detected on each ultrasonic sensor (repulsive forces), acting on the wheelchair. The resultant wheelchair's behavior is obtained by the sum of the attractive force and all the repulsive forces at a given position. Experimental results from drive tests in a cluttered office environment provided statistical evidence that the proposed algorithm is effective to reduce the number of collisions and still improve the user's safety perception.

Motivated to answer to numerous mobility problems, many intelligent wheelchair related projects have been created in the last years [71]. While some initiatives have improved the autonomous function of the mobility aid [94, 206], others focused their work in sharing the wheelchair's control with the user [207, 102]. Shared control initiatives take advantage of the user's intelligence and assist the driver in the navigation process when dangerous situations are detected, extending and complementing user capabilities. In such a way, techniques as obstacle avoidance developed in fields of robotics have the potential to improve user's safety and reduce the navigation complexity. These methodologies consist basically on shaping the robot's path to overcome unexpected obstacles. A number of algorithms were develop to overcome obstacles and differ basically in the sensorial data used and control strategies. However, not all techniques are suitable to be implemented in a shared control paradigm. Some of the desired properties of shared control algorithms are:

- Avoid obstacles in real-time: since wheelchairs operate in dynamic environments, it is not feasible to implement popular time-consuming global path planners. Instead, such application is more suitable to approaches based on fast response like reactive/reflexive controls.
- Low computational cost: low memory and processing consuming algorithms are more likely to achieve a real-time reflexive behavior in embedded systems.
- Increase user safety and user safety perception: beyond a quantitative reduction in the number of collisions, shared control initiatives may consider qualitative evaluations of the wheelchair's overall behavior. In spite of imposing the control to the wheelchair, the algorithm may adapt the control signal to reduce the discomfort caused in driving tasks.

Furthermore, once intelligent wheelchairs are designed to carry people with disabilities, they should have the same durability, functionality and ergonomics concern of the standard powered wheelchairs. It not only constrains the number of sensors, but their type and position on the wheelchair. Therefore, the shared control algorithm may be robust enough to ensure the user safety even with non-optimal amount of information.

This section proposes and implements an extension of a classic obstacle avoidance technique known as potential field. Special attention was given to user autonomy, assisting the wheelchair's control just when dangerous situations were detected. The potential field concept was chosen as base for our implementation given its simplicity [22]. Especially due to the possibility to easily adapt the algorithm to cover the specific requirements of shared control paradigms and to run it on the limited computational capability of our prototype's embedded system. However, our work differs from the original PF because it does not try to build a world map of the environment. Instead, our approach is closer to the implementation described by Bicho *et al.* [167], where each ultrasonic range reading is treated as a repulsive force.

Once an object is detected by a sensor S_i, a virtual repulsive force F_i towards the robot is computed. The direction of each repulsive force is determined by the direction of λ_i, from the object point O_i to the Robot Center Point C, (Figure 3.7). Notice that since sonar sensors return radial measures of the environment, it is not possible to determine precisely the angular location of the object. However, it is much more likely that the detected object is closer to the acoustic

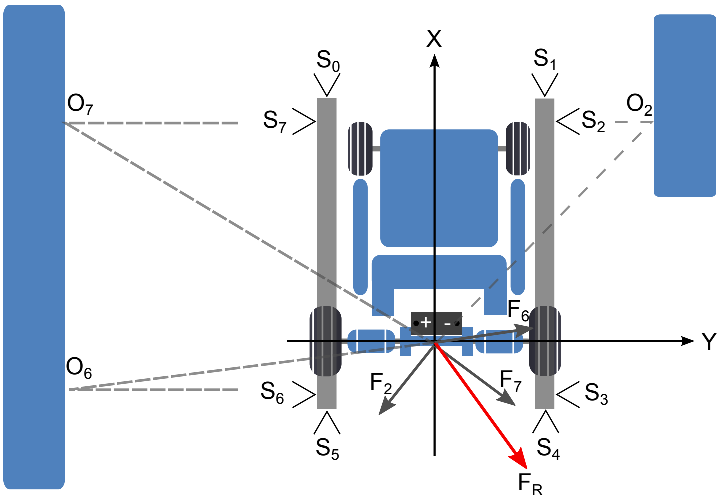


Figure 3.7: Representation of the repulsive forces acting on the wheelchair. F2, F6 and F7 represent the repulsive forces generated respectively by the objects O2, O6 and O7. Fr represents the sum of all the three repulsive forces.

axis of the ultrasonic transceiver then in the periphery of the conical field of view [163]. Thus, the position of obstacle O_i is computed as the measured distance D_i under the acoustic axis of the sensor.

$$\sigma_i = atan2(O_{iy}, O_{ix}) \tag{3.1}$$

where (O_{ix}, O_{iy}) is the relative position of obstacle detected by the sensor S_i, and σ_i is the direction from the detected object O_i to the wheelchair's center point C. The magnitude of the repulsive forces grow exponentially accordingly to the pair (D_i, Sp):

$$|F_i| = \alpha \exp(-\beta D_i + \omega Sp)|F_a| \tag{3.2}$$

where α, β and ω are positive constants deduced from the desired safety range, $|F_a|$ is the magnitude of the attractive force, D_i the distance to an obstacle O_i measured by the sensor S_i, and Sp the wheelchair speed. Once all repulsive forces are computed, they are added up to yield a resultant repulsive force F_r:

$$F_r = \sum_{i=0}^{n} F_i \tag{3.3}$$

Next, the virtual attractive force F_a induced by the target is updated. In the wheelchair implementation the force F_a is directly proportional to the current user input, which can be either the standard wheelchair's joystick or a special user interface which is based on the user's head position. Summing both the resultant repulsive force F_r and current attractive force F_a it is possible to derive the final force F_t that steers the wheelchair, Figure 3.7.

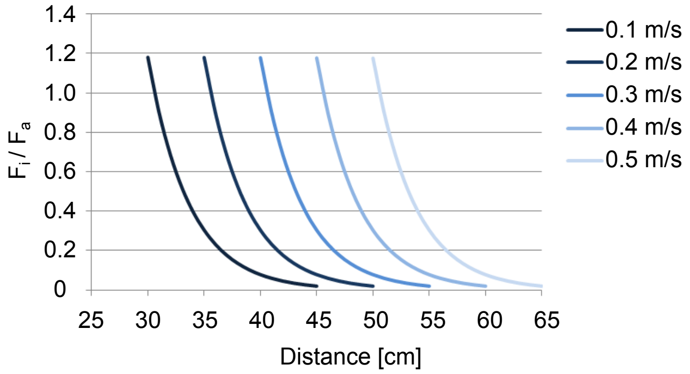


Figure 3.8: Safety distance range acording to the IW's speed and distance.

$$F_t = F_a + F_r \tag{3.4}$$

In order to keep user autonomy at the utmost, control signals are only adapted in situations were the user faces an eminent risk of collision. Therefore, repulsive forces start acting just when a safety range is reached. Due to inertia, the distance needed to completely stop the wheelchair increases with its speed Sp. Thus, the risk of collision is considered a bi-dimensional variable, both distance and speed dependent. Such safety range was designed not just to avoid obstacles in the wheelchair's neighborhood, but also to avoid oscillations that non-critical far objects could cause in the control's behavior. For the experiments, the values of the constants were empirically tuned according to the dynamics of the wheelchair, and set to $\alpha = 0.51$, $\beta = 0.271$ and $\omega = 14.8$. Figure 3.8 depicts the relation of F_i and F_a (Equation 3.2) according to the speed of the wheelchair and the distance to an obstacle.

3.7 Experiments and Results

The goal of this section is to present the description of the experiments performed to validate three contributions of this thesis regarding the development of intelligent wheelchairs. First, we present the evaluation of the obstacle avoidance methodology proposed in the Section 3.6. Next, we assess the features of popular robotic simulators in order to choose the most appropriate one regarding the IntellWheels requirements. Finally, we evaluate the extent of the visual/ergonomic modifications comparing the IntellWheels prototype with other intelligent wheelchair prototypes and with the original powered wheelchair used in the project.

3.7.1 Obstacle Avoidance Experiments

In order to evaluate the efficiency of the proposed obstacle avoidance algorithm, eight volunteers performed each one a set of four driving tests. Each set was composed of four laps: two laps in a simulated environment (one lap with and one lap without the assistance of the algorithm) and two laps in a real environment (one lap with and one lap without the assistance of the shared wheelchair

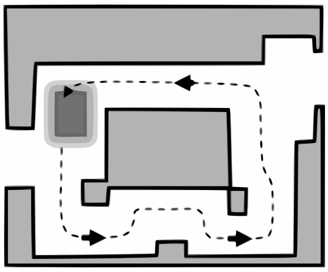


Figure 3.9: Representation of the closed circuit were experiments were conducted.

control). All of the eight recruited participants were aged between 26 and 39 years old, and have spent around 40 minutes running the experiments and answering a post-test questionnaire. Based on the work proposed by Parikh *et al.* [90], a well-defined protocol to conduct the test was designed. The protocol aims to ensure that data were collected accurately and in the same way across the tests, and will be better explained in the next sub-section.

Participants were instructed about the objective of the task and about the closed circuit they should drive, Figure 3.9. It was reinforced that their main goal was to drive safely and then to finish each lap as fast as they could. Time was just mentioned as a secondary objective to prevent volunteers from navigating too slowly, and was not used on the evaluation process.

Tests in the real environment were run using IntellWheels intelligent wheelchair prototype, and tests in the simulated environment were run under the IntellWheels Cyber-Mouse based simulator. During these trials, some conditions faced by handicapped individuals were simulated. To accomplish that, all participants were asked not to drive the wheelchair using its standard hand driven joystick. Instead, volunteers were requested to perform all four laps using IntellWheels head gestures input.

The experiment protocol has been defined to standardize the results of both tests, and consists basically of seven steps:

- **Step 1** Volunteers were instructed about the test procedure and about their objectives during the four drive tests.

- **Step 2** It was given to each participant a 10 minutes driving trial in a simulated environment. Thus, the user could experiment the wheelchair and make the necessary adjustments to the special human-machine interface.

- **Step 3** Once prepared, the participant was asked to drive the wheelchair (1 lap) through the circuit in the simulated environment with the manual control paradigm.

- **Step 4** After the first test, it was asked to the volunteer to drive the wheelchair (1 lap) through the same circuit in the simulated environment, but with the assistance of the shared control.

- **Step 5** Accomplished both tests in the simulator, the user were asked to drive the wheelchair (1 lap) in the real environment with the manual control.

- **Step 6** In the last test the user had to drive the wheelchair (1 lap) in the real environment with the shared control paradigm.
- **Step 7** To evaluate the shared control paradigm, the user safety perception and to conclude the set of experiments, a pot-task questionnaire was applied.

From the set of experiments described above, both quantitative and qualitative data have been generated. All analysis were performed within subjects, which allowed us to estimate if providing assistance actually helped each individual, rather than testing the performance of individuals against each other. Therefore, experimental data were subjected to a nonparametric test for paired samples (Wilcoxon Signed Rank 1-tailed test) [208], which made possible to reach some conclusions with a confidence level of 95% (p<0.05).

Based on the number of collisions of each trial, the shared algorithm performance could be evaluated in the simulated (Figure 3.10) and real environments (Figure 3.11). In the real environment, the statistical analysis indicate a significant reduction in the number of collisions with the shared control paradigm $(T = 0.00, p = 0.0135)$. The same conclusion can be drawn for the shared control in the simulated environment $(T = 0.00, p = 0.009)$.

Another interesting aspect to consider is the evaluation of the algorithm from the user's perspective. Related projects concluded that, despite the reduction in the number of collisions provided by their shared control algorithms users did not felt safer indeed, and gave preference to the standard driving paradigm without any assistance.

In order to measure the user perception, we invited the volunteers to specify their opinion regarding twelve statements (six for each control paradigm) of a questionnaire. The respondent level of agreement with statement was measured through a typical five-point Likert item (1 = Strongly disagree, 2 = Disagree, 3 = Neither agree nor disagree, 4 = Agree, 5 = Strongly agree).

1. I feel comfortable when driving the wheelchair.
2. I feel that I have the control of the wheelchair behavior.
3. It is easy to drive the wheelchair in cluttered spaces.
4. Driving the wheelchair requires little attention.
5. The wheelchair has the same behavior either in the simulated and the real environments.
6. I believe that the shared control helped me during the navigation task.

In our analysis, the user safety perception was treated as an indirect variable measured through the sum of the points of the statements 1, 2, 3 and 4. Results are depicted in Figure 3.12). The difference between the user safety perception with and without the shared control paradigm was significantly greater than zero $(T = 3.0, p = 0.027)$, providing evidence that the shared control is effective to improve user's safety perception.

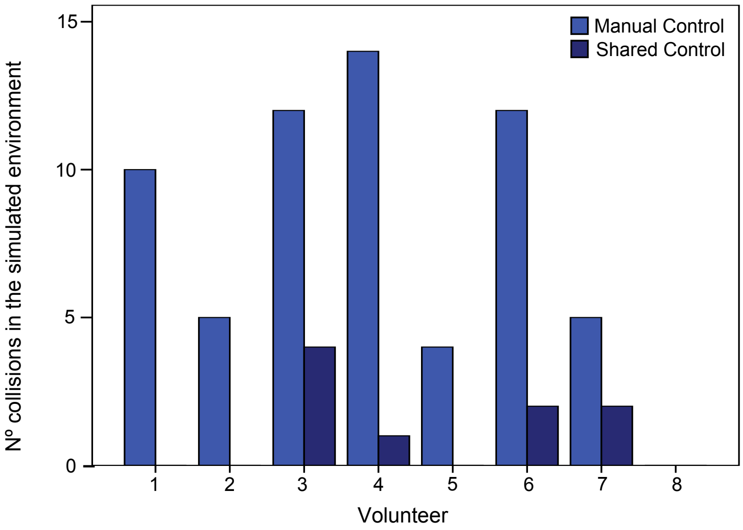


Figure 3.10: Number of collisions per volunteer in the simulated environment.

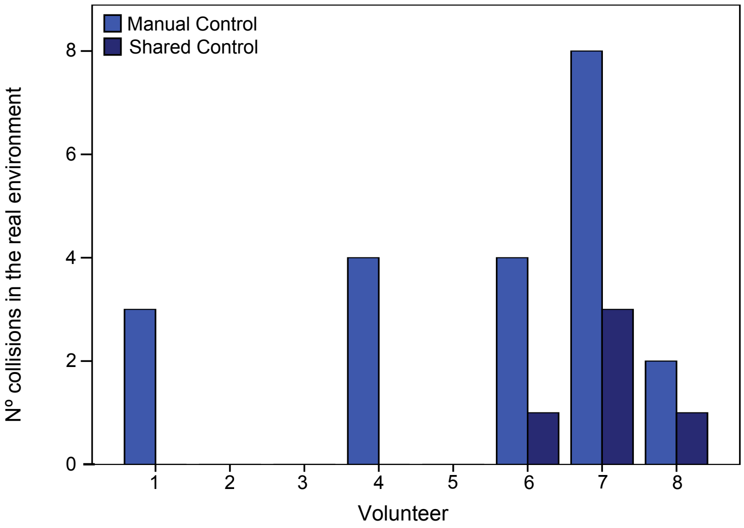


Figure 3.11: Number of collisions per volunteer in the real environment.

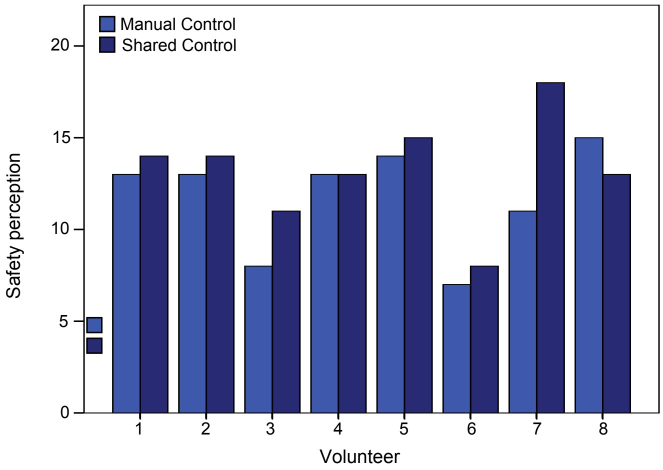


Figure 3.12: User’s perception of safety with and without the assistance of the shared control.

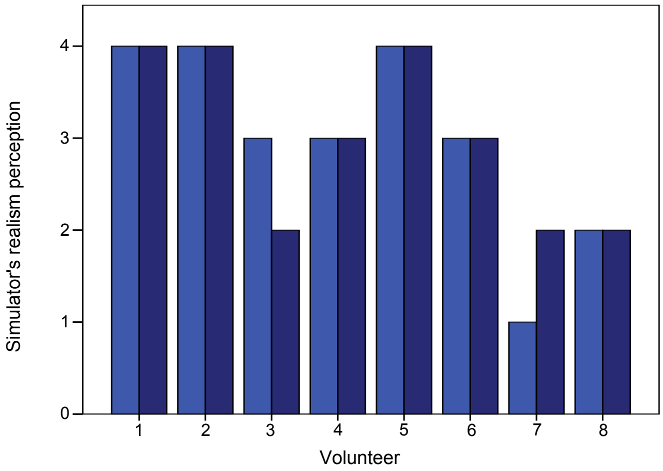


Figure 3.13: User’s perception of the simulator realism regarding the manual and the shared control paradigms.

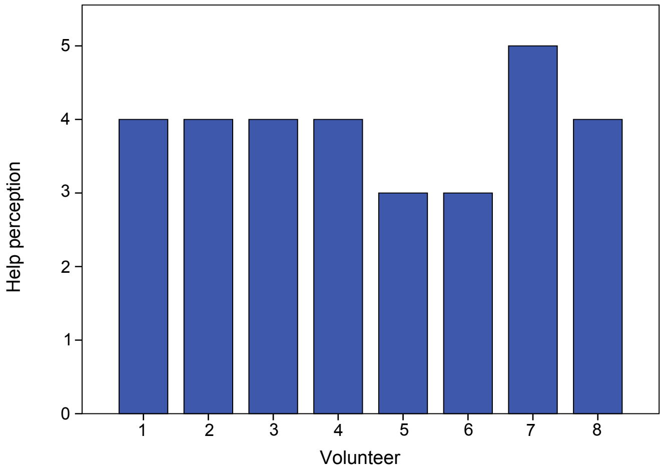


Figure 3.14: User’s perception of the assistance provided by the shared control paradigm.

Another inference can be drawn regarding to the behavior of the wheelchair in the simulator. Through the fifth statement, we tried to measure how close to the reality the simulated behavior of the wheelchair is. A threshold value of 3 was used to compare results, Figure 3.13. Since in the Likert item a value of 3 means that respondents neither agree nor disagree with the statement, a value greater than 3 means that simulated wheelchair reacted just like the real one in the user's perspective. Through the Wilcoxon signed rank test it was not possible to state with a confidence level of 95% that the wheelchair presented the same behavior in neither the manual $(p = 0.0655)$ and shared control $(p = 0.051)$ paradigms.

Finally, one last result of the questionnaire concerns with the user's perception of the assistance provided by the wheelchair. In this case it was evaluated through the statement 6 (only present in the shared control section of the questionnaire), comparing it with a threshold value of 3 (Figure 3.14). Similar to what was mentioned before, a value greater than 3 means that the user felt assisted by the algorithm. The statistical analysis provided evidence that volunteers indeed felt that the shared control paradigm helped them to drive the wheelchair $(T = 0.0, p = 0.01)$.

3.7.2 Assessment of Robotic Simulators

Currently, an extensive number of general simulators are available for robotics research [193]. However, specificities of the project had to be taken into account when choosing a new tool to simulate the IntellWheels prototype. Therefore, in order select the robotic simulator that better fits the requirements of the IntellWheels project, we evaluated four popular robotic simulators using a set of seven criteria:

- Support to import 3D models – we define this criteria as the ability of a simulator to import three-dimensional models of objects from typical Computer Aided Design (CAD) programs (such as Solidworks, Autocad, Pro-engineer, etc). We believe that this ability can facilitate the development of a more realistic model, thus improving the simulation. The evaluation of this criterion receives "yes" when the simulator supports importing objects and "no" when it does not.

- Programming language – in this criterion we identify which programming languages are supported by the simulator to create the program that controls the robot. A wide support in the programming language criteria is desired. In addition, we look specifically for a simulator that supports object pascal, once the IntellWheels platform is currently under development in that language. The evaluation of this criterion receives the list of the supported languages.

- External agent support – concerns the ability to run the agent(s) that control the robot from outside of the simulator. This characteristic is desired because we want to be able to distribute the agents that control an IntellWheels prototype and the agents that provide additional services in more than one computer. This way, it is possible to increase the robustness of the system, since an agent can assume the tasks of other agents that for any reason are not answering. The evaluation of this criterion receives "yes" when the simulator supports external agents and "no" whenever it does not.

- Multi-thread support – is the ability of the simulator to run more than one simulation task simultaneously. This ability is important to improve the simulation efficiency. The evaluation of this criterion receives "yes" when the simulator supports multi-thread and "no" when it does not support.

- Physics Engine - concerns the identification of the libraries used for computing physics simulation. The main task of all physics engines is to solve the motion of the system given the forces acting on it. Therefore, they play a very important role in the simulation of dynamic systems because they are directly responsible for its functional fidelity. On the other hand, physics engines have a indirect responsibility also in the physical fidelity of the simulation. Particularly, the way that a simulation looks is closely dependent on the type of features the physic engine is able to simulate. For example, simulations with deformable objects demonstrate a greater realism over those which consider objects as rigid bodies, the simulation of fluids, like fog, may be important for machine vision and for video feedback, and so on. The evaluation of this criterion receives the name of the library used in each simulator.

- License – corresponds to the monetary cost for the developer and for the end user. The evaluation of this criterion can receive the value "Open Source" for those simulators that are released with their source code, "free" for simulators that are available without any monetary compensation and without their source code, and "commercial" for those simulators that require monetary compensation.

- Sensors – in this last criterion we identify which sensors are released with the simulators and if the simulator allows developers to create new sensors.

Each simulator was assessed through its user manual or equivalent documentation. The result of such evaluation and a summary of the sensors available in each simulator is presented in the Table 3.2. With the exception of Gazebo and SimTwo, all simulators evaluated can import 3D models from typical CAD tools. Regarding the programming language, both USARSim, SimTwo, V-REP and Gazebo can cope with a wider support. These simulators rely on a client/server architecture with communication through TCP/UDP protocol, which also provides the support to external agents. Regarding multi-thread support, only USARSim, Microsoft Robotics Studio and V-REP are able to benefit from the simultaneous task processing.

Despite several libraries for physics computation available (PhysX, Bullet, JigLib, Newton, ODE, Tokamak, True Axis) [193], PhysX and ODE are dominant in the simulators under analysis. ODE (Open Dynamics Engine) is an open-source library that is designed for simulations of rigid bodies and articulated bodies dynamics. For this reason, this library is not able to support the simulation of deformable objects, particles and fluids. ODE is platform independent with an easy

Table 3.2: Comparison of the characteristics and sensors of the six 3D general robotic simulators.

Features	USARSim [195]	RDS [196]	Webots [197]	SimTwo [198]	V-REP [199]	Gazebo [200]
Import 3D models	yes	yes	yes	no	yes	yes
Programming language	Any (UDP)	C#, VB, JScript, IronPython	C,C++, Java, Python, MATLAB	Any (UDP)	Any (TCP/UDP)	Any (TCP/UDP)
External agent support	yes	no	no	yes	yes	yes
Multi-thread support	yes	yes	no	no	yes	no
Physics Engine	UT3 with PhysX	PhysX	ODE	ODE	Bullet and ODE	ODE
License	Open Source*	Free	Commercial	Free	Free*	Open Source

Sensors	USARSim [195]	RDS [196]	Webots [197]	SimTwo [198]	V-REP [199]	Gazebo [200]
Encoder	yes	yes	yes	yes	no	yes
Camera	yes	yes	yes	yes	yes	yes
Touch sensor	yes	yes	yes	yes	no	yes
Infra-red	yes	yes	yes	yes	yes	yes
Ultra-sound	yes	yes	yes	yes	yes	yes
Sound sensor	yes	no	no	no	no	no
GPS	yes	yes	yes	yes	no	yes
RFID	yes	no	no	no	no	yes
Laser Range Finder	yes	yes	yes	yes	yes	yes
Create new sensor	yes	no	yes	yes	yes	yes

to use C/C++ API. The kind of applications ODE was developed for also explains some of its characteristics, since ODE was developed to prioritize computational speed over physics accuracy. On the other hand, PhysX is a proprietary solution widely used in Epic games. It provides support to the main platforms for games and graphics (such as PS3, XBOX, PC, etc.). Its main advantage consists in supporting not only rigid and articulated bodies, but also fluids (such as water, blood, smoke, gas, etc.) and particles (such as sparks, scattered glass fragments, dust, etc.). PhysX has a faster physics integration algorithm, and provides a more stable simulation when dealing with the collision of several objects [209]. In addition to the physics library, nVidia has also developed a special hardware device: the Physics Processing Unit (PPU).

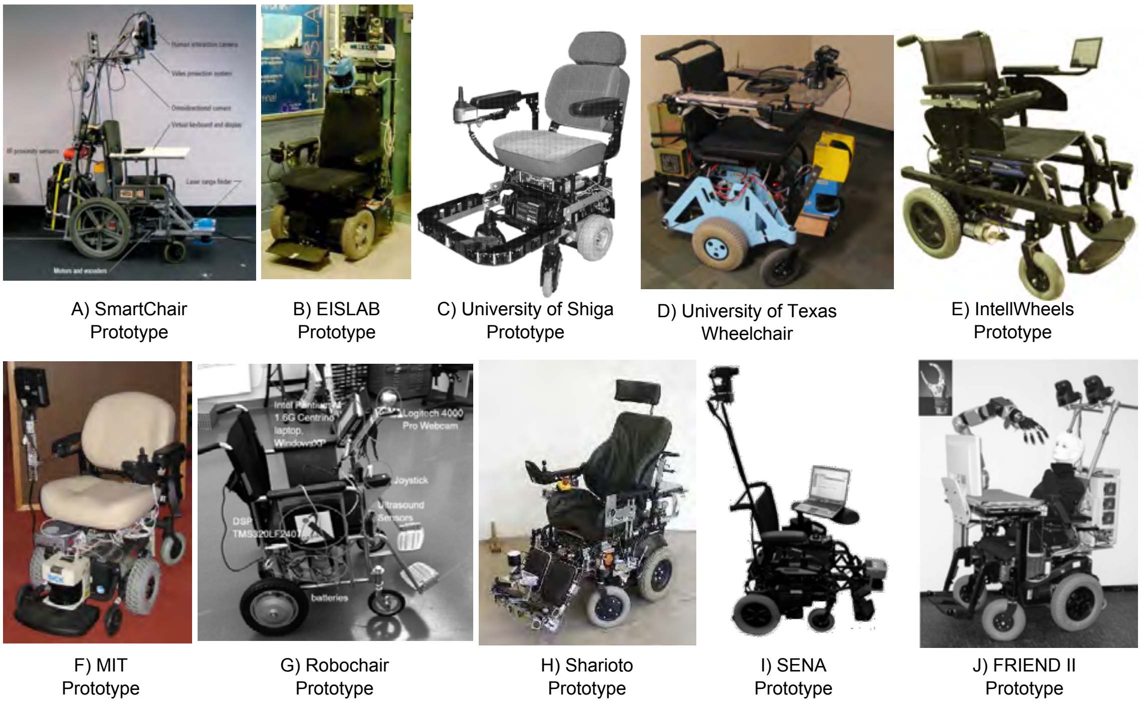
With respect to the license, Gazebo and USARSim are open source simulators. At this point, it may be noticed that USARSim is open source simulator, but its current version relies on a proprietary engine that is free for noncommercial and educational use. In the RDS 2008 R3 version, Microsoft has combined the previous Express, Standard and Academic licenses into one license free of charge. SimTwo is free and V-REP only recently adopted a license model which is free for noncommercial and educational use. Webots is the only simulator evaluated that has only a commercial license, with versions that costs from 250.00€for up to 2600.00€. Finally, the analyses of the sensors criteria revealed that both SimTwo and V-REP does not provide simulation to all the sensors used in IntellWheels hardware framework. Another severe limitation was observed in the RDS, SimTwo and V-REP, which does not allow researchers to develop new sensors.

3.7.3 Assessment of the Visual Appearance of the IntellWheels Prototype

The low visual/ergonomic impact of the IntellWheels prototype is another contribution in the development of intelligent wheelchairs. In order to evaluate the extent of such changes we conducted a public opinion poll about the visual appearance of several intelligent wheelchair prototypes. In the survey, respondents were invited to express their level of agreement to fourteen statements through a typical five-level Likert scale (1 = Strongly disagree, 2 = Disagree, 3 = Neither agree nor disagree, 4 = Agree, 5 = Strongly agree).

The assessment was answered by 128 individuals, of which 43.8% males and 56.3% females, with a mean of age of 24.2 years old (Std = 7.26). Respondents were selected by convenience, composed essentially by Master and PhD. students from the Faculty of Engineering of the university of Porto and from the School of Allied Health Sciences of the Polytechnic Institute of Oporto. For this reason, subjects presented a high education level, with 32% having high school diplomas, 39% Bachelor degrees, 25% Master degrees and 3.1% Doctors degrees. The majority of the sample, 54.7%, is composed by subjects with no direct relation with physically disabled people, while 31.3% were health care professionals, 12.5% were relatives or friends of wheelchairs users and 1.6% were users of manual wheelchairs. The lower proportion of people with disabilities may not affect the validity of this study since the object under analysis, the public opinion about the visual appearance of intelligent wheelchair prototypes, is not determined by the condition of the subjects.

The questionnaire was composed of two parts. The goal of the first part was to evaluate the visual appearance of IntellWheels comparatively to other intelligent wheelchair prototypes. Based



A) SmartChair Prototype

B) EISLAB Prototype

C) University of Shiga Prototype

D) University of Texas Wheelchair

E) IntellWheels Prototype

F) MIT Prototype

G) Robochair Prototype

H) Sharioto Prototype

I) SENA Prototype

J) FRIEND II Prototype

Figure 3.15: Prototypes of ten intelligent wheelchair projects: (A) SmartChair, (B) EISLAB, (C) University of Shiga, (D) University of Texas, (E) IntellWheels, (F) MIT, (G) Robochair, (H) Sharioto, (I) SENA, (J) FRIEND II.

on Figure 3.15, respondents were asked to express their level of agreement with the following statement:

> *The addition of sensors and other hardware devices had visual/ergonomic impact on the wheelchair (e.g. changed the normal appearance/usage of the Wheelchair).*

Results of the first part of the questionnaire are depicted in the Figure 3.16. An analysis within subject performed with the Wilcoxon signed rank test (Table 3.3) provides statistical evidence that, among the ten projects, IntellWheels presented the lowest change in the normal appearance of the wheelchair.

Table 3.3: Wilcoxon signed rank test: comparison between the visual impact of IntellWhells with other intelligent wheelchair prototypes.

	- Ranks	+ Ranks	Ties	Total	Z	p
IntellWhells - SmartChair	76	20	32	128	-5.622	<0.001
IntellWhells - Robochair	41	21	66	128	-2.403	0.008
IntellWhells - Shiga	48	24	56	128	-2.756	0.003
IntellWhells - Texas	69	15	44	128	-5.547	<0.001
IntellWhells - MIT	39	26	63	128	-2.306	0.010
IntellWhells - EISLAB	58	14	56	128	-4.566	<0.001
IntellWhells - Sharioto	34	22	72	128	-2.446	0.007
IntellWhells - SENA	73	15	40	128	-5.613	<0.001
IntellWhells - FRIENDII	80	15	33	128	-6.218	<0.001

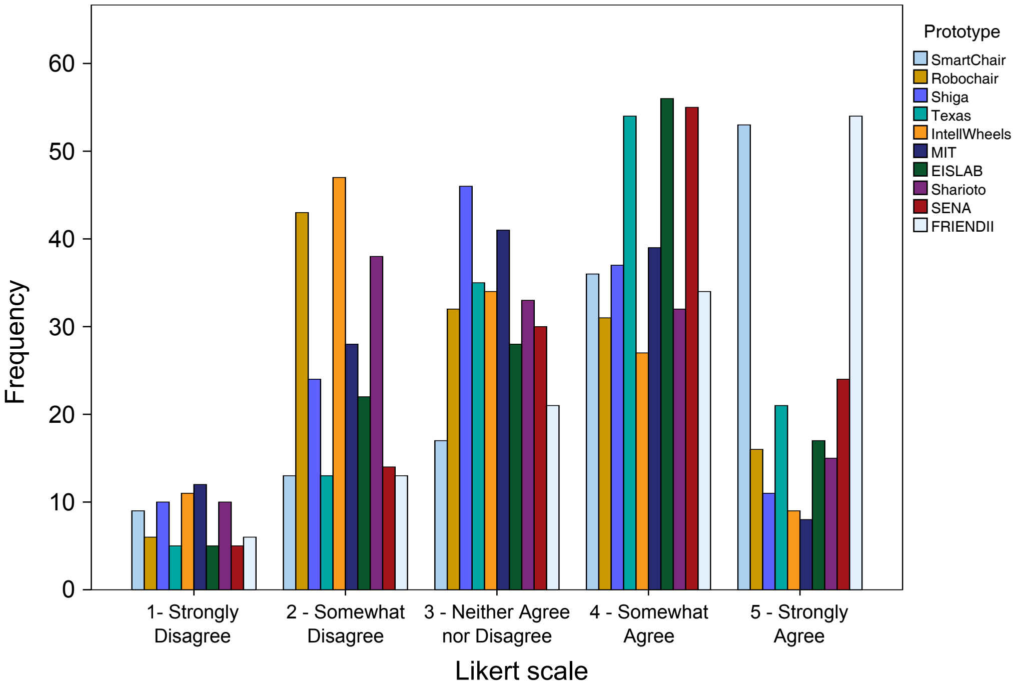


Figure 3.16: Responses indicating the level of agreement with the statement: The addition of sensors and other hardware devices had visual/ergonomic impact on the wheelchair (e.g. changed the normal appearance/usage of the Wheelchair).

The second part presented to the respondents two images: one image of the IntellWheels prototype and one image of the original powered wheelchair used in the project. The goal was to assess the visual impact of the modifications performed in the wheelchair as a whole, as well as the visual changes introduced by specific hardware devices (display, sensor bars and other hardware devices). Based on the Figure 3.17, respondents were requested to express their level of agreement with four statements:

In comparison with the original powered wheelchair, global visual/ changes of the IntellWheels prototype are small.

In comparison with the original powered wheelchair, visual changes introduced by the display are small.

In comparison with the original powered wheelchair, visual changes introduced by the sensor bars are small.

In comparison with the original powered wheelchair, visual changes introduced by the computer and other hardware are small.

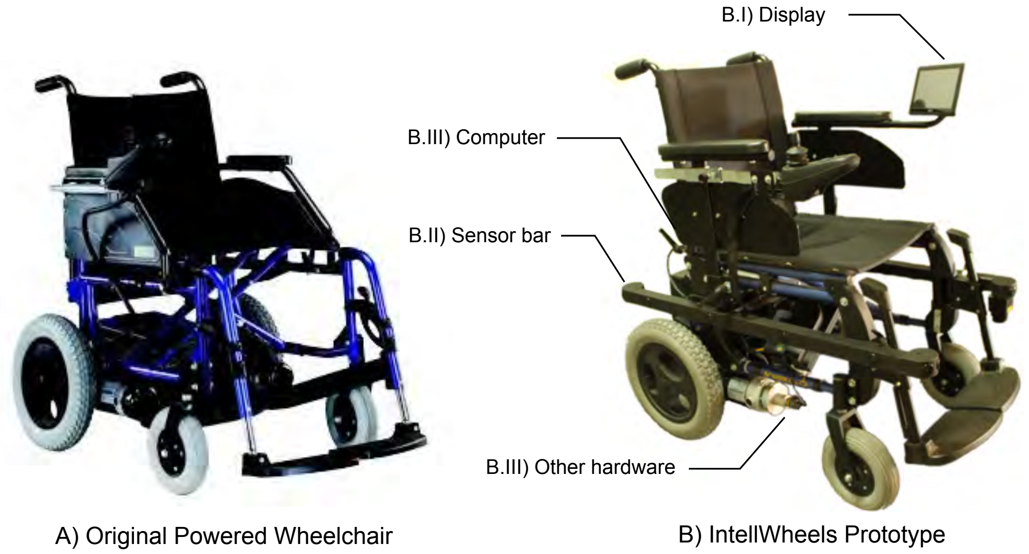


Figure 3.17: Original powered wheelchair (A) and the IntellWheels prototype (B). In detail, the (B.I) display, (B.II) sensor bars and (B.III) computer and other hardware devices.

Figure 3.18 depicts the responses of the second part of the questionnaire. An analysis within subject was performed with the Wilcoxon signed rank test (Table 3.4) by comparing the opinions regarding the four statements to the neutral hypothesis (3 = Neither agree nor disagree). At a level of significance of 0.05, there exists enough evidence to conclude that both the display, sensor bars and other hardware devices had only a small visual impact. Further statistical results indicate that the IntellWheels prototype was able to keep the overall aspect of the original wheelchair.

Table 3.4: Wilcoxon test: visual impact in the IntellWhells prototype. Opinions regarding four items were compared to the neutral hypothesis (3 = Neither agree nor disagree) to verify if there is statistical support to the claim that the modifications performed in the wheelchair are small.

	- Ranks	+ Ranks	Ties	Total	Z	p
Hypothesis - Global	22	86	20	128	-6.036	<0.001
Hypothesis - Display	23	86	19	128	-6.319	<0.001
Hypothesis - Sensor bar	31	72	25	128	-4.217	<0.001
Hypothesis - Other hardware	19	90	19	128	-7.051	<0.001

Despite none of the hardware devices analysed presented a significant visual modification in the wheelchair, an interesting outcome of the second part of the questionnaire is the ordering of the hardware devices by its visual impact level. Therefore, next designs can take such information into account an collaborate on reducing the rejection to assistive robotics. In order to sort the hardware devices, it was performed a within subject analysis with the Wilcoxon signed rank test, comparing in pairs the level of agreement given to each one of the three devices. From the results summarized in the Table 3.5, it was found evidence that highest impact was provided by the sensor bars, followed respectively by the display and by the other hardware.

Table 3.5: Wilcoxon test: comparison of the visual changes introduced by (I) display, (II) sensor bars and (III) computer and other hardware devices.

	- Ranks	+ Ranks	Ties	Total	Z	p
Sensor bar - Display	40	25	63	128	-1.948	0.025
Other hardware - Display	17	36	75	128	-2.532	0.006
Other hardware - Sensor bar	12	46	70	128	-4.263	<0.001

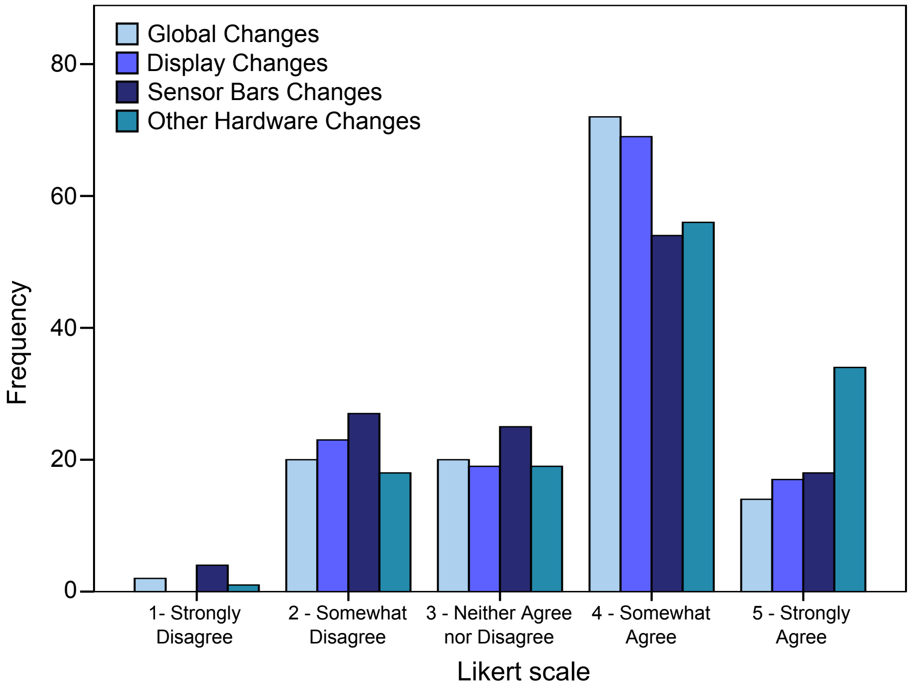


Figure 3.18: Responses indicating the level of agreement with the statements: In comparison with the original powered wheelchair... global visual changes of the IntellWheels prototype are small; visual changes introduced by the display are small; visual changes introduced by the sensor bars are small; visual changes introduced by the PC and other hardware are small.

3.8 Conclusions

In this chapter, we presented a general overview of the project IntellWheels, and three contributions of this thesis to the development of intelligent wheelchairs. IntellWheels was defined as a generic platform for research and development of intelligent wheelchairs. Its modular architecture enables an easy integration of distinct sensors, actuators, user input devices, navigation methodologies, intelligent planning techniques and cooperation methodologies. The communication module demonstrated to be a mean to enable fault-tolerant communications in open transmission systems, and to work as a facilitator for entity collaboration. The generic hardware framework of the platform is designed to facilitate the conversion of ordinary powered wheelchairs into intelligent wheelchairs with minor changes. In addition, due to the use of low cost off-the-shelf devices, it presented a cost effective solution for assisting severely impaired individuals. The estimated cost of the hardware framework was designed not to exceed the cost of ordinary powered wheelchairs, available in the market with a starting prices around 2.000,00€. The multimodal interface extended the regular human-wheelchair interaction by allowing the user to create its own association between combinations of multiple (and/or heterogeneous) inputs with one wheelchair action.

The proposed obstacle avoidance methodology relies on the dynamic approach of the classic field of forces concept. This work extends and complements the potential field methodologies from a shared control perspective. To reduce the computational cost and run the algorithm in real-time, each ultrasonic range reading is treated as a repulsive force. Thus, it is not necessary to build a map of the environment and compute thousands of parameters. Furthermore, as localization is not required, dead reckoning errors are not introduced when computing the distance to obstacles. The experimental results in both simulated and real environments indicate that the proposed methodology is effective in reducing the number of collisions. In addition, the algorithm demonstrated to be able to increase the user perception of safety and their feeling of assistance. An interesting observation concerns to the number of collisions in the simulated environment, which is in general much greater than in the real environment. A possible explanation for this might be related to the fact that the collisions in the simulated environment does not represent a life-threatening situation. Therefore, there is a tendency to relax and to reduce the attention to the circuit. Another possible cause is related to lack of realism provided by the Cyber-mouse based simulator. Volunteers reported that the 3D environment of the simulator could not provide an accurate perception of depth and distance to objects, causing collisions in the cluttered test circuit.

Such results showed the importance of increasing the simulation realism. The first step towards a new and more realistic version of the simulator was the assessment of the robotics simulators available. For this purpose, six general robotic simulators were evaluated based on a set of seven criteria. Due to specific project requirements and to its unique combination of features, USARSim was selected to simulate the IntellWheels prototypes. We have considered the lack of support for Object Pascal of the RDS, V-REP and Webots, the lack of sensors of SimTwo and V-REP, the limitation in the development of new sensors of RDS, SimTwo and V-REP, the cost of Webots, and the lack of support to multi-task processing and to import 3D models as the main problems of the other simulators. USARSim, on the other hand, presented a superior physics engine, validation of several sensors and actuators and is currently one of the most used robotics simulator within the scientific community.

Another contribution of this thesis is the mitigation of the visual and ergonomic impacts caused by the sensorial and processing capabilities. Figure 3.5 demonstrates that despite the assemblage of the several sensors that composes the hardware framework, the accessibility to the wheelchair was not compromised. The assessment of the IntellWheels visual appearance indicated that not only the prototype presented the lowest visual impact between ten other intelligent wheelchair prototypes, but also that its overall aspect is similar to the original powered wheelchair. A more detailed analysis of three groups of hardware devices added to the wheelchair suggested that none of than caused a significant visual impact. Such result validate the design of the IntellWheels prototype and contribute to increase the acceptance of assistive robotics by the general population.

Finally, despite the several concepts proposed through the IntellWheels platform, not all modules achieved the robustness required to assist in an autonomous fashion physically disabled and elderly people. An example is the IntellWheels dead-reckoning based localization. Although, Borenstein and Feng [6] proposed a methodology to compensate systematic errors in odometry, they assume that the dimension of the wheels does not vary in time. In the first IntellWheels prototype [15] we proposed two solutions to reduce dead-reckoning errors. The error accumulated over time is reset whenever the wheelchair detects an artificial landmarks. Therefore, the strategic level of the control module forces the wheelchair to re-plan its path to pass over the nearest artificial landmark whenever the localization uncertainty overcomes a pre-defined threshold. The second was the couple the encoders in passive wheels (rather than on the motor shaft), reducing the errors caused by wheel slippage. These additional wheels were mounted on levers located internally to the wheelchair and parallel to the rear wheels. The contact of the auxiliary wheels with the floor was granted by gravity and also by compression springs [19, 15]. Another advantage of such approach was that the errors derived from variations in the wheel diameter (in the wheelchair case due to the use of rubber tires filled with compressed air) can be avoided since solid undeformable wheels can be used. The problem of such an approach is that the mechanical assemblage of the mechanism is not trivial and has to be designed specifically to each model of wheelchair. An additional limitation of such approach is that it is not suited to estimate localization in rush and uneven grounds, like those usually found in outdoor environments.

In an attempt to match IntellWheels flexibility requirements, the latest prototypes assembled the encoders directly in the motor shaft. This solution, however, lacked to provide robust and reliable odometry estimations. Vision methodologies, on the other hand, present themselves as an interesting alternative due to its independence from wheel-terrain interactions. Therefore we propose the use of feature based visual odometry as means to provide localization to the wheelchair. Next chapter presents the proposed methodologies to increase robustness in the detection of image features.

Chapter 4

Photometric Invariant Feature Detection

Image features are the main primitives for several visual tasks. Therefore the overall algorithm will often only be as good as its feature detector. However, most of the original detectors are not able to cope with large photometric variations, and the extensions that should improve detection eventually increase the computational cost and introduce more noise to the system. In this Chapter, we present two approaches that extend the original SURF algorithm increasing its invariance to illumination changes. While some authors use color space mapping to achieve invariance, our first approach uses local normalization and the second approach uses local space average color descriptor to detect invariant features. A theoretical analysis demonstrate the effects that distinct photometric variations have on the response of local image features detected with the Harris corners, SIFT and SURF algorithms. Experimental results demonstrate the effectiveness of the proposed approaches in several illumination conditions including the presence of two or more distinct light sources, variations in color, offset and scale.

4.1 Introduction

As previously discussed in the Background Chapter (Section 2.8), feature points are pixels that differ from its local neighborhood (such as T-junctions, corners and symmetry points) and are likely to be found in other images of the same object. Since features are used as main primitives for several vision-based localization algorithms, the overall algorithm will be only as good as its feature detector. For this reason, it is extremely important that the extracted features are robust to noise and invariant with regard to geometric (such as changes in scale, translation, rotation, affine/projective transformation) and photometric variations (illumination direction, intensity, color and highlights). According to Lemaire [210], the only solution to guarantee bounded errors on the position estimates of vision-based localization methodologies is to rely on stable environment features.

A common alternative to deal with the presence of outliers in the features extracted is the use of iterative data fitting algorithms, like RANSAC (Section 2.7). However, depending on the characteristics of the data, RANSAC can become computationally expensive, since the number of iterations N is exponential in the number of data points required to estimate the model. Therefore, there is a high interest in finding the minimal parameterization of the model. The most general motion model required 8 point correspondences [211]. Using the 6-point algorithm [212] would decrease the number of necessary iterations and therefore speed up the motion estimation algorithm. For unconstrained motion of a calibrated camera there would be necessary 5 point correspondences [213, 214]. In the case of planar motion, the motion model complexity is reduced and the parameters estimated with 2 point correspondences [215]. Even more restrictive motion models can be chosen, allowing a parameterization with only 1 feature correspondence [216, 217]. Computational savings of these restrictive models can be easily derived. For example, with the inlier ratio $\omega = 0.5$ and a probability $p = 0.99$, the number of random hypothesis can be reduced from 146 ($s = 5$, no prior information used) to only 7 ($s = 1$, using prior information). Table 4.1 shows the number of RANSAC iterations needed for motion estimation algorithms with different number of minimal data points s.

Table 4.1: RANSAC: relationship between the minimum set of points and the minimum number of iterations necessary to guarantee that at least one of them is mismatch-free ($p = 0.99$ and $\omega = 0.5$).

Minimum set of points	8	6	5	3	2	1
N^o of iterations	1177	293	146	35	16	7

The drawback of such approaches is that they assume certain motion constraints, or required partially known camera calibration parameters. Another alternative to reduce the number of RANSAC iterations consist on increasing the inlier ratio. As an example, with a probability $p = 0.99$, the number of random hypothesis of a general motion model algorithm can be reduced from 70188 to only 78 by increasing the rate of inliers from 30% to 70%. Table 4.2 shows the number of RANSAC iterations N needed for 8-point motion estimation algorithms according to the expected rate of inliers. The reader can note that by increasing the rate of inliers the required number of iterations can also become extremely low.

Table 4.2: RANSAC: relationship between the inlier ratio and the minimum number of iterations necessary to guarantee that at least one of them is mismatch-free ($p = 0.99$ and $s = 8$).

Inlier ratio %	90	80	70	60	50	40	30	20	10
N^o of iterations	9	26	78	272	1177	7025	70188	1798893	460517014

In this chapter, we propose two methodologies to increase the robustness in feature detection, and thus the number of inliers extracted from each image scene. We demonstrate that the two detectors extend the original SURF algorithm by providing invariance to large changes in illumination. We characterize the performance of the proposed extensions on a large dataset of controlled images. The robustness of the detectors is validated through real experiments using two datasets containing real world image.

The outline of the chapter is the following. Section 4.2 describes relevant related works in the areas of color image feature detectors and color constancy algorithms. Section 4.3 presents a theoretical derivation of the effects of illumination changes in the three most common feature detectors. Section 4.4 lays out our procedure to extract robust image features through a local normalization approach. In Section 4.5 we address the problem of large illumination variations proposing a methodology that combines SURF with a color constancy methodology. Section 4.6 presents a detailed description of the image datasets, the metrics used to perform the evaluation and the results comparing the repeatability of SURF with the two proposed extensions. Finally, the summary and conclusions of this chapter are presented in Section 4.7.

4.2 Literature Review

Color information can be used in image processing to simplify the identification and extraction of objects from a scene. color images carry more information than gray level images, and thus provide a broader class of discrimination between material boundaries [57]. In addition, color information enables one to distinguish between true color variation and photometric distortions [58]. Thus, when colored images are represented only through their intensity value, a very important source of information is lost [48].

4.2.1 Color Feature Detector and Descriptors

Originally, most of the feature detectors and descriptors were designed to cope only with the pixel intensities, discarding the three layers of information provided by RGB images. Later, in order to take advantage of all the information that RGB cameras are able to provide, some researchers proposed extensions for the original algorithms.

In [218], Ancuti and Bekaert proposed an extension to the SIFT descriptor (SIFT-CCH) that combines the SIFT approach with the color co-occurrence histograms (CCH) computed from the Nrgb color space. Their algorithm performs the same as SIFT in the detection step, but introduces one dimension to the descriptor. Thus, features are described by a two element vector that combines the SIFT and the CCH descriptor vectors. The main problem of such an approach is the increase in the computational effort during the feature matching due to the extra 128 elements added to the descriptor vector. The color-SURF proposed by Fan *et al.* [219] was maybe the first approach suggesting the use of colors in SURF descriptors. Through a methodology similar to the SIFT-CCH, the authors propose the addition of a new dimension to the descriptor vector. This extra information corresponds to the color histogram computed from the YUV color space, and adds a 64-element vector for each feature descriptor. For this reason, just like in the SIFT-CCH, the extra elements in the descriptor vector increase the computational effort necessary during the matching step.

In [48], Abdel-Hakim and Farag use the invariant property H (related to hue) of the Gaussian color model [57] as working space. Thus, instead of using gray gradients to track SIFT features, they use the gradients of the color invariant to detect and describe features. Although the authors used the H invariant instead of the C invariant, the approach is called CSIFT in a reference to the introduction of color in the SIFT operator. In [58], Burghouts and Geusebroek also use invariants derived from the Gaussian color model [57] to reduce the photometric effects in SIFT descriptions. They compare the individual performance of four invariants with the original SIFT approach, with the CSIFT approach [48] and with the HSV-SIFT approach [220]. Their evaluation suggests that the C-invariant, which can be intuitively seen as the normalized opponent color space, outperforms the original SIFT description and all the other approaches. In reference to the results of the C-invariant, the combination of this invariant with the SIFT operator is called C-SIFT.

Bosch *et al.* [220], on the other hand, take advantage of a Harris operator in order to detect stable features. Each feature is than characterized by computing SIFT descriptors for each component of the HSV color space (128 for each channel), which gives a total of 3 x 128-element vector for each feature. The main problem to such approach is to add the value component to the feature description. Once this component is the lightness by definition, it does not provide the desired photometric invariance. Thus, the complete descriptor has no invariance properties. Sande *et al.* [39] presents an evaluation of the different approaches that attempt to provide photometric invariance to SIFT like descriptors.

In addition to the methodologies previously proposed in the literature (i.e. HSV-SIFT, Hue-SIFT, C-SIFT), the authors compare the performance of new strategies like the rgSIFT, OpponentSIFT and RGB-SIFT. Tests were performed in three data sets of different visual categories under varying illumination conditions, like light intensity change, intensity shift, color change, arrangement change, viewpoint change and change in the quality of the image compression. The conclusion suggests that the performance of the descriptors vary according to the analyzed dataset, but that in general the OpponentSIFT and C-SIFT strategies presents the best results. Table 4.3 summarizes the extensions of local feature detection algorithms that somehow consider color information.

Table 4.3: Summary of the main characteristic of the related works.

Related work	color space	Detector	Descriptor	Dimension
SIFT-CCH [218]	Grayscale/ nRGB	SIFT	SIFT	2 x 128
Color-SURF [219]]	Grayscale/YUV	SURF	SURF	2 x 64
CSIFT [48]	Gaussian	SIFT	SIFT	1 x 128
HSV-SIFT [220]	HSV	Harris	SIFT	3 x 128
C-SIFT [58]	Gaussian	Harris-affine	SIFT	1 x 128
OpponentSIFT [39]	Opponent	SIFT	SIFT	2 x 128

4.2.2 Color Constancy

The first Section of Background Theory Chapter (Section 2.1) presented to the reader the theory of image formation by modeling images through the Lambertian Reflectance Model, Equation (2.3). As discussed, assuming that the objects exhibits Lambertian reflectance, the image RGB values are directly proportional to the light source $E(\lambda, X_{obj})$. Consequently, the measured color values are significantly influenced by the color of the scene illuminant. In other words, the same object, taken by the same camera but under different illumination, may vary in its measured color values.

This color variation may introduce undesirable effects and negatively affect the performance of computer vision methods. For example, shading, shadows, specularities, and interreflections, as well as changes due to local variation in the intensity or color of the illumination all make it more difficult to achieve basic visual tasks such as image retrieval, image classification, image segmentation, object recognition, tracking and surveillance [221, 222, 223]. For this reason, one of the most fundamental tasks of visual systems is to distinguish the changes due to underlying imaged surfaces from those changes due to the effects of the scene illumination.

The ability to perceive color as constant under changing conditions of illumination is known as color constancy, and is a natural ability of human observers. It was demonstrated that color constant cells have been found inside the visual area V4 of the human extrastriate visual cortex [224, 225]. These cells seem to respond to the reflectance of an object irrespective of the wavelength composition of the light it reflected [226]. Although the mechanism used by the brain to achieve color constancy is not yet well understood, the brain somehow does arrive at a descriptor which is independent of the illuminant [227].

The problem of computing a color constant descriptor based only on data measured by the retinal receptors is actually underdetermined, as both the illuminant spectrum distribution $E(\lambda, X_{obj})$ and the camera sensitivity $p_k(\lambda)$ are unknown. Therefore, one need to impose some assumptions regarding the imaging conditions. The most simple and general approaches to color constancy (i.e. White Patch and the Gray World hypothesis) make use of a single statistic of the scene to estimate the illuminant, which is assumed to be uniform in the region of interest. Approaches like Gamut Mapping, on the other hand, make use of assumptions of the surface reflectance properties of the objects [222].

Gamut mapping is a color constant algorithm introduced by Forsyth [37] in the early 1990's, and later extended in several works [228, 229, 230, 231, 232, 233]. The concept of gamut mapping is based on the observation that, in real world images, only a limited number of colors can be observed under a given illuminant. Therefore, any deviation from this set of colors is related to variations in the color of the light source. This method can also be referred to as a constraint based approach, since color constancy is achieved by imposing constraints in the set of possible transformations that maps the image under the unknown illuminant to the image under the known, canonical, illuminant [234].

In general, gamut mapping algorithms consists of two phases. The first is a learning phase where the algorithm estimates the set of all possible camera responses (pixel RGB values) by observing a wide number of images under a known, reference, illuminant (canonical illuminant). Such set of all possible colors under the canonical illuminant is referred to as canonical gamut, and can be represented through a convex hull in RGB space. The second is a testing phase, responsible to estimate the illuminant of an image under an unknown light source. In this phase the algorithm takes an input image under an unknown light source and estimate its gamut, which colors also form convex set. Then, using the Diagonal model (Equation 2.24), it determines the set of all feasible transformations that, when applied to the gamut of the input image, result in the canonical gamut. Under the assumption of the Diagonal model, it should exist a unique mapping that converts the gamut of the unknown light source to the canonical gamut. However, since the gamut of the unknown light source is estimated by using the gamut of only one input image, several consistent mappings can be obtained in practice. The transformation resulting in the most colorful scene (diagonal matrix with largest trace) is then chosen between the set of possible transformations [37]. In addition to the original heuristic, Gijsenij [234] reminds that Barnard [235] presents other alternatives, like the average of the feasible set and a weighted average.

The algorithm described above corresponds to the original work presented by Forsyth in [228], and is known as "coefficient-rule", or just CRULE. The Color in Perspective algorithm (CiP)[228] demonstrates not only that the gamut mapping algorithm can be computed in the chromaticity space $(R/B, G/B)$, but also that the diagonal maps can be further constrained to correspond to the expected illuminants. The Gamut Constrained Illumination Estimation (GCIE) [233] demonstrates that the gamut mapping algorithm is improved by considering only transformations which correspond to existing illuminants. In [232] the Cubical Gamut Mapping (CGM) propose a simpler version of the gamut mapping, representing the gamut of image chromaticities as a cube characterized by the image's maximum and minimum RGB chromaticities, rather than the original more complicated convex hull. In [231] Gijsenij *et al.* discusses that since gamut based algorithms use only the pixel values to estimate the illuminant, additional information present in higher-order structures is ignored. Thus, they extend the gamut mapping to incorporate image derivatives, which has the advantage to be invariant to disturbing effects such as saturated colors and diffuse light.

White Patch is the hypothesis of several algorithms derived from Land's Retinex theory [236]. Algorithms based on the White Patch are inspired by the eye biological mechanisms to adapt itself to poor illumination conditions. In some conditions, the human visual system normalizes its channel values, maximizing towards an hypothetical white reference area.

It supports a simple and fast color constancy algorithm known as max-RGB algorithm [234], also referred in the literature as scale by max algorithm (SBM) [237]. This algorithm estimates the color of the light source based on the observation that a surface with perfect reflectance properties reflects the full range of incident light. Thus, assuming that a White Patch presents these perfect reflectance properties, it is possible the estimate the scene illuminant by measuring the maximum of the responses in the RGB channels [223]. In practice, the assumption of perfect reflectance is alleviated by considering the color channels separately. Since the maxima of the separate channels is not required to be on the same location, it can also correctly estimate the illuminant when the maximum reflectance is equal for the three channels.

$$maxf(x) = (maxR(x), maxG(x), maxB(x)) \tag{4.1}$$

Thus, the color invariant max-RGB descriptor can be obtained through

$$O(x) = \frac{f(x)}{maxf(x)} \tag{4.2}$$

The traditional White Patch models the eye adaptation mechanism considering a white global reference [227]. Provenzi *et al.* [238], on the other hand, proposed the use of a local reference white. They refer that a local reference white, which is equivalent to a locally biased adaptation, would allow a more effective tone reproduction and mimic the way the human visual system extracts useful information from the light areas in backlight situations.

Another well known and simple color constancy method is the Gray World hypothesis. In the literature, the Gray World has been proposed in a variety of forms by a number of different authors, from the seminal work of Buchsbaum [239] in the early 1980s, until newer extensions like those proposed by Ebner [226, 240] in the late 2000s. In physical terms, this method assumes that the average reflectance of the surfaces in a scene is achromatic, i.e. in average the world is gray. Considering this assumption, any deviation from achromaticity in the average of the scene color is related to the color illuminant.

One traditional method to estimate the scene illuminant is to firstly find the average intensity of the image's R, G, and B color components, and use them to determine a common gray value $\bar{g}$ for the image, Equation (4.4). Each color component is then scaled according to it's deviation from this gray value. The scale factors α_k can be computed dividing the gray value by the appropriate average of each color component, Equation (4.5). By forcing the Gray World assumption on the image we are in essence removing the colored lighting effect of the scene illuminant, and thus restoring the true colors of the image.

$$avrI_k = \frac{1}{N}\sum_{x=0}^{N} I_k(x) \tag{4.3}$$

$$\bar{g} = \frac{1}{3}(avrI_R + avrI_G + avrI_B) \tag{4.4}$$

$$\alpha_k = \frac{\bar{g}}{avrI_k} \tag{4.5}$$

$$Igw_k = \alpha_k\, I_k \tag{4.6}$$

where $k \in \{R, G, B\}$, $avrI_k$ is the average intensity in each one of the three RGB channels, $\bar{g}$ is the common average gray value for the R, G and B components and Igw is the final image under the Gray World assumption. The Gray World approach works Similar to the exposure control of digital cameras, which centres the image histogram dynamically. Since this happens independently on the three RGB chromatic channels, any eventual global chromatic dominance is eliminated.

Buchsbaum used the Gray World together with a description of lights and surfaces to derive an algorithm to recover the scene illuminant $E(\lambda,x)$ and the surface reflectance functions $S(\lambda,x)$. Gershon *et al.* [241] showed that the spatial average computed in the Equation (4.3) is biased towards surfaces of large spatial extent. In order to alleviate this problem they proposed to segment the image into patches of uniform color prior to estimating the illuminant. This way, it is possible to account the response of each surface only once, and thus guarantee an equal weight to surfaces of any size during the illuminant estimation stage. In [223], Weijer *et al.* discuss the potential benefits of considering local averaging instead of the original global average proposition. This work demonstrates that the global averaging operation ignores important local correlation between pixels, which can be used to reduce the influence of noise. To exploit this local correlation, they make use of a local smoothing preprocessing steps, which was proven to be beneficial for color constancy algorithms [67].

A more recent method is based on the Local Space Average Color (LSAC), which can be defined as a computational model of how the human visual system performs averaging of image pixels. The theory is proposed by Ebner and discussed in a number of his publications [226, 227, 240, 242]. The model proposed by Ebner makes two important assumptions. The first, supported by the research of Zeki and Marini [243] about the cells found in V4, is that the essential processing required to compute a color constant descriptor in human observers is located in V4. The second, supported by the work of Herault [244], is that gap junctions behave like resistors. Thus, Ebner models the gap junctions between neurons in V4 as a resistive grid, which can be used to compute Local Space Average Color, and then color constant descriptors. Each neuron of this resistive grid computes the local space average color $a(x,y)$ by iterating the update equations (4.7) and (4.8) indefinitely for all three bands.

$$a'(x,y) = \frac{1}{|N(x,y)|} \sum_{(x',y')\in N(x,y)} a(x',y') \tag{4.7}$$

$$a(x,y) = I(x,y)\ p + a'(x,y)\ (1-p) \tag{4.8}$$

where $a(x,y) = [a_R(x,y), a_G(x,y), a_B(x,y)]$ and $N(x,y)$ is the set of the nearest neighbouring neurons. The parameter p is a small percentage that determines the extent over which LSAC is computed. Since p is inversely proportional to the area size, for small values of p LSAC is computed over large areas, while for large values of p LSAC is computed over a small neighbourhood.

According to Ebner, the iterative computation of Local Space Average Color produce results which are similar to the convolution of the input image with a smoothing kernel [226]. Approximating LSAC with a Gaussian kernel we have

$$a(x,y) = \int\int I(x',y') \frac{1}{2\pi\sigma^2} \exp^{-\frac{(x-x')^2+(y-y')^2}{\sigma^2}} \tag{4.9}$$

while with an exponential kernel

$$a(x,y) = \int\int I(x',y') \exp\frac{|x-x'|+|y-y'|}{\sigma} \tag{4.10}$$

The correspondence between the smoothing factor σ and the parameter p is given by

$$\sigma = \sqrt{\frac{1-p}{4p}} \tag{4.11}$$

The Local Space Average color alone is just a biologically inspired theory that tries to explain how the brain averages image pixels. However, when this theory is combined with others it can provide means to achieve color invariant descriptors. Below we summarize two different methodologies that Ebner demonstrated can be combined to LSAC to derive such descriptors. The first is the integration of the LSAC into the Gray World hypothesis, while the second integrates LSAC, the Gray World hypothesis and the color Shifts theory.

Ebner uses two assumptions to estimate the scene illuminant based on the Gray World [226, 240]. The first is that the illumination is constant over the entire image $(E_k(x,y) = E_k)$. For the sake of simplicity and without loss of generality, instead of former $S(\lambda_k, x_{obj})$ and $E(\lambda_k, x_{obj})$ notation, we are going to use $S_k(x,y)$ and $E_k(x,y)$ respectively to define the surface reflectance and the illuminant of wavelength λ_k in the location (x,y). Through this premise the Equation (2.5) can be re-written as:

$$I_k(x,y) = S_k(x,y)E_k \tag{4.12}$$

Using the above derivation, the global space average color of an image with n pixels is given by:

$$\begin{aligned} a_k &= \frac{1}{n}\sum I_k(x,y) \\ &= \frac{1}{n}\sum S_k(x,y)E_k \\ &= E_k\frac{1}{n}\sum S_k(x,y) \end{aligned} \tag{4.13}$$

The second assumption is that all colors are likely to occur in the scene. At first, this assumption sounds too strong. But if we assume that several colored objects are present in the scene, and that no other information about these objects is available, it is reasonable to consider that the colors of these objects are uniformly distributed over the entire color range. An attempt reader should note that such premise is not valid for all scenes, but only when the scene contains a sufficient number of different colors. When that is true, and reflectances are distributed between $[0,1]$, the expected average reflectance of a sufficient large number of pixels is close to:

$$\frac{1}{n}\sum S_k(x,y) \approx \frac{1}{2} \tag{4.14}$$

Considering a global scene illuminant and uniform reflectance, one can replace the Equation (4.14) in (4.13), and express the illuminant in function of the global space average color:

$$a_k = E_k \frac{1}{2} \quad \therefore \quad E_k = 2a_k \tag{4.15}$$

The advantage of Ebner's work is that if we consider the Gray World assumption in a local perspective, it is possible to estimate the color of the illuminant at each image pixel.

$$E_k(x,y) \approx 2a_k(x,y) \tag{4.16}$$

Thus, one can derive a local color invariant descriptor O_k by dividing the intensity I_k of each pixel by twice the local space average color:

$$O_k(x,y) = \frac{I_k(x,y)}{2a_k(x,y)} \approx \frac{I_k(x,y)}{E_k(x,y)} \approx \frac{S_k(x,y)E_k(x,y)}{E_k(x,y)} \approx S_k(x,y) \tag{4.17}$$

As we have seen in the previous above, one possible constant color descriptor is obtained by dividing each image pixel by twice the space average color. Ebner [226], though, discusses a second method to obtain a color constant descriptor. This method is based on the research of Helson [245], which indicates that human observers appear to use color shifts to estimate the color of achromatic samples illuminated by colored light.

The Gray World hypothesis assumes that the average color of image pixels should be gray. In other words, the average color should be located on the vector that corresponds to the main diagonal of the RGB cube, Figure 2.3. If the average color is not located on the gray vector then it has to be corrected such that the gray world assumption is fulfilled. Ebner's idea to correct the average color is to shift the Local Space Average color onto the gray vector. Consider $a = [a_R,\ a_G,\ a_B]^T$ the LSAC in a given pixel located in (x,y), $w = \frac{1}{\sqrt{3}}[1,\ 1,\ 1]^T$ the normalized gray vector and $I = [I_R,\ I_G,\ I_B]^T$ the color of the current pixel. The first step is to compute the projection of the vector a into the white vector w, Equation (4.18). The second step is to compute the component a_p of the LSAC that is perpendicular to the gray vector, Equation (4.19). The perpendicular component is then subtracted from the color of the current pixel, resulting in the color constant descriptor O.

$$a' = (a\, w)\, w \tag{4.18}$$

$$a_p = a - a' \tag{4.19}$$

$$O = I - a_p \tag{4.20}$$

Considering $k \in \{R,G,B\}$, one can express the color descriptor O_k at individual color channels as:

$$O_k = I_k - a_k + \frac{1}{3}(a_R + a_G + a_B) \tag{4.21}$$

4.3 Effects of Photometric Variations

Not all color spaces are suitable for digital image processing. For example, one problem of the CIE RGB and the sRGB spaces in feature tracking is that the algorithm needs to be performed in a 3-D space, thus increasing the computational effort. Furthermore, the RGB space is not invariant to lighting changes because the intensity channel is a combination of the R, G and B channels. Therefore, all the gain in information provided by the colorimetric dimensions can be useless since varying light conditions affect the colors observed. In fact, photometric invariance is less trivial to achieve, but it is of the utmost importance when dealing with problems such as changes in the color and direction of the illuminator, and changes in the camera viewpoint.

To understand the effects of the light source variation in filter responses consider an observed single channel image I_o with pixel intensity $I_o(x,y)$ at a given point $X = (x,y)$. As we have demonstrated in the Section 2.5, through the central difference method it is possible to express the second derivatives of $I_o(x,y)$ as:

$$\frac{\partial^2 I_o(x,y)}{\partial x^2} = I_o(x+1,y) - 2I_o(x,y) + I_o(x-1,y) \tag{4.22}$$

Now, consider that I_o has a corresponding image I_u, taken under unknown illuminant. Assuming the Diagonal-offset model (2.26) these two images are related by a linear transformation determined by a scalar constant α and an offset β. Therefore, the pixel intensity $I_u(x,y)$ of the image I_u at the same point $X = (x,y)$ can be modeled as:

$$I_u(x,y) = \alpha I_o(x,y) + \beta \tag{4.23}$$

Thus, it is possible to conclude that the first derivative of $I_u(x,y)$ with respect to x and y is respectively

$$\frac{\partial I_u(x,y)}{\partial x} \approx \alpha I_o(x+1,y) + \beta - \alpha I_o(x-1,y) + \beta = \alpha \, \frac{\partial I_o(x,y)}{\partial x} \tag{4.24}$$

$$\frac{\partial I_u(x,y)}{\partial y} \approx \alpha I_o(x,y+1) + \beta - \alpha I_o(x,y-1) + \beta = \alpha \, \frac{\partial I_o(x,y)}{\partial y} \tag{4.25}$$

It is also possible to infer that the second derivative of $I_u(x,y)$ with respect to x, y and xy is respectively

$$\begin{aligned} \frac{\partial^2 I_u(x,y)}{\partial x^2} &\approx \alpha I_o(x+1,y) + \beta - 2\left(\alpha I_o(x,y) + \beta\right) + \alpha I_o(x-1,y) + \beta \\ &= \alpha \, \frac{\partial^2 I_o(x,y)}{\partial x^2} \end{aligned} \tag{4.26}$$

$$\frac{\partial^2 I_u(x,y)}{\partial y^2} = \alpha \, \frac{\partial^2 I_o(x,y)}{\partial y^2} \tag{4.27}$$

$$\frac{\partial^2 I_u(x,y)}{\partial xy} = \alpha \, \frac{\partial^2 I_o(x,y)}{\partial xy} \tag{4.28}$$

When computing the derivatives, the diffuse term β is canceled out, causing no impact on the final outcome. However, by varying the illumination with a scalar α, both the first and the second derivatives vary proportionally with the scalar.

These assumptions, though, are only valid when image pixels are represented using a sufficient large number of bits per pixel. Lets consider the intensity $I_o(x,y) = 58$ of a pixel located at (x,y) at an observed image I_o. Consider that I_o has a corresponding image I_u, of the same scene, taken under a different illumination. Suppose that the illumination varied according to the Diagonal-offset model, due to the linear transformation of a scalar term $\alpha = 1.2$ and an offset term $\beta = 5$. Through 4.23 one can approximate the intensity of I_u at (x,y) as $I_u(x,y) = 74.6$. Since image pixels are usually represented using 8 bits ($2^8 = 256$ colors), $I_u(x,y)$ has to be approximated to the nearest integer, which gives us $I_u(x,y) = 75$. Now, consider a neighboring pixel $I_o(x+1,y) = 57$. Assuming the same transformation as before, one could estimate the intensity of I_u at $(x+1,y)$ as $I_u(x+1,y) = 73.4$, witch would gives us $I_u(x,y) = 73$ when represented in an 8-bit image. From the examples above one can note that the rounding operation modifies the intensity independently for each image pixel. In the first case, the side effect was the addition 0.4 to the pixel intensity, which accounts for a final $\beta = 5.4$. In the second case, the side effect was the subtraction of 0.4, accounting for a final $\beta = 4.6$. In other words, the offset term β is in reality not constant over the image, and thus the image derivative does not completely cancel the effect of the offset.

In Chapter 2, Section 2.2, the reader was introduced to the concept of color spaces and familiarized with the mathematical models used to map from an image representation to another. Given most feature detection algorithms use luminous intensities as the main input, and that most cameras senses the environment through three spectral components, it is usual to convert images from RGB to grayscale. For this reason we now demonstrate the effects of illumination variations in the derivatives of the luminance of RGB images. Consider an observed three layer RGB image $I_o(x,y,c)$. From the Equation (2.8), we can express the luminance $Y_o(x,y)$ of this image as the weighted sum of its color channels:

$$Y_o(x,y) = \sum_c w_c I_o(x,y,c) \tag{4.29}$$

where $c \in \{R,G,B\}$, and w_c is the weight of the corresponding color channel. Now, consider that I_o has a corresponding image I_u, taken under unknown illuminant. Assuming the Diagonal-offset model (2.26) these two images are related by a linear transformation that affects independently each color channel and that is determined by a scalar constant α_c and an offset β_c. Therefore, the luminance $Y_u(x,y)$ of $I_u(x,y)$ can be modeled with respect to $I_o(x,y,c)$:

$$\begin{aligned} Y_u(x,y) =& w_R(\alpha_R I_o(x,y,R)+\beta_R)+w_G(\alpha_G I_o(x,y,G)+\beta_G) \\ &+w_B(\alpha_B I_o(x,y,B)+\beta_B) \end{aligned} \tag{4.30}$$

Since each color channel of the RGB image can be treated as an independent single channel image, it is possible to take advantage of the logic developed in the Equations (4.24) and (4.25) to express the first derivative of $Y_u(x,y)$ with respect to x and y respectively as:

$$\frac{\partial Y_u(x,y)}{\partial x} = \alpha_R w_R \frac{\partial I_o(x,y,R)}{\partial x} + \alpha_G w_G \frac{\partial I_o(x,y,G)}{\partial x} + \alpha_B w_B \frac{\partial I_o(x,y,B)}{\partial x} \tag{4.31}$$

$$\frac{\partial Y_u(x,y)}{\partial y} = \alpha_R w_R \frac{\partial I_o(x,y,R)}{\partial y} + \alpha_G w_G \frac{\partial I_o(x,y,G)}{\partial y} + \alpha_B w_B \frac{\partial I_o(x,y,B)}{\partial y} \tag{4.32}$$

With the same mechanism, it is possible to express the second derivative of $Y_u(x,y)$ with respect to x, y and xy respectively as:

$$\frac{\partial^2 Y_u(x,y)}{\partial x^2} = \alpha_R w_R \frac{\partial^2 I_o(x,y,R)}{\partial x^2} + \alpha_G w_G \frac{\partial^2 I_o(x,y,G)}{\partial x^2} + \alpha_B w_B \frac{\partial^2 I_o(x,y,B)}{\partial x^2} \tag{4.33}$$

$$\frac{\partial^2 Y_u(x,y)}{\partial y^2} = \alpha_R w_R \frac{\partial^2 I_o(x,y,R)}{\partial y^2} + \alpha_G w_G \frac{\partial^2 I_o(x,y,G)}{\partial y^2} + \alpha_B w_B \frac{\partial^2 I_o(x,y,B)}{\partial y^2} \tag{4.34}$$

$$\frac{\partial^2 Y_u(x,y)}{\partial xy} = \alpha_R w_R \frac{\partial^2 I_o(x,y,R)}{\partial xy} + \alpha_G w_G \frac{\partial^2 I_o(x,y,G)}{\partial xy} + \alpha_B w_B \frac{\partial^2 I_o(x,y,B)}{\partial xy} \tag{4.35}$$

The Equations (4.31) to (4.35) illustrate the derivatives of images subjected to transformations like the light color change and the light color change and shift (Section 2.3), which present non-uniform scalars over the three color components $(\alpha_R \neq \alpha_G \neq \alpha_B)$. Since each scalar has a different value and affect individually one of the three terms that describe the derivatives of $Y_u(x,y)$, it is not possible to rearrange or simplify the equation in order to combine and isolate the effect of the scalars over the $Y_u(x,y)$ derivative. Consequently, it is not possible to achieve full photometric invariance through arithmetic operations when performing feature detection in the luminance of RGB images.

A simple but yet effective way of increasing the robustness of the features detected in sequences of images consists in performing the detection operation independently over each color channel of the RGB image. An additional advantage of such approach is the increase in the number of features, since detection occurs in three "images", instead of in only one.

As previously discussed, feature localization is a three-step process that starts disregarding points in which blob-response is lower than a fixed threshold value. If, however, the detector responses vary with the illumination, a given feature that is detected in a bright image may not be detected in a corresponding image with lower illumination levels. For this reason, the impact of illumination changes in the response of the three most popular feature detectors will be analyzed in the following subsections. Note that, for reasons discussed above, the further analysis will consider that images are single channel, or that the detection occurs in each color channel independently.

4.3.1 Effects of Photometric Variations in Harris Corners Responses

As detailed in the Section 2.8.1, the Harris Corner algorithm detects image features when both the eigenvalues of the covariance matrix are high. Through 2.53 the covariance matrix of the pixel located at (x,y), of an image I_u taken under unknown illuminant, can be expressed as

$$C_u(x,y) = \begin{bmatrix} \left(\frac{\partial I_u(x,y)}{\partial x}\right)^2 & \frac{\partial I_u(x,y)}{\partial x}\frac{\partial I_u(x,y)}{\partial y} \\ \frac{\partial I_u(x,y)}{\partial x}\frac{\partial I_u(x,y)}{\partial y} & \left(\frac{\partial I_u(x,y)}{\partial y}\right)^2 \end{bmatrix} \tag{4.36}$$

We have also discussed that since the computation of the eigenvalues is computationally expensive, Harris and Stephens [50] suggested that the detector response could be approximated using the determinant and the trace of the covariance matrix. Thus, the detector function at $I_u(x,y)$ can be obtained through

$$R_u(x,y) = det(C_u(x,y)) - k\, trace^2(C_u(x,y)) \tag{4.37}$$

The determinant and the trace of the covariance matrix $C_u(x,y)$ can be expressed respectively as

$$det(C_u(x,y)) = \left(\frac{\partial I_u(x,y)}{\partial x}\right)^2 \left(\frac{\partial I_u(x,y)}{\partial y}\right)^2 - \left(\frac{\partial I_u(x,y)}{\partial x}\frac{\partial I_u(x,y)}{\partial y}\right)^2 \tag{4.38}$$

$$trace(C_u(x,y)) = \left(\frac{\partial I_u(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I_u(x,y)}{\partial y}\right)^2 \tag{4.39}$$

Replacing the Equations (4.24) and (4.25) into (4.38), it is possible to obtain the determinant of the covariance matrix $C_u(x,y)$ with respect of $I_o(x,y)$ by

$$\begin{aligned} det(C_u(x,y)) &= \alpha^2 \left(\frac{\partial I_o(x,y)}{\partial x}\right)^2 \alpha^2 \left(\frac{\partial I_o(x,y)}{\partial y}\right)^2 - \left(\alpha^2\frac{\partial I_o(x,y)}{\partial x}\,\alpha^2\frac{\partial I_o(x,y)}{\partial y}\right)^2 \\ det(C_u(x,y)) &= \alpha^4 \left(\left(\frac{\partial I_o(x,y)}{\partial x}\right)^2 \left(\frac{\partial I_o(x,y)}{\partial y}\right)^2 - \left(\frac{\partial I_o(x,y)}{\partial x}\frac{\partial I_o(x,y)}{\partial y}\right)^2\right) \\ det(C_u(x,y)) &= \alpha^4\, det(C_o(x,y)) \end{aligned} \tag{4.40}$$

With the same mechanism, one can express the trace of $C_u(x,y)$ in function of $I_o(x,y)$ by replacing the Equations (4.24) and (4.25) into (4.39)

$$\begin{aligned}
trace(C_u(x,y)) &= \alpha^2 \left(\frac{\partial I_o(x,y)}{\partial x}\right)^2 + \alpha^2 \left(\frac{\partial I_o(x,y)}{\partial y}\right)^2 \\
trace(C_u(x,y)) &= \alpha^2 \left(\left(\frac{\partial I_o(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I_o(x,y)}{\partial y}\right)^2\right) \\
trace(C_u(x,y)) &= \alpha^2\, trace(C_o(x,y))
\end{aligned} \tag{4.41}$$

Finally, replacing the determinant (4.40) and the trace (4.41) into (4.37) we can express the Harris Corner response function R_u in terms of the pixel intensities of the observed image $I_o(x,y)$.

$$\begin{aligned}
R_u(x,y) &= \alpha^4\, det(C_o(x,y)) - k\, \left(\alpha^2\, trace(C_o(x,y))\right)^2 \\
R_u(x,y) &= \alpha^4\, det(C_o(x,y)) - \alpha^4\, k\, trace^2(C_o(x,y)) \\
R_u(x,y) &= \alpha^4\, R_o(x,y)
\end{aligned} \tag{4.42}$$

The Equation (4.42) demonstrates the correlation between the Harris corners responses R_u and R_o, given the linear transformation determined by a scalar term α and an offset term β. The degree of the polynomial (α^4) provides the theoretical explanation to why even small variations in the scene illuminant cause large variation in the magnitude of the detector response.

4.3.2 Effects of Photometric Variations in SIFT Responses

As detailed in the Section 2.8.3, the response of the SIFT detector at a given pixel located at (x,y), of an image $I_u(x,y)$ taken under unknown illuminant, is given by the trace and the determinant of the Hessian matrix $H_u(x,y)$:

$$R_u(x,y) = \frac{(trace\,(H_u(x,y)))^2}{det\,(H_u(x,y))} \tag{4.43}$$

From the Hessian matrix $H_u(x,y)$,the determinant and the trace are given respectively by:

$$det(H_u(x,y)) = \frac{\partial^2 I_u(x,y)}{\partial x^2}\frac{\partial^2 I_u(x,y)}{\partial y^2} - \left(\frac{\partial^2 I_u(x,y)}{\partial xy}\right)^2 \tag{4.44}$$

$$trace(H_u(x,y)) = \left(\frac{\partial^2 I_u(x,y)}{\partial x^2} + \frac{\partial^2 I_u(x,y)}{\partial y^2}\right) \tag{4.45}$$

Replacing the Equations (4.26), (4.27) and (4.28) in (4.44) we can express the determinant of $H_u(x,y)$ with respect to the determinant of the Hessian matrix of the observed image $I_o(x,y)$:

$$\begin{aligned}
det(H_u(x,y)) &= \alpha\frac{\partial^2 I_o(x,y)}{\partial x^2}\alpha\frac{\partial^2 I_o(x,y)}{\partial y^2} - \left(\alpha\frac{\partial^2 I_o(x,y)}{\partial xy}\right)^2 \\
det(H_u(x,y)) &= \alpha^2 \left(\frac{\partial^2 I_o(x,y)}{\partial x^2}\frac{\partial^2 I_o(x,y)}{\partial y^2} - \left(\frac{\partial^2 I_o(x,y)}{\partial xy}\right)^2\right) \\
det(H_u(x,y)) &= \alpha^2\, det(H_o(x,y))
\end{aligned} \tag{4.46}$$

A similar operation can be performed to express the trace of $H_u(x,y)$ with respect to the trace of the reference image $I_o(x,y)$:

$$\begin{aligned}
trace(H_u(x,y)) &= \alpha\,\frac{\partial^2 I_o(x,y)}{\partial x^2} + \alpha\,\frac{\partial^2 I_o(x,y)}{\partial y^2} \\
trace(H_u(x,y)) &= \alpha\,\left(\frac{\partial^2 I_o(x,y)}{\partial x^2} + \frac{\partial^2 I_o(x,y)}{\partial y^2}\right) \\
trace(H_u(x,y)) &= \alpha\, trace(H_o(x,y))
\end{aligned} \tag{4.47}$$

Finally, replacing the determinant (4.46) and the trace (4.47) into (4.43) we can express the SIFT response function R_u in terms of the pixel intensities of the observed image $I_o(x,y)$.

$$\begin{aligned}
R_u(x,y) &= \frac{\alpha^2\ (trace\,(H_o(x,y)))^2}{\alpha^2\, det(H_o(x,y))} \\
R_u(x,y) &= R_o(x,y)
\end{aligned} \tag{4.48}$$

The Equation (4.48) provides the theoretical foundations to explain the robustness of the SIFT detector to variations in scene illumination. Since both the scalar and the offset terms are canceled, SIFT responses should be invariant to all types of illumination variations in single channel images. Note that the same behavior is not observed when using the luminance of the RGB images that are subjected to transformations like the light color change and the light color change and shift.

4.3.3 Effects of Photometric Variations in SURF Responses

Using SURF for feature detection, the filter response R_u is computed through the determinant of the Hessian matrix.

$$H_u(x,y) = \begin{bmatrix} \dfrac{\partial^2 I_u(x,y)}{\partial x^2} & \dfrac{\partial^2 I_u(x,y)}{\partial xy} \\ \dfrac{\partial^2 I_u(x,y)}{\partial xy} & \dfrac{\partial^2 I_u(x,y)}{\partial y^2} \end{bmatrix} \tag{4.49}$$

$$R_u(x,y) = \frac{\partial^2 I_u(x,y)}{\partial x^2}\frac{\partial^2 I_u(x,y)}{\partial y^2} - \left(\frac{\partial^2 I_u(x,y)}{\partial xy}\right)^2 \tag{4.50}$$

Replacing the Equations (4.26) to (4.28) into (4.50), the filter response R_u can be expressed in terms of R_o

$$\begin{aligned}
R_u(x,y) &= \alpha\frac{\partial^2 I_o(x,y)}{\partial x^2}\alpha\frac{\partial^2 I_o(x,y)}{\partial y^2} - \left(\alpha\frac{\partial^2 I_o(x,y)}{\partial xy}\right)^2 \\
R_u(x,y) &= \alpha^2 R_o
\end{aligned} \tag{4.51}$$

The Equation (4.51) demonstrates the correlation between the SURF responses R_u and R_o, given the linear transformation determined by a scalar term α and an offset term β. The degree of the polynomial (α^2) provides the theoretical explanation to why variations in the scene illuminant cause significant variations in the magnitude of the detector response.

4.4 Photometric Invariance Through Local Normalization: LN SURF

As detailed in the Literature Review (Section 4.2.1), authors usually use color space mapping to deal with illumination changes. The concept is that decoupled color spaces provide invariant data, which in turn can be used to compute invariant filter responses. However, this kind of approach presents several undesirable secondary effects, such as the introduction of noise and of instabilities in pixels near the grayscale. Normalization techniques are popular in computer vision, specially for solving the variable illumination problem in face recognition context [246, 247]. The contribution of this work consists on the combination of the local normalization (LN) technique with feature detection algorithms to provide them with photometric invariance properties.

For this reason, our first approach works under a different viewpoint. Rather than trying to achieve invariance through color space mapping, the variables used to compute filter responses are normalized, thus deriving invariant feature responses over regular RGB images. To normalize the variables and eliminate the effects of the scalar noise α, we take advantage of the local normalization (LN) technique. Pixel normalization (I_z) is achieved by dividing the difference between an the pixel's intensity (I) and the mean (μ) by the standard deviation σ, equation(4.52).

$$I_z = \frac{I - \mu}{\sigma} \tag{4.52}$$

Rather than working with the mean intensity and standard deviation of the entire image, the mean and the standard deviation are computed for the set of N pixels inside the filter under analysis. This way, our approach can handle variable offsets and scale for different regions of the image. The resulting intensity value is of local zero mean and with a unit variance within the filter. Considering an observed image f_o with pixel intensity $I_o(x,y)$ at a given point $X = (x,y)$, mean μ_o and standard deviation σ_o, the normalized intensity $I_{oz}(x,y)$ can be expressed as follows:

$$\mu_o = \frac{1}{N}\sum_{i=0}^{N} I_o(x,y) \tag{4.53}$$

$$\sigma_o = \sqrt{\frac{1}{N}\sum_{i=0}^{N} (I_o(x,y) - \mu_o)^2} \tag{4.54}$$

$$I_{oz}(x,y) = \frac{I_o(x,y) - \mu_o}{\sigma_o} \tag{4.55}$$

The effects of the variation caused by a scalar α and by an offset β in the mean (μ_u) and in the standard deviation (σ_u) of the corresponding image f_u, can be described as:

$$\mu_u = \frac{1}{N}\sum_{i=0}^{N} (\alpha I_o(x,y) + \beta) = \alpha\mu_o + \beta \tag{4.56}$$

$$\sigma_u = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\alpha I_o(x,y)+\beta-(\alpha\mu_o+\beta))^2} = \alpha\sigma_o \tag{4.57}$$

Therefore, one can express the normalized intensity I_{uz} in function of I_{oz}

$$I_{uz}(x,y) = \frac{(\alpha I_o(x,y)+\beta)-(\alpha\mu_o+\beta)}{\alpha\sigma_o} \tag{4.58}$$

$$= \frac{\alpha(I_o(x,y)-\mu_o)}{\alpha\sigma_o} \tag{4.59}$$

$$= \frac{I_o(x,y)-\mu_o}{\sigma_o} \tag{4.60}$$

$$= I_{oz}(x,y) \tag{4.61}$$

Computing the second derivative of $I_z(x,y)$ with respect to x gives:

$$\frac{d^2I_z(x,y)}{dx^2} = \frac{I_z(x+1,y)-2I_z(x,y)+I_z(x-1,y)}{\sigma} \tag{4.62}$$

The same applies for the second derivative in y and xy. Once the derivatives of z-scored values are independent of the scalar α and the offset β, the response values for both images are equal:

$$R_u = R_o \tag{4.63}$$

Computing the standard deviation through its definition is a very demanding task, especially because this operation is performed each time the filter response is computed. A more efficient way to compute the standard deviation is given through the Equation (4.64), which reduces the necessary number of operations from $(4N+2)$ to $(3N+4)$, and the number of memory access from $(3N+1)$ to $(2N+2)$.

$$\sigma = \sqrt{\frac{1}{N}\left(\sum_{i=0}^{N} I(x,y)^2\right) - \mu^2} \tag{4.64}$$

Indeed, despite the reduction in the number of operations, the computational cost necessary to estimate the standard deviation through is still very high. Note that the number of required operations grows fast with the size of the filter, i.e. becoming as high as 885 for $l = 21$ and 19605 for $l = 99$. For this reason, we propose a much more efficient way to estimate the standard deviation, combining (4.64) with the concept of integral images.

$$I^2_{\Sigma}(x,y) = \sum_{i=y-h}^{i<y+h}\sum_{j=x-h}^{j<x+h} I(x,y)^2 \tag{4.65}$$

By computing beforehand the sum of the squared pixel intensity for the entire image, it is possible to compute the standard deviation of any filter with only 5 operations and 4 memory access.

Algorithm 1 LN SURF feature detector

Definitions:

rgbImg: 3 layer matrix containing the red, green and blue values of image pixels

features: structure containing the 2D location (camera reference frame), description, and response of salient image regions

```
 1: function LNSurf($rgbImg$)
 2:     $grayImg \leftarrow rgb2gray(rgbImg)$
 3:     $nCols \leftarrow rgbImg.cols$
 4:     $nRows \leftarrow rgbImg.rows$
 5:     $intImg \leftarrow gray2int(grayImg)$
 6:     $intSqrImg \leftarrow gray2intSqr(grayImg)$

 7:     for $scale = 1$ to $nScales$ do
 8:         $size = scale2filterSize(scale)$
 9:         for $row = 1$ to $nRows$ do
10:             for $col = 1$ to $nCols$ do
11:                 $surfResp = DxxDyy - 0.81Dxy^2$
12:                 $\mu = sumPixels(intImg, row, col, size)/size^2$
13:                 $\sigma^2 = sumPixels(intSqrImg, row, col, size)/size^2 - \mu^2$
14:                 $response[scale, row, col] = surfResp/\sigma^2$    ▷ Normalized SURF response
15:             end for
16:         end for
17:     end for

18:     $features \leftarrow extremum(response)$    ▷ Non-maximal suppression

19:     $features \leftarrow describeFeatures(intImg, features)$

20:     return $features$
21: end function
```

Just like the classic integral image, computing the squared integral image has some computational cost, but this initial cost is paid off after just a few filter convolutions.

The pseudo-code for the LN SURF algorithm is detailed in the Algorithm 1. The system first convert the rgb image to gray scale (line 2). Next, it computes the corresponding integral and integral squared images (lines 5 and 6). For each pixel and scale, it is computed the SURF response, mean and standard deviation of the pixel intensities inside the filter (lines 11, 12 and 13). Then it is computed the locally normalized SURF response (line 14). Finally, the algorithm performs the usual SURF operations of non-maximal suppression (to detect salient image regions), interpolation in scale and space (for sub-pixel accuracy), and feature description.

4.5 Photometric Invariance Through Color Constancy: LSAC SURF

Among color constancy methods, gamut mapping is referred in literature as one of the most successful algorithms [228, 223, 248, 231]. It has demonstrated good results in different datasets of several works. The method is though computationally quite complex. Its implementation requires the computation of two convex hulls, which is a difficult problem when using finite precision arithmetic. Another drawback is that the algorithm requires an image data set with known light sources. As previously discussed, only through this dataset the algorithm is able to estimate the canonical gamut (learning phase) that will be used to compute the transformation matrix, and thus estimate the illuminant (testing phase). In practice, such methodology is not viable for robotic vision systems since robots are not constrained to one specific scenario, but subjected to multiple and dynamic environments.

Low level color constant algorithms, on the other hand, are less complex, faster and only slightly outperformed by the gamut mapping [249, 248, 67]. These characteristics make them perfect candidates for improving robotic vision systems. One limitation of the Gray World assumption is that it is only valid in images with sufficient amount of color variations. Only when the variations in color are random and independent, the average value of the R, G, and B components of the image would converge to a common gray value. This assumption is, however, held very well in several real world scenarios, where it is usually true that there are a lot of different color variations.

Another limitation of most color constancy algorithms is that they are modeled with the assumption that the scene is uniformly illuminated. Since in practice multiple illuminants are present in the scene, the illumination is not uniform, and thus the premise is not fully verified. For instance, some daylight may be falling through a window while an artificial illuminant may be switched on inside the room. In fact, that may be the main advantage of the descriptors derived from the Local Space Average color methodology. Since LSAC estimates the illuminant locally for each point of the scene, its descriptors are better prepared to deliver color constancy in real world images.

Most of the proposed color invariant feature detectors combine the original detector with some sort of color space mapping. Our approach to achieve photometric invariant feature responses, LSAC SURF, consists on taking advantage of the invariant properties of the LSAC descriptor, using it as working space for feature detection. The inclusion of this pre-processing step adds a small computational load, but may provide a significant increase in feature detection robustness.

The size of the window that LSAC is computed plays an important role in the robustness of the feature detection. Empirical observation demonstrated that feature repeatability tends to perform better when LSAC is computed over small neighborhoods. In fact, due to the multiple illumination sources the values of α and β tends to vary significantly in distant image pixels, which makes the assumption that $E_k(x,y) \approx 2a_k(x,y)$, Equation (4.16), to be valid only for small regions.

Algorithm 2 LSAC SURF feature detector

Definitions:

rgbImg: 3 layer matrix containing the red, green and blue values of image pixels
features: structure containing the 2D location, description and response of visual features

```
function LSACSurf($rgbImg$)
    $nCols \leftarrow rgbImg.cols$
    $nRows \leftarrow rgbImg.rows$
    $grayImg \leftarrow rgb2gray(rgbImg)$
    $intImg \leftarrow gray2int(grayImg)$

    for $row = 1$ to $nRows$ do
        for $col = 1$ to $nCols$ do
            $a_k = gauss(intImg, filterSize)$                    ▷ LSAC Gaussian approximation
            $lsacDesc[row, col] = grayImg[row, col]/(2\ a_k)$              ▷ LSAC descriptor
        end for
    end for

    $features \leftarrow SURF(lsacDesc)$            ▷ SURF detection over the LSAC descriptor

    for $i = 1 \rightarrow nFeatures$ do
        $px \leftarrow grayImg[features[i].row, features[i].col]$
        if $lowerTh > px > upperTh$ then
            $hardFeatures.add(features[i])$
        else if $0 > px > lowerTh$ or $upperTh > px > 254$ then
            $softFeatures.add(features[i])$
        end if
    end for
    return $softFeatures$, $hardFeatures$
end function
```

When a pixel reaches saturation, it does not present the same variation as its neighbors, causing non linear variations in the response of the feature detector and decreasing the probability to be correctly matched in subsequent images. Therefore, features which pixel intensities are close to saturation are not good matching purposes. However, such features can not simple be ignored since under certain illumination variations their pixel intensity can move away from saturation, and make them good candidates for matching in subsequent images. For this reason, each detected feature is classified into hard and soft features according to their pixel intensities. If the pixel intensity of a distinct image region is lower than an upper threshold and higher than a lower threshold the feature is classified as hard feature, on the contrary, the feature is classified as soft feature. The choice of the proper upper and lower threshold values might be determined according to the expected variation in the scene illumination.

Since hard features are more likely to be found in subsequent images, we can reduce the search space and match only the current hard features with the subsequent set of features. In this context, soft features are used only to support matching of previous hard features, while hard features are used in in the computation of sensitive visual tasks.

The pseudo-code for the LSAC SURF algorithm is detailed in the Algorithm 2. First, the system convert the rgb image to gray scale (line 4). Next, it computes its corresponding integral image (line 5). For each image pixel, the algorithm approximates the local space average color with a Gaussian kernel (line 8). In order to speed up the algorithm, the convolution with the Gaussian kernel is approximated using box type filters and integral images. Later, it computes the local space average color descriptor as the ratio between the image pixel intensity and the local average color (line 9). Following, the algorithm performs the usual SURF feature detection and description operations using the LSAC descriptor as working space (line 12). Finally, the algorithm classifies the detected features into hard or soft features according to an upper and a lower thresholds (lines 13 to 22).

4.6 Experiments and Results

To evaluate the performance of our approach we adopted the repeatability criterion similar to the proposed by Schmid *et al.* [250]. The repeatability rate evaluates the ratio between the number of point-to-point correspondences that can be established for detected points in all images of the same scene $C(I_1,...,I_n)$ and the total number of features detected in the current image m_i, as described the following relation:

$$R_i = \frac{C(I_1,I_2,...,I_n)}{m_i} \tag{4.66}$$

where R_i denotes the repeatability rate of the image under analysis, $C(I_1,I_2,...,I_n)$ the number of corresponding features, n the number of images of the same scene, and m_i the number of features detected in i. Therefore, the higher the repeatability, the more likely features are to be matched and the better the matching results tend to be.

The repeatability rate of our approaches are compared with the repeatability rate of the SURF algorithm available in the OpenCV library. For the comparison to be fair, the optimum values were assigned to the SURF parameters, as described in [55], and the threshold of the LN SURF was adjusted to match the number of features detected by the original algorithm.

The next experiments compare the repeatability rates of the original SURF algorithm and the two proposed extensions. Four images collections provide shift, scale, color variations, that allows us to asses the performance of the algorithms in the most common illumination transformations. In addition, two other image collections provide small and large photometric changes in real world scenarios. For each image collection, both algorithms are applied and their repeatability rate computed. Finally, through a hypothesis test of matched pairs, a statistical analysis is performed to determine if there was a significant improvement in the feature repeatability.

Since several hypothesis tests assume that data is normally distributed, a first step to choose the most appropriate test consists on determining whether or not the observations can be modeled by a normal distribution. For this purpose, we performed an exploratory data analysis (EDA) and normality tests. The EDA provides clues (mean, median, skewness and kurtosis) about the normality of the distribution, and is important to summarize and understand the data set main characteristics. A first indication about the normality of the observation is given by the values of the median and the mean. Since normality implies symmetry around an inflection point, the mean, median, and mode are expected to be all the same and coincident with the peak of the curve. Another indication of symmetry is given by the skewness of the distribution. The distribution is considered different from the normal to a significant degree when the absolute value of skewness is more than twice the standard error of skewness. The kurtosis characterizes the relative peakedness or flatness (relative concentration of rates) of a distribution compared to the normal distribution. Like the skewness, the distribution is considered different from the normal to a significant degree when the absolute value of kurtosis is more than twice the standard error of kurtosis.

The normality test is used to verify the conclusions inferred from the EAD, in which samples are standardized and compared with a standard normal distribution. Here, the size of the sample indicates the most appropriate test. Observations performed in image collections with more than 50 images will be tested with the Kolmogorov-Smirnov test, and with less than 50 images will be tested with the Shapiro–Wilk test. Considering a 95% confidence interval, the distribution is not considered normally distributed when the p-value is lower than $\alpha = 0.05$ (i.e. the null hypothesis is rejected).

Finally, a statistical hypothesis test is used to verify if there is significant difference in the repeatability rate of the original and the proposed algorithms. As mentioned above, the most appropriate test depends on the nature of the observations. Normally distributed observations will be evaluated through the paired Student's t-test, while non-normally distributed observations will be evaluated through the Wilcoxon signed rank test. Like the normality tests, a 0.05 significance level is used to verify if the the data rejects $(p - value < 0.05)$ or fail to reject the null hypothesis.

4.6.1 Controlled Image Set

The purpose of the controlled image set is to provide images with known illumination variations to assess the robustness of the feature detection algorithms. The dataset is composed of four collections, containing 9,000 images of 1,000 objects each, under three categories of the most common changes in the illumination values: LIC, LIS and LCC, Equations (2.29),(2.30) and (2.25) respectively. The LICS and LCCS category were not included because they are only a combination of the other three categories.

Images were taken of the Amsterdam Library of Object Images (ALOI) [251]. ALOI is a color image dataset of one thousand small objects recorded under different viewing angles, illumination angles and illumination colors with a total of 110,250 images. From the desired categories of illumination variation, ALOI could only provide images to evaluate features in the presence of light color change. In the ALOI illumination color collection (ALCC), the color of the illumination source varies from yellow to white according to the voltage v_0 of the lamps, where $v_0 = 12i/255$ volts and $i \in \{110, 120, 130, 140, 150, 160, 170, 180, 190\}$ (Figure 4.1).

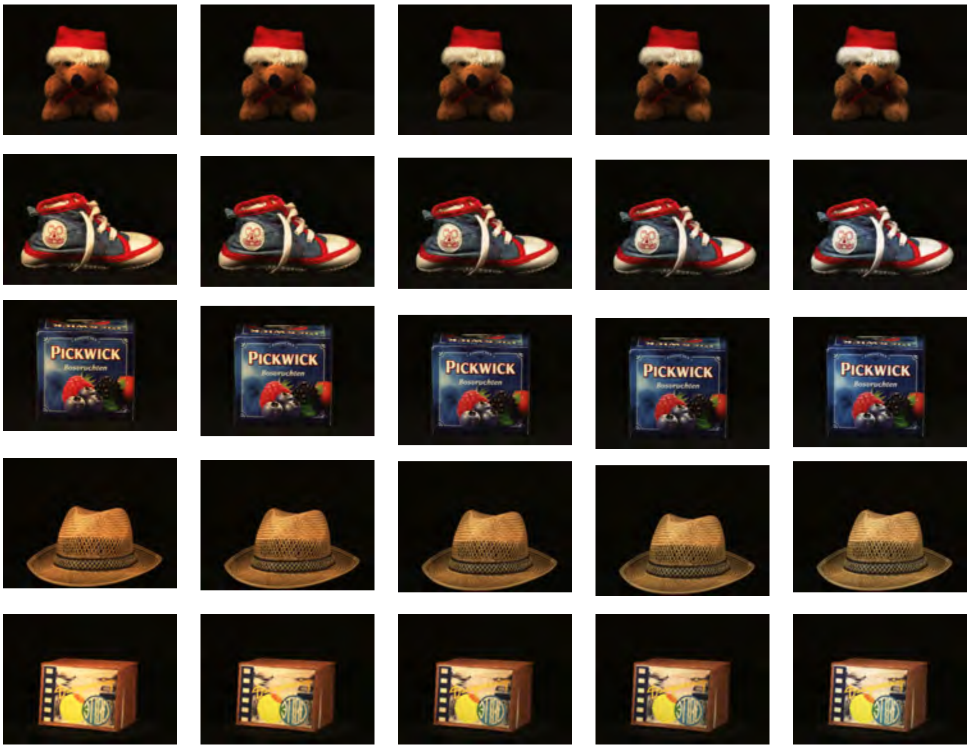


Figure 4.1: Samples from the ALCC collection. From left to right: the value of i is respectively 120, 140, 160,180, 210.

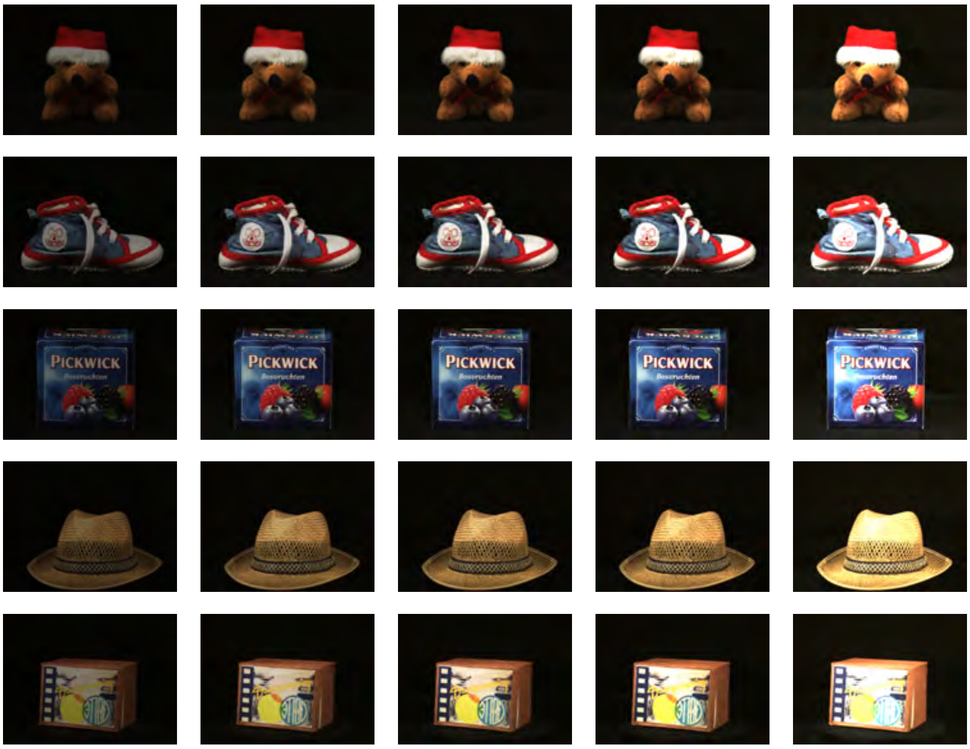


Figure 4.2: Samples from the LIC collection. From left to right: the value of α is respectively 1/1.20, 1/1.2, 1, 1.2, 2.0.

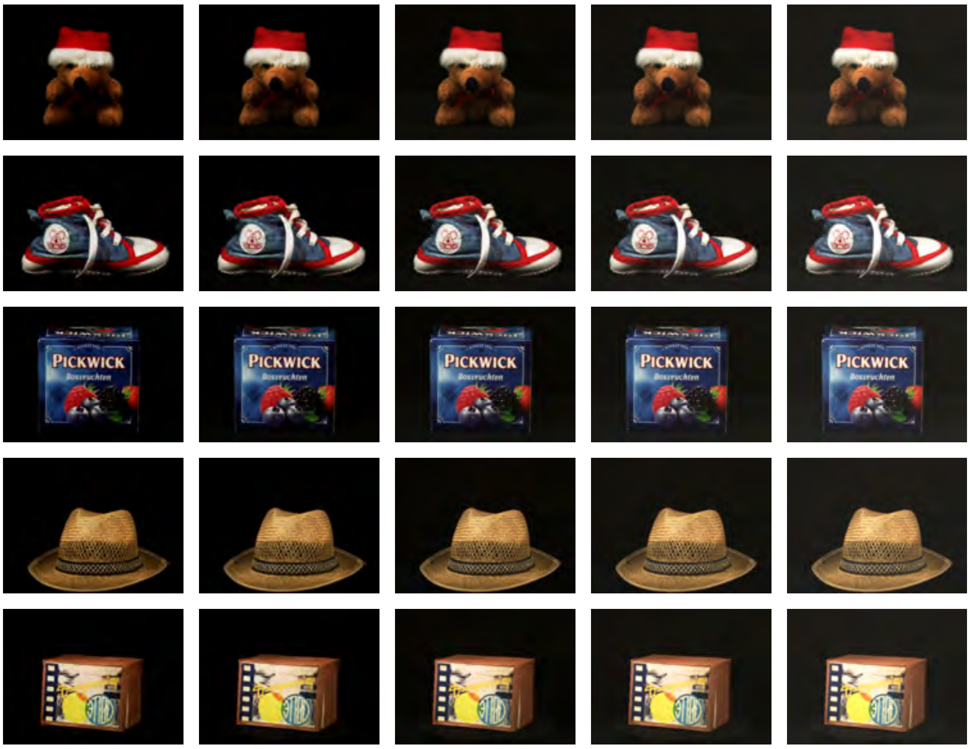


Figure 4.3: Samples from the LIS collection. From left to right: the value of β is respectively -20, -10, 0, 10, 20.

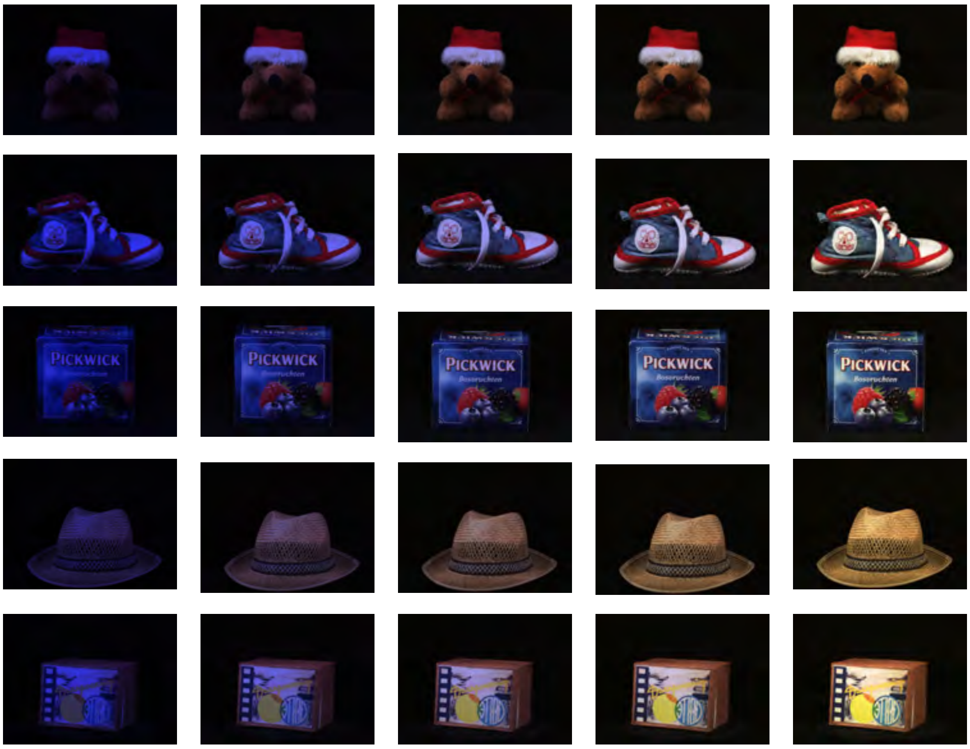


Figure 4.4: Samples from the LCC collection. From left to right: the value of α is respectively 0.2, 0.4, 0.6, 0.8, 1.0 for the Red and Green channels while keeping $\alpha = 1.0$ for the Blue channel.

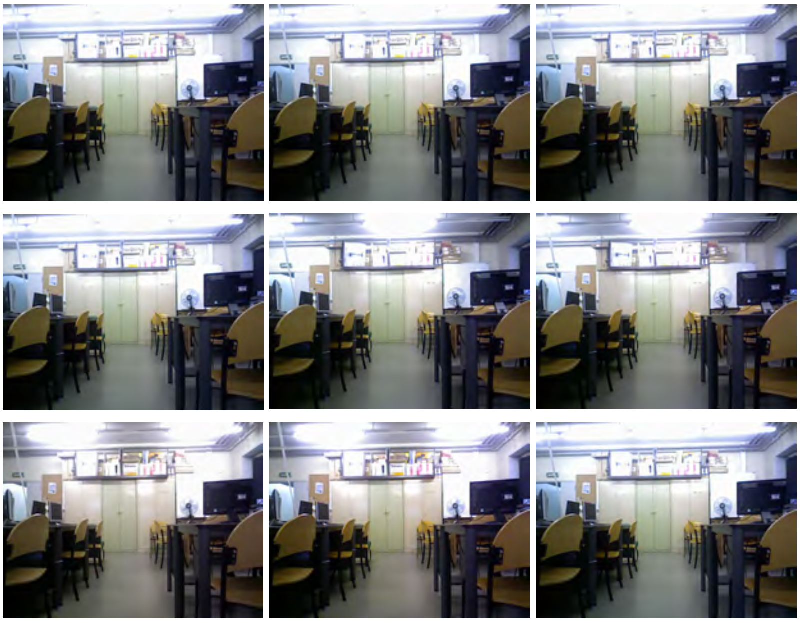


Figure 4.5: Real world image set: camera mounted in a fixed position in an office environment. The scene illumination was slight varied through the combination in the state (on/off) of ceiling lightings.

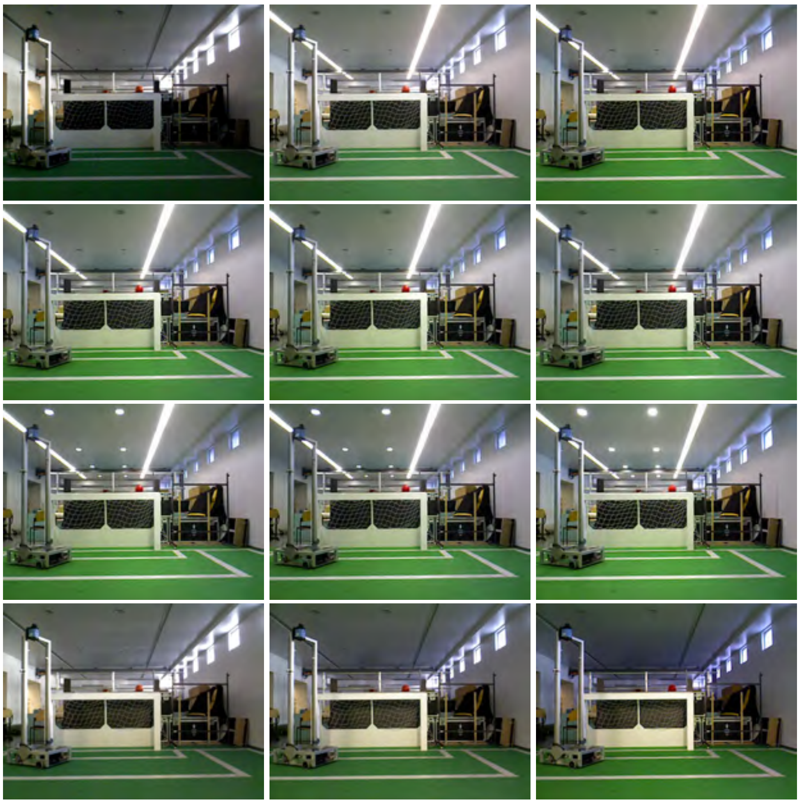


Figure 4.6: Real world image set: camera mounted in a fixed position in an robotic soccer field. The scene illumination was significantly varied through the combination in the state (on/off) and intensity of ceiling lightings.

The remaining conditions were artificially created by performing the appropriate transformations to one reference image of each object in the ALOI dataset. The LIC collection was designed to evaluate the effects of a scalar variation in the light source. To create this collection, all the RGB channels of the reference images were equally multiplied by a factor $\alpha \in \{1/2.0, 1/1.5, 1/1.2, 1/1.1, 1, 1.1, 1.2, 1.5, 2.0\}$, Figure 4.2. The LIS collection is designed to evaluate the algorithm's shift invariance. To create this collection, all the RGB channels of the reference images were equally shifted by an offset $\beta \in \{-20, -15, -10, -5, 0, 5, 10, 15, 20\}$, Figure 4.3. The LCC was designed to evaluate the algorithm's color invariance. Its main difference to the ALCC collection is the color transition of the illuminant, which varies from bluish to white in the LCC. To create this collection, the red and green channels of the reference images were multiplied by a factor $\alpha \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, while keeping $\alpha \in \{1.0\}$ for the Blue channel, Figure 4.4.

4.6.2 Real World Image Set

In order to assess the robustness of the proposed feature detection algorithms on real world images, real illumination conditions and variations, we created a real world image set composed by the Office and the Soccer image collections. The Office collection is a set of nine images. To collect these images, a camera was mounted in a fixed position in a typical office environment. Between each scene captured, the room illumination was slight varied through combinations in the state (on/off) of ceiling lightnings. A reduced size sample of the images from this collection are depicted in the Figure 4.5.

The Soccer data set is a more harsh set of images. It consists of thirteen indoor images of a robotic soccer field, taken with the camera mounted in a fixed position. In this collection, the scene illumination was varied through several combinations in the state as well as in the intensity of individually regulated ceiling lightnings. This dataset offers a more challenging environment for robust feature detection since it contains non-uniform illumination due to multiple sources (different bulb lamps in the ceiling and natural illumination from the windows), as well as variations in the color of the illuminant, shading, shadows, specularities, and interreflections. A reduced size sample of the images from this collection are depicted in the Figure 4.6

4.6.3 Shift Invariance Experiment

Figures 4.7,4.8 and 4.9, presents the results of SURF, LN SURF and LSAC SURF repeatability rates for shift variation $\beta \in \{-20, -15, -10, -5, 0, 5, 10, 15, 20\}$ (LIS image collection). An exploratory data analysis reveals a difference between the mean and the median repeatability rate in both algorithms, Table 4.4, which indicates that the observations may not be well modeled by a normal distribution. The same tendency is verified through the values of the skewness and kurtosis, which fall out of the two standard error range.

At 0.05 significance level, the Kolmogorov-Smirnov test confirms that the distribution of either algorithms ($p<0.001$) are not normally distributed. The box plot presented in the Figure 4.10

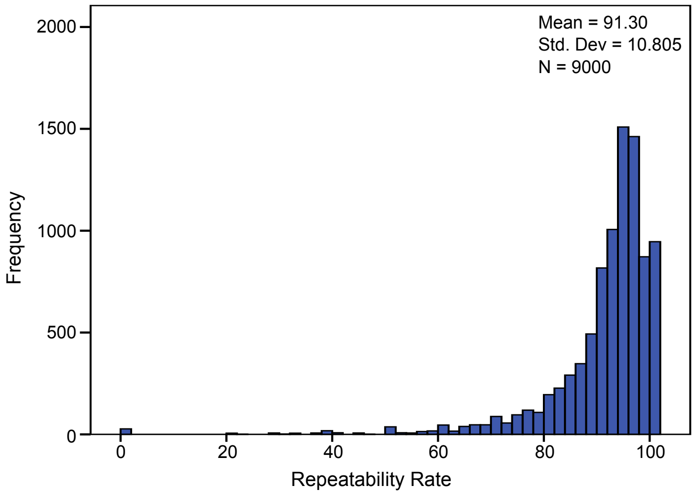


Figure 4.7: Histogram of SURF repeatability rate in the LIS image collection.

depicts the increase of the median repeatability rate from the original algorithm to both LN SURF and LSAC SURF. One drawback of both proposed algorithms was the increase of the statistical dispersion, demonstrated by an increase in the interquartile range and in the standard deviation.

Figure 4.11 depicts a high median repeatability rate for the three algorithms, which was held above 85% in all illumination conditions. The statistical analysis performed with the Wilcoxon signed rank test (Table 4.5) indicated no statistical evidence that the repeatability of LN SURF is greater than the original SURF ($Z = -0.970,\ p_{UD} = 0.162$). On the other hand, when comparing SURF with LSAC SURF the test indicated that LSAC SURF provided greater feature repeatability ($Z = -13.999,\ p_{UD} < 0.001$).

Table 4.4: Descriptives of the shift invariance experiment - LIS image collection.

	SURF		LN SURF		LSAC SURF	
	Statistic	Std Error	Statistic	Std Error	Statistic	Std Error
Mean	91.2992	0.11390	87.9670	0.20866	92.3679	0.13705
Median	94.3396		100.0000		100.0000	
Variance	116.749		391.853		169.035	
Std. Deviation	10.80506		19.79528		13.00133	
Skewness	-3.648	0.026	-2.120	0.026	-2.514	0.026
Kurtosis	20.443	0.052	4.562	0.052	7.647	0.052

Table 4.5: Wilcoxon signed rank test of the shift invariance experiment - LIS image collection.

	LN SURF - SURF			LSAC SURF - SURF		
	N	Mean Rank	Sum of Ranks	N	Mean Rank	Sum of Ranks
Negative Ranks	3206	5506.78	17654727.00	3120	4680.63	14603563.50
Positive Ranks	5146	3347.73	17227401.00	5300	3933.74	20848846.50
Ties	648			580		

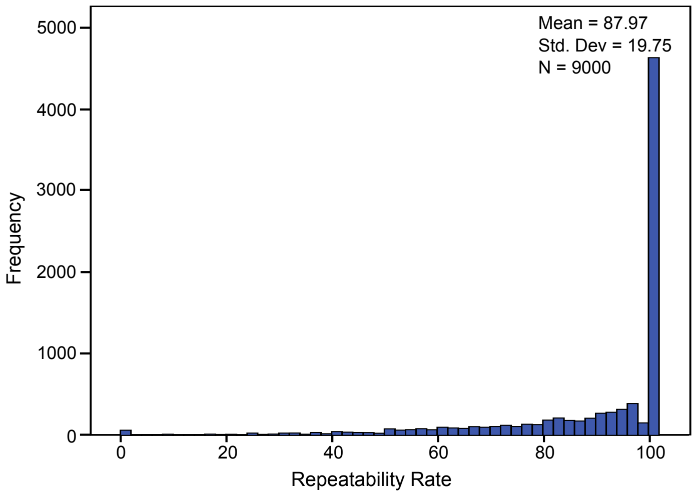


Figure 4.8: Histogram of LN SURF repeatability rate in the LIS image collection.

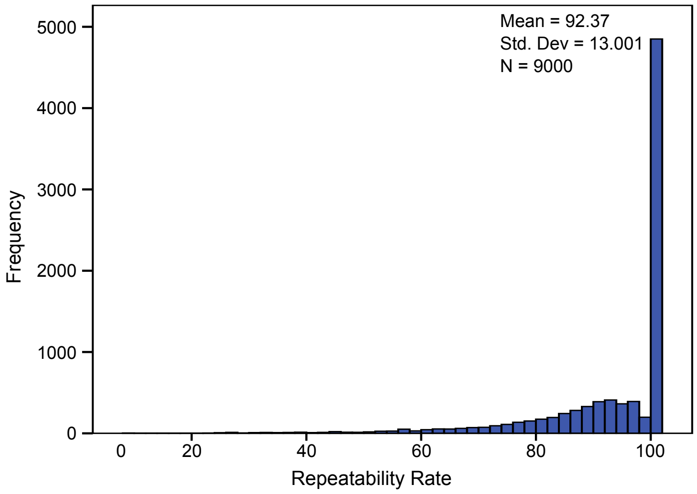


Figure 4.9: Histogram of LSAC SURF repeatability rate in the LIS image collection.

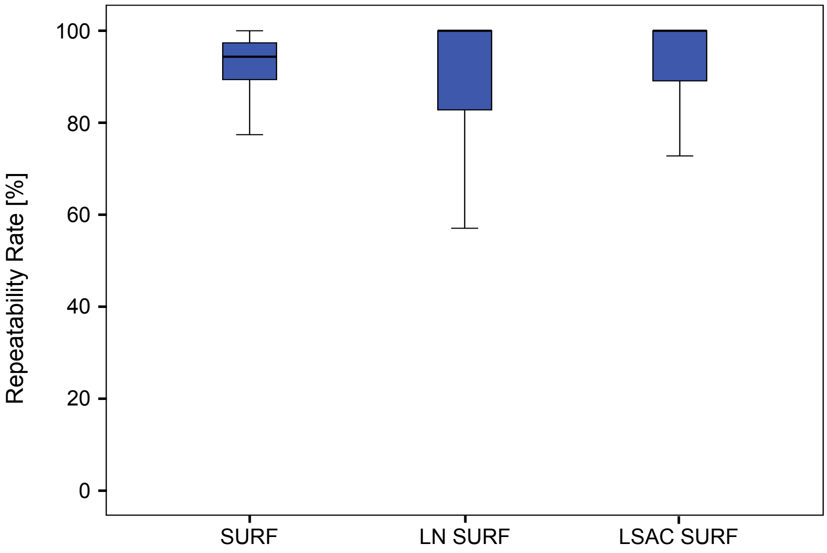


Figure 4.10: Box plot of SURF, LN SURF and LSAC SURF repeatability rates in the LIS image collection. The median repeatability is represented by the horizontal line drawn in the box. The lower quartile value is at the bottom of the box, while the upper quartile value is at the top. The whiskers represent the minimum and the maxim values.

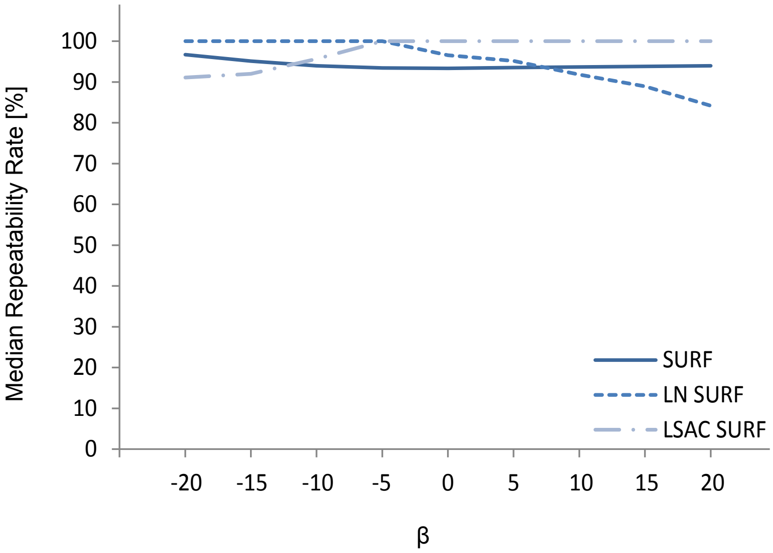


Figure 4.11: A comparison between the median repeatability rate of SURF and the two proposed algorithms over the controlled images present in the LIS collection ($\beta \in \{-20,-15,-10,-5,0,5,10,15,20\}$).

4.6.4 Scale Invariance Experiment

Figures 4.12, 4.13 and 4.14, presents the results of SURF, LN SURF and LSAC SURF repeatability rates for a scale variation $\alpha \in \{1/2.0, 1/1.5, 1/1.2, 1/1.1, 1, 1.1, 1.2, 1.5, 2.0\}$ (LIC image collection). Once again, the exploratory data analysis reveals a difference between the mean and the median repeatability rate in both algorithms, Table 4.6, indicating that the observations may not be well modeled by a normal distribution. This tendency is also verified through the values of the skewness and kurtosis, which fall out of the two standard error range.

At 0.05 significance level, the Kolmogorov-Smirnov test confirms that the distribution of either algorithms (p<0.001) are not normally distributed. The box plot presented in the Figure 4.15 depicts a significant increase of the median repeatability rate from SURF to both LN SURF and LSAC SURF. An additional advantage of both proposed algorithms was the decrease of the statistical dispersion, demonstrated by a reduction in the interquartile range, and in the standard deviation.

As shown in Figure 4.16, the median repeatability rate of SURF algorithm tend to decrease with higher values of α. Such observation is explained from the maximum number of common features, which is limited by the image with the smallest number of features detected. Since SURF detection response is proportional to α^2, the smallest number of features tends to be detected in darker images ($\alpha < 1$). The proposed algorithms, on the other hand, demonstrate a much higher and constant mean repeatability rate. The Wilcoxon signed rank test (Table 4.7) indicated that both LN SURF ($Z = -78.782$, $p_{UD} < 0.001$) and LSAC SURF ($Z = -79.999$, $p_{UD} < 0.001$) provided a significant increase in the feature repeatability.

Table 4.6: Descriptives of the scale invariance experiment - LIC image collection.

	SURF		LN SURF		LSAC SURF	
	Statistic	Std Error	Statistic	Std Error	Statistic	Std Error
Mean	46.4145	0.28265	88.5662	0.20866	87.7919	0.16143
Median	44.0000		100.0000		93.0035	
Variance	719.025		386.916		234.530	
Std. Deviation	26.81465		19.67019		15.31438	
Skewness	0.418	0.026	-2.126	0.026	-1.840	0.026
Kurtosis	-0.483	0.052	4.381	0.052	4.090	0.052

Table 4.7: Wilcoxon signed rank test of the scale invariance experiment - LIC collection.

	LN SURF - SURF			LSAC SURF - SURF		
	N	Mean Rank	Sum of Ranks	N	Mean Rank	Sum of Ranks
Negative Ranks	308	885.76	272815.50	354	706.12	249965.50
Positive Ranks	8221	4391.60	36103369.50	8267	4465.36	36915165.50
Ties	471			379		

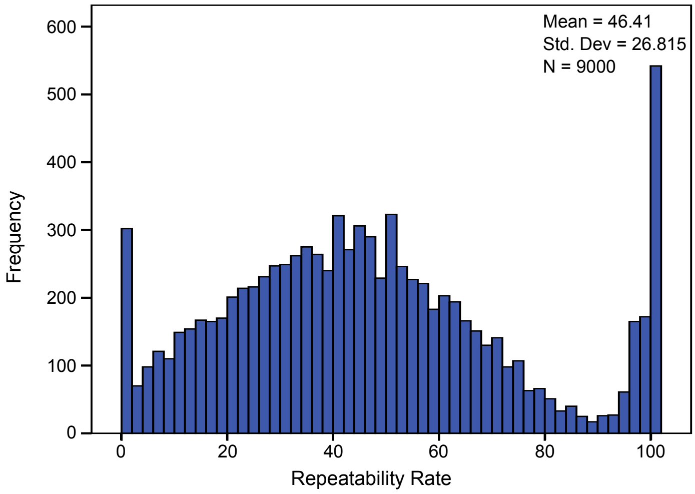


Figure 4.12: Histogram of SURF in the LIC image collection.

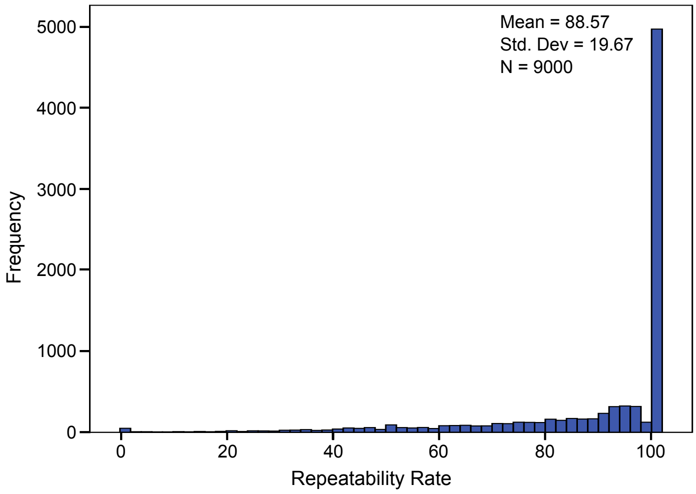


Figure 4.13: Histogram of LN SURF in the LIC image collection.

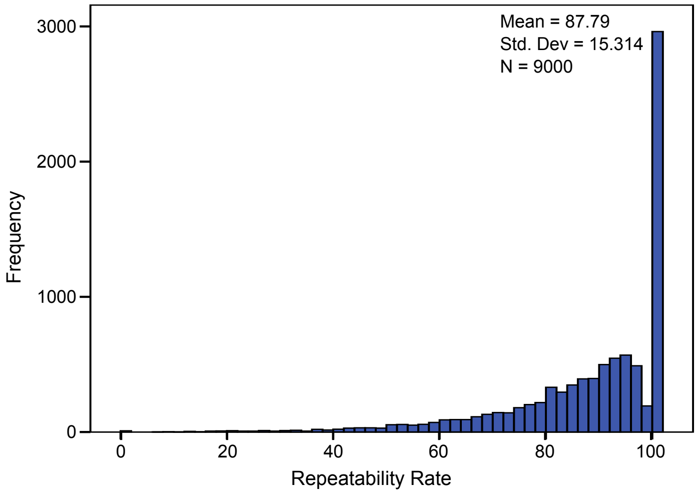


Figure 4.14: Histogram of LSAC SURF in the LIC image collection.

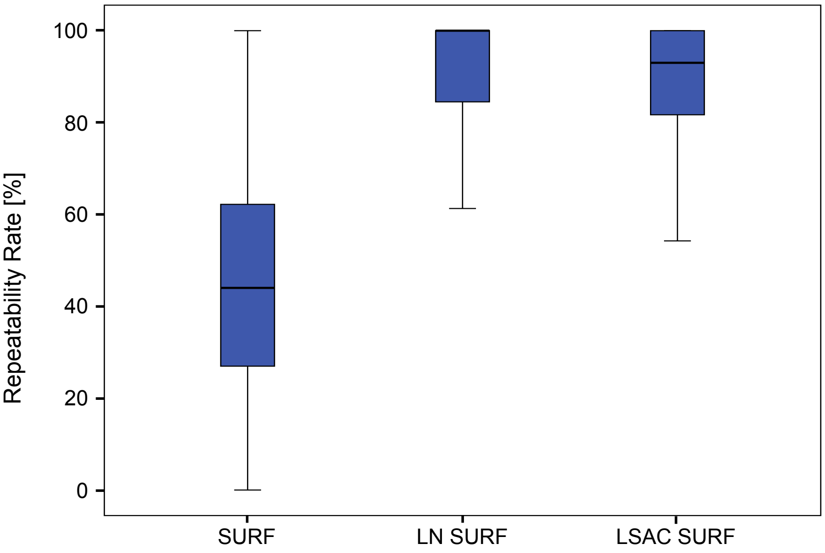


Figure 4.15: Box plot of SURF, LN SURF and LSAC SURF repeatability rates in the LIC image collection. The median repeatability is represented by the horizontal line drawn in the box. The lower quartile value is at the bottom of the box, while the upper quartile value is at the top. The whiskers represent the minimum and the maxim values.

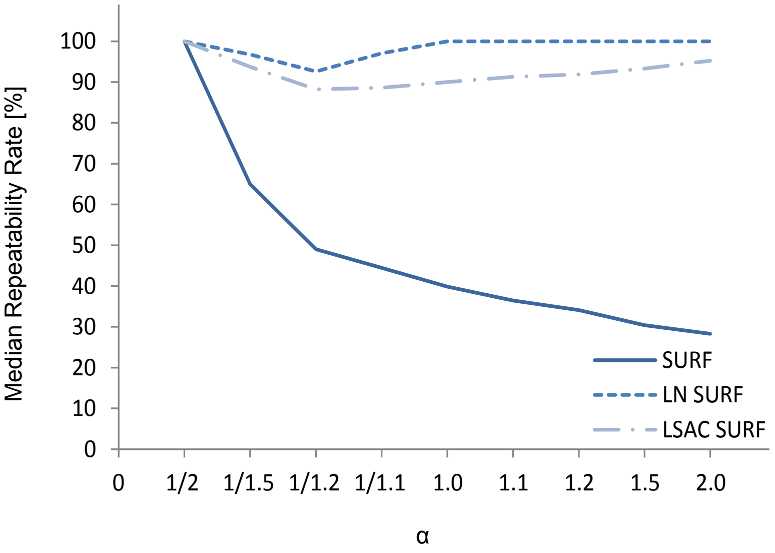


Figure 4.16: A comparison between the median repeatability rate of SURF and the two proposed algorithms over the controlled images present in the LIC collection ($\alpha \in \{1/2.0, 1/1.5, 1/1.2, 1/1.1, 1, 1.1, 1.2, 1.5, 2.0\}$).

4.6.5 Color Invariance Experiment

Figures 4.17, 4.18 and 4.19, presents the repeatability rates for a color variation from yellow to white (ALCC image collection). The exploratory data analysis reveals a difference between the mean and the median repeatability rate in both algorithms, Table 4.8, indicating that the observations may not be well modeled by a normal distribution. This tendency is also verified through the values of the skewness and kurtosis, which fall out of the two standard error range.

At 0.05 significance level, the Kolmogorov-Smirnov test confirms that the distribution of either algorithms (p<0.001) are not normally distributed. The box plot presented in the Figure 4.20 depicts the constancy of the median repeatability rate. When comparing SURF with LN SURF, one can note an increase in the statistical dispersion, denoted by both the interquartile range and the standard deviation. On the other hand, when comparing with LSAC SURF one can note a decrease in the standard deviation, and consequently a reduction in the statistical dispersion.

It can be seen from the Figure 4.21 that none of the algorithms was significantly affected when varying the color of the illuminant from yellow to white. In this illumination condition, the Wilcoxon signed rank test (Table 4.9) did not provide statistical evidence that the neither LN SURF ($Z = -10.583,\ p_{UD} = 0.160$) nor LSAC SURF ($Z = -10.583,\ p_{UE} < 0.001$) were able to improve the repeatability.

Table 4.8: Descriptives of the color invariance experiment. Variation in the color of the illuminant from yellow to white - ALCC collection.

	SURF		LN SURF		LSAC SURF	
	Statistic	Std Error	Statistic	Std Error	Statistic	Std Error
Mean	92.4384	0.11390	91.7943	0.20866	91.8599	0.08735
Median	94.8617		95.8791		93.5484	
Variance	102.391		157.733		68.663	
Std. Deviation	10.11887		12.55920		8.28633	
Skewness	-3.912	0.026	-3.266	0.026	-2.881	0.026
Kurtosis	25.054	0.052	16.254	0.052	18.952	0.052

Table 4.9: Wilcoxon signed rank test of the color invariance experiment. Variation in the color of the illuminant from yellow to white - ALCC collection.

	LN SURF - SURF			LSAC SURF - SURF		
	N	Mean Rank	Sum of Ranks	N	Mean Rank	Sum of Ranks
Negative Ranks	3763	4279.20	16102617.50	4808	4244.52	20407645.50
Positive Ranks	4314	3829.48	16520385.50	3681	4245.63	15628159.50
Ties	923			511		

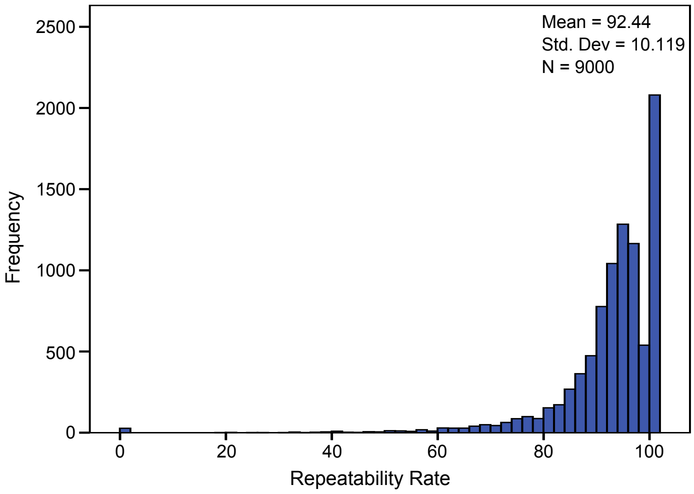


Figure 4.17: Histogram of SURF repeatability rate in the ALCC image collection.

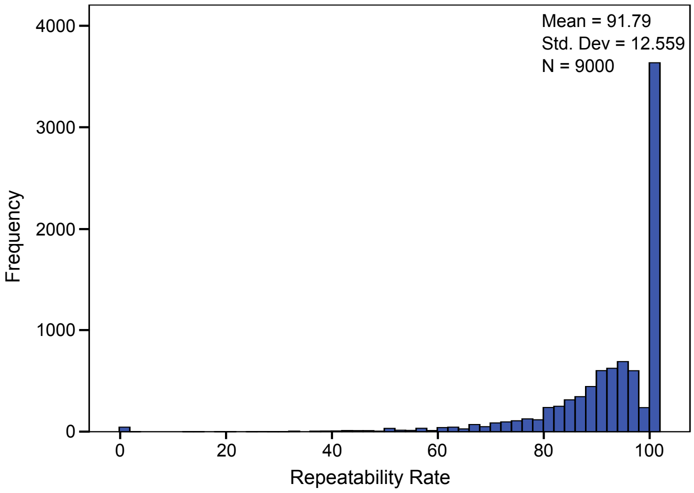


Figure 4.18: Histogram of LN SURF repeatability rate in the ALCC image collection.

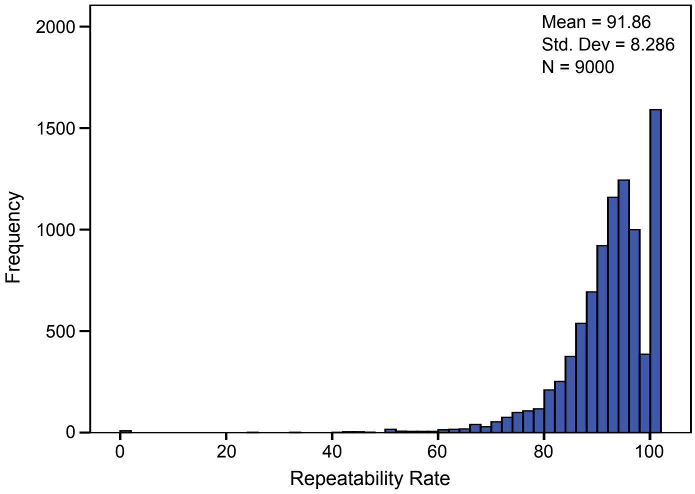


Figure 4.19: Histogram of LSAC SURF repeatability rate in the ALCC image collection.

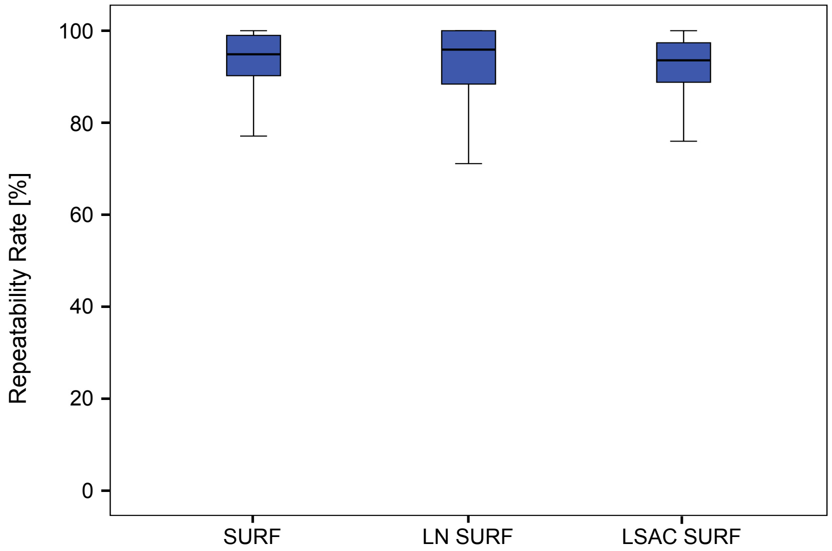


Figure 4.20: Box plot of SURF, LN SURF and LSAC SURF repeatability rates in the ALCC image collection. The median repeatability is represented by the horizontal line drawn in the box. The lower quartile value is at the bottom of the box, while the upper quartile value is at the top. The whiskers represent the minimum and the maxim values.

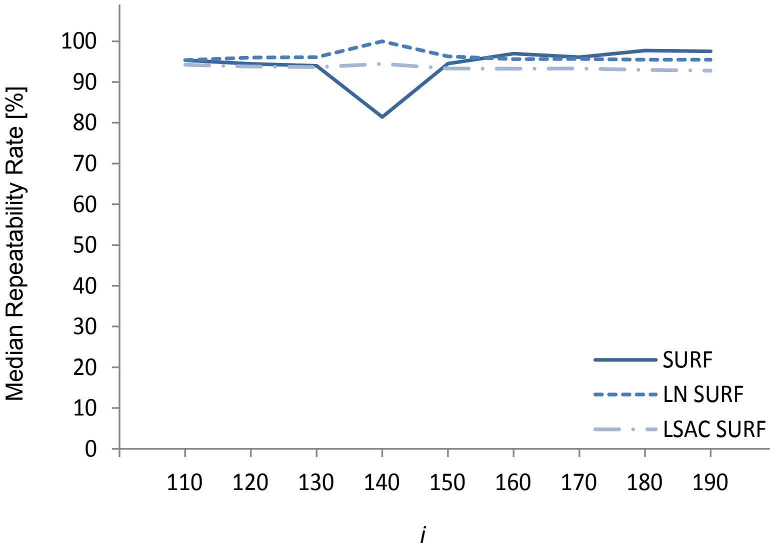


Figure 4.21: A comparison between the median repeatability rate of SURF and the two proposed algorithms over the controlled images present in the ALCC collection (light source varying from yellow to white - $i \in \{110, 120, 130, 140, 150, 160, 170, 180, 190\}$).

Figures 4.22, 4.23 and 4.24, presents the repeatability rates for a color variation from bluish to white (LCC image collection). An exploratory data analysis reveals a difference between the mean and the median repeatability rate in both algorithms, Table 4.10, indicating that the observations may not be well modeled by a normal distribution. This tendency is also verified through the values of the skewness and kurtosis, which fall out of the two standard error range.

At 0.05 significance level, the Kolmogorov-Smirnov test confirms that the distribution of either algorithms (p<0.001) are not normally distributed. The box plot presented in the Figure 4.25 depicts a significant shift of the median repeatability rate from SURF to LN SURF and LSAC SURF. An additional advantage of both proposed algorithms was the decrease of the statistical dispersion.

The results depicted in Figure 4.26 ($\alpha \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ in the red and green color channels) demonstrates that the repeatability rate of SURF algorithm declines sharply. The tendency to find higher mean repeatability rate for the lower values of α is once again verified in SURF. The proposed algorithms, on the other hand, demonstrate a much higher and constant overall mean repeatability rate. The Wilcoxon signed rank test (Table 4.11) indicate that both LN SURF ($Z = -79.887$, $p_{UD} < 0.001$) and LSAC SURF ($Z = -79.015$, $p_{UD} < 0.001$) provided a significant increase in the feature repeatability.

Table 4.10: Descriptives of the color invariance experiment. Variation in the color of the illuminant from bluish to white - LCC collection.

	SURF		LN SURF		LSAC SURF	
	Statistic	Std Error	Statistic	Std Error	Statistic	Std Error
Mean	29.7037	0.30586	91.7943	0.13239	73.0255	0.23540
Median	21.2121		95.8791		77.7778	
Variance	841.948		157.733		498.730	
Std. Deviation	29.01634		12.55920		22.33225	
Skewness	1.181	0.026	-3.266	0.026	-0.865	0.026
Kurtosis	0.480	0.052	16.254	0.052	0.196	0.052

Table 4.11: Wilcoxon signed rank test of the color invariance experiment. Variation in the color of the illuminant from bluish to white - LCC collection.

	LN SURF - SURF			LSAC SURF - SURF		
	N	Mean Rank	Sum of Ranks	N	Mean Rank	Sum of Ranks
Negative Ranks	558	608.57	339579.50	371	821.80	304887.00
Positive Ranks	8256	4664.26	38508125.50	8234	4459.85	36722428.00
Ties	186			395		

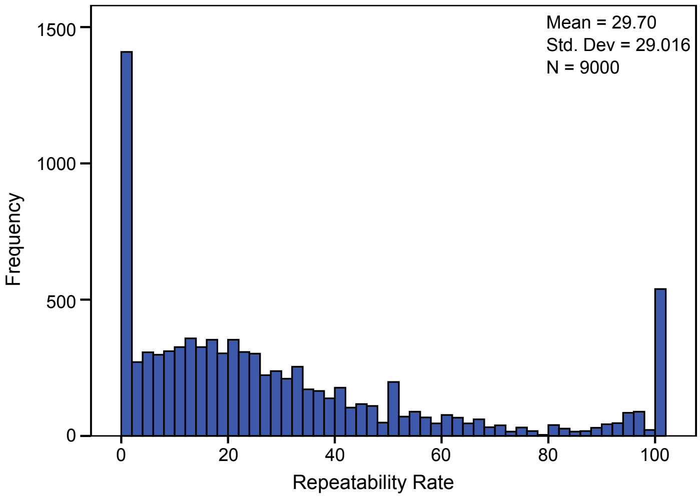


Figure 4.22: Histogram of SURF repeatability rate in the LCC image collection.

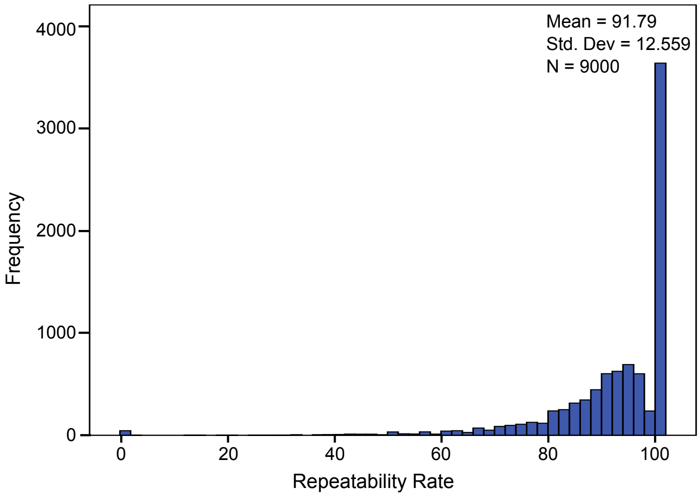


Figure 4.23: Histogram of LN SURF repeatability rate in the LCC image collection.

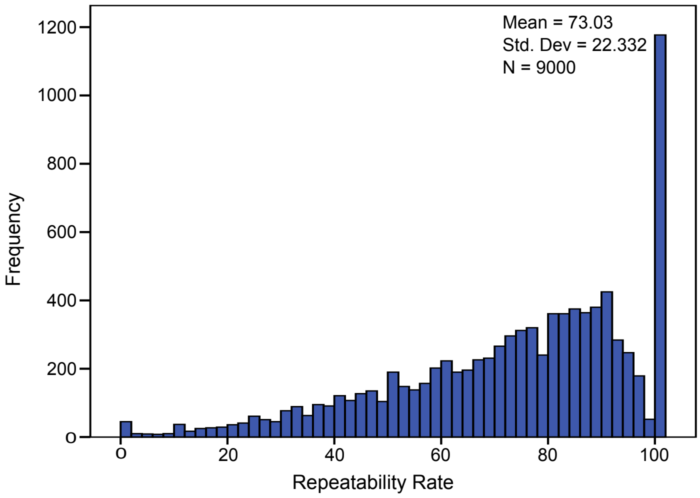


Figure 4.24: Histogram of LSAC SURF repeatability rate in the LCC image collection.

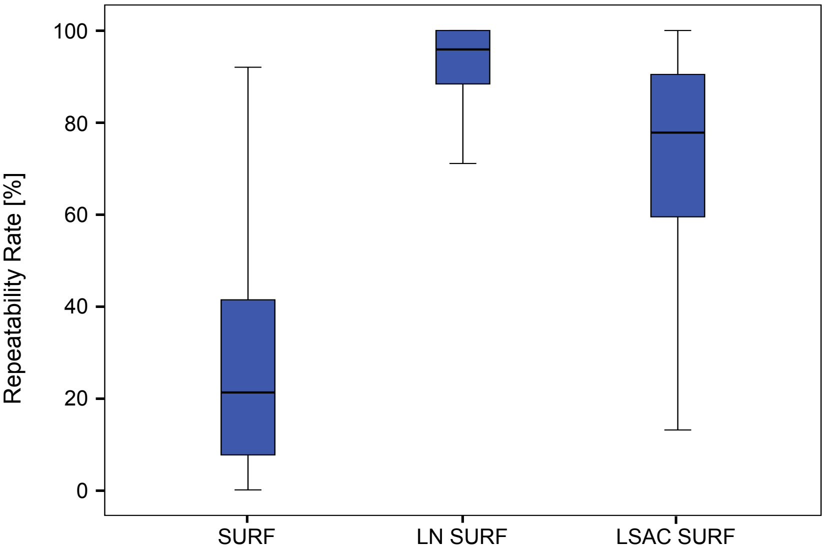


Figure 4.25: Box plot of SURF, LN SURF and LSAC SURF repeatability rates in the LCC image collection. The median repeatability is represented by the horizontal line drawn in the box. The lower quartile value is at the bottom of the box, while the upper quartile value is at the top. The whiskers represent the minimum and the maxim values.

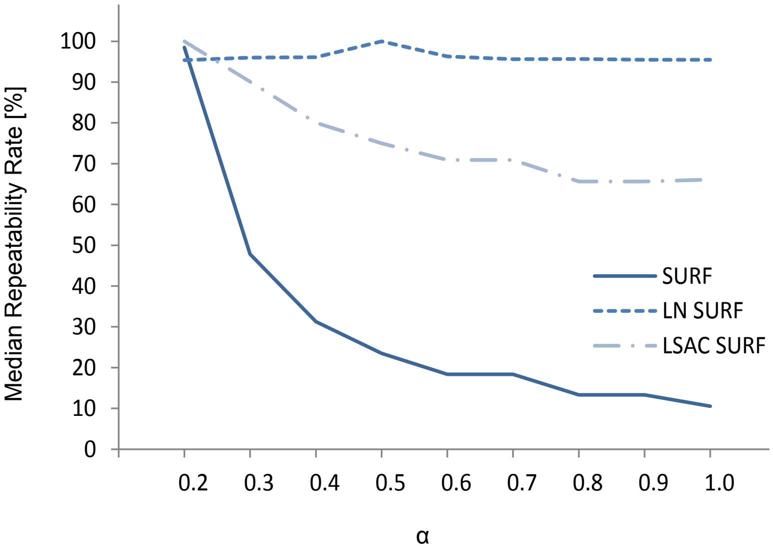


Figure 4.26: A comparison between the median repeatability rate of SURF and the two proposed algorithms over the controlled images present in the LCC collection (light source varying from bluish to white - $\alpha \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ for the red and green color channels).

4.6.6 Small Photometric Variation Experiment

Figure 4.27 presents the repeatability rates for a set of real world images with small photometric variation (Office image collection). An exploratory data analysis reveals a difference between the mean and the median repeatability rate in both algorithms, Tables 4.12, which indicates that the observations may not be modeled by a normal distribution. The same tendency, however, is not verified through the values of the skewness and kurtosis, which fall within the two standard error range. At 0.05 significance level, the Shapiro-Wilk test indicates that the distribution of either SURF ($p = 0.037$) and LN SURF ($p = 0.025$) can not be modeled as normal distribution, while LSAC SURF ($p = 0.495$) can. The box plot presented in the Figure 4.28 depicts a decrease of the median repeatability rate from from SURF to LN SURF, and a significant increase from SURF to LSAC SURF. An additional advantage of the LSAC SURF was the decrease of the statistical dispersion of the observations.

The results demonstrate a high repeatability rate for both SURF and LSAC SURF algorithms, which was held above 75% in all illumination conditions. LN SURF, on the other hand, demonstrated to be highly susceptible to image noise, yielding a repeatability rate ranging from 50% to 70%. Such behavior was not evident in the previous image collections because, when the controlled images were created, the noise of the reference image was subjected to the same transformations that the rest of the pixels. In real world images like the Office collection, though, the noise varies not only differently from the rest of the pixels but also in its location. Despite the results of the EDA and the Shapiro-Wilk test, due to the small number of observations the statistical analysis was performed with the Wilcoxon signed rank test (Table 4.13). The test indicated that while LN Surf failed to improve the repeatability rate ($Z = -2.549,\ p_{UE} = 0.004$), LSAC SURF was indeed able to improve feature illumination invariance ($Z = -2.380,\ p_{UD} = 0.008$).

Table 4.12: Descriptives of the small photometric variation experiment - Office image collection.

	SURF		LN SURF		LSAC SURF	
	Statistic	Std Error	Statistic	Std Error	Statistic	Std Error
Mean	82.7506	2.93849	59.8088	2.94757	90.1660	2.15289
Median	78.1915		57.6101		91.3580	
Variance	77.713		78.194		41.715	
Std. Deviation	8.81548		8.84272		6.45868	
Skewness	0.093	0.717	0.543	0.717	0.173	0.717
Kurtosis	-2.095	1.400	1.400	1.481	-1.327	1.400

Table 4.13: Wilcoxon signed rank test - Office collection.

	LN SURF - SURF			LSAC SURF - SURF		
	N	Mean Rank	Sum of Ranks	N	Mean Rank	Sum of Ranks
Negative Ranks	8	5.50	44.00	1	1.00	1.00
Positive Ranks	1	1.00	1.00	8	5.50	44.00
Ties	0			0		

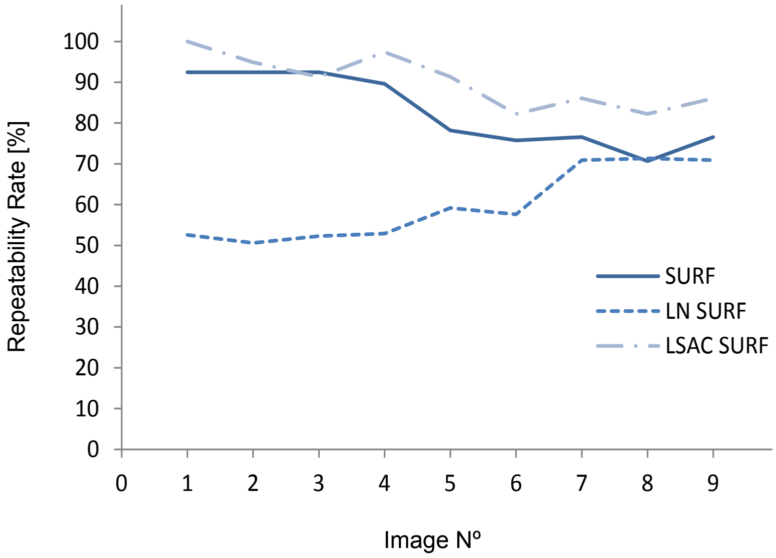


Figure 4.27: A comparison between the median repeatability rate of SURF and the two proposed algorithms over the real world images present in the Office collection.

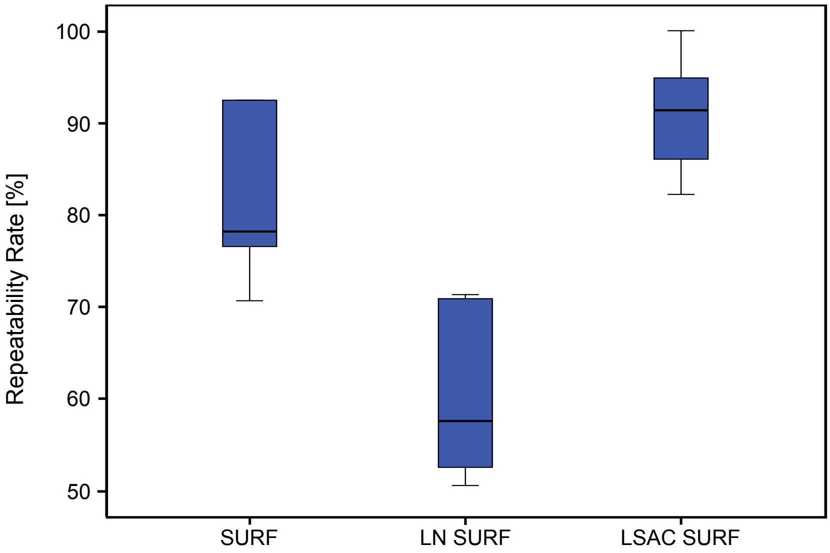


Figure 4.28: Box plot of SURF, LN SURF and LSAC SURF repeatability rates in the Office image collection. The median repeatability is represented by the horizontal line drawn in the box. The lower quartile value is at the bottom of the box, while the upper quartile value is at the top. The whiskers represent the minimum and the maxim values.

4.6.7 Large Photometric Variation Experiment

Figure 4.29 presents the repeatability rates in the Soccer image collection. An exploratory data analysis reveals a large difference between the mean and the median repeatability rate in both algorithms, Table 4.14. A similar tendency is verified through the values of the skewness and kurtosis, which fall out of the two standard error range.

At 0.05 significance level, the Shapiro-Wilk test confirms that both distributions, SURF (p=0.042) and LSAC SURF (p=0.005), are not well modeled as normal distribution, while LN SURF (p=0.185) is. The box plot presented in the Figure 4.30 does not indicate an improvement from SURF to LN SURF. However, when comparing SURF and LSAC SURF one can note a great shift of the median repeatability rate. One drawback of both proposed algorithms was the increase of the statistical dispersion.

The results demonstrate that the repeatability rates of SURF and LN SURF are very low in the Soccer dataset. LSAC, on the other hand, demonstrates a much higher and constant repeatability rate. The Wilcoxon signed rank test (Table 4.15) indicate that LN SURF failed to provide an improvement ($Z = -0.941,\ p_{UD} < 0.189$). When compared to LSAC SURF, the test suggests that there is a significant increase in the feature repeatability ($Z = -3.059,\ p_{UD} < 0.001$).

Table 4.14: Descriptives of the large photometric variations experiment - Soccer image collection.

	SURF		LN SURF		LSAC SURF	
	Statistic	Std Error	Statistic	Std Error	Statistic	Std Error
Mean	29.5735	1.61725	31.6195	1.86469	75.8602	3.72678
Median	27.7605		30.1630		69.4541	
Variance	31.386		41.725		166.667	
Std. Deviation	5.60231		6.45947		12.90996	
Skewness	1.535	0.637	1.168	0.637	0.864	0.637
Kurtosis	2.819	1.232	1.552	1.232	-1.186	1.232

Table 4.15: Wilcoxon signed rank test of the large photometric variation experiment - Soccer collection.

	LN SURF - SURF			LSAC SURF - SURF		
	N	Mean Rank	Sum of Ranks	N	Mean Rank	Sum of Ranks
Negative Ranks	4	6.75	27.00	0	0.00	0.00
Positive Ranks	8	6.38	51.00	12	6.50	78.00
Ties	0			0		

Figure 4.31 presents the features detected with SURF and LSAC SURF algorithms. To facilitate the distinction, hard features were represented by red dots, and soft features represented by green dots. A visual inspection indicates that several features detected with SURF are located in saturated pixels. We can also note that several LN SURF features were detected in flat rather than on blob-like image regions, demonstrating a significant susceptibility to image noise.

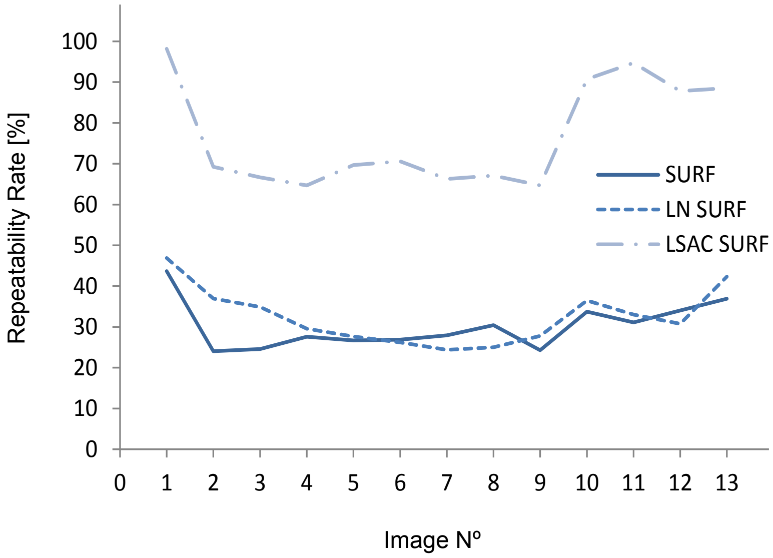


Figure 4.29: A comparison between the median repeatability rate of SURF and the two proposed algorithms over the real world images present in the Soccer collection.

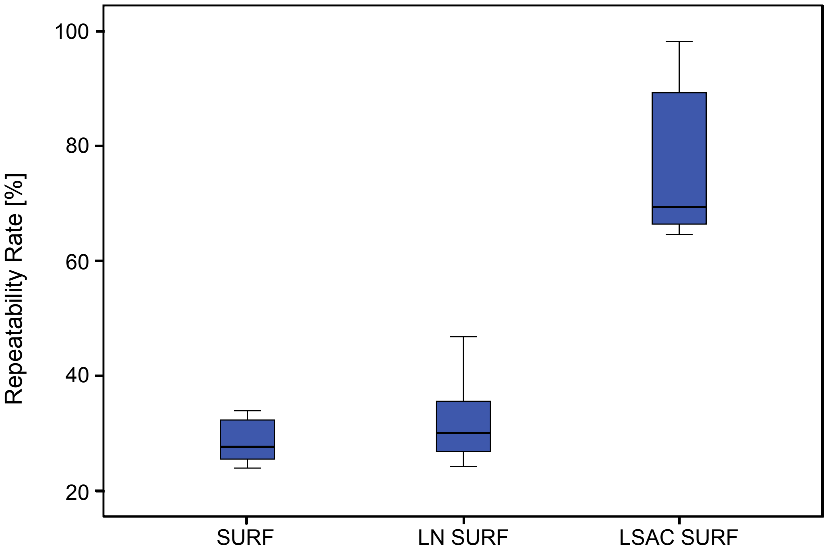
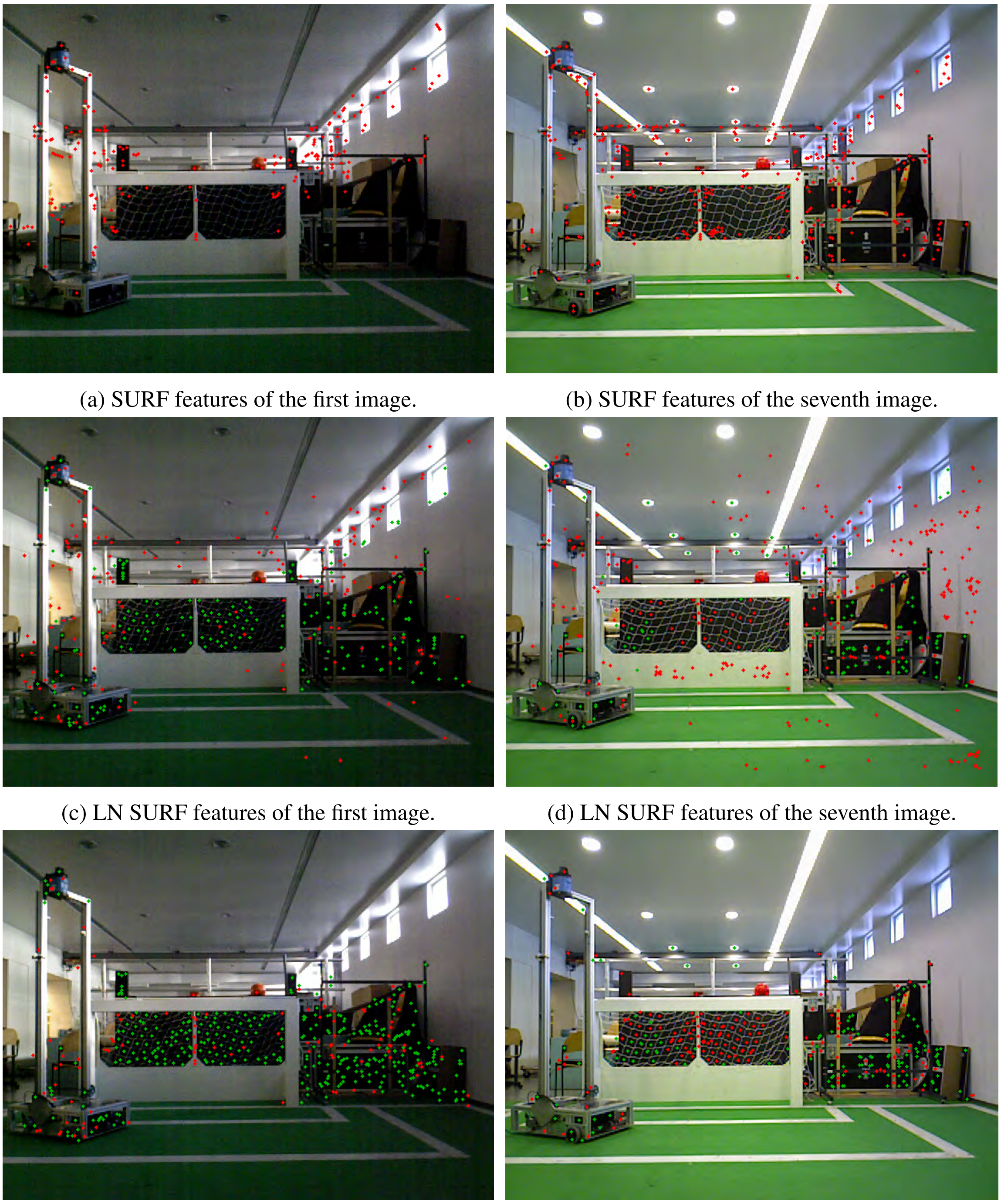


Figure 4.30: Box plot of SURF, LN SURF and LSAC SURF repeatability rates in the Soccer image collection. The median repeatability is represented by the horizontal line drawn in the box. The lower quartile value is at the bottom of the box, while the upper quartile value is at the top. The whiskers represent the minimum and the maxim values.



(a) SURF features of the first image.

(b) SURF features of the seventh image.

(c) LN SURF features of the first image.

(d) LN SURF features of the seventh image.

(e) LSAC SURF features of the first image.

(f) LSAC SURF features of the seventh image.

Figure 4.31: Comparison between the SURF, LN SURF and LSAC SURF features detected in the first (A, C, E) and the seventh (B, D, F) images of the Soccer collection. Red dots represent hard features, while green dots represent soft features.

4.7 Conclusions

In this chapter, the reader was introduced to the related work on color feature detectors and methodologies of color constancy. We have seen that despite considered one of the most successful color constant algorithms, Gamut mapping is not a viable option for robotic vision systems since it is computationally complex and requires an image data set with known light sources. Several low level color constant algorithms are less complex, faster, and only slightly outperformed by the gamut mapping, but generally assume that the scene is uniformly illuminated. Since in real world images the illumination is generally not uniform, this premise is not fully verified. An alternative is the Local Space Average Color descriptor, which estimates the illuminant locally for each point of the scene.

Next, the analysis developed in the Section 4.3 provided the theoretical foundations to understand the elements that compromise the illumination invariance in feature detection functions. It demonstrated that the image derivative is not affected by the diffuse term β (Equation 2.26) when it is constant over the image (or at least over the image patch in which the filter is being computed). Therefore, all the three algorithms analyzed presented partial invariance to illumination changes, provided by the computation of the image derivatives. However, only SIFT demonstrated invariance when subjected to the scalar variation. SIFT invariance is provided by the division of the trace by the determinant of the Hessian matrix, which cancel the effects of α. Harris corners showed the biggest variation, proportional to α^4 , while SURF an intermediate variation, proportional to α^2.

The proposed LN SURF methodology extended the original SURF feature detector combining it with the local normalization methodology. Unlike other authors that use color space conversion to achieve illumination invariance, our method normalize the SURF detection function to provide invariant responses. The algorithm demonstrated a very high repeatability rate in all the four controlled image sets tested. In addition, it was able to statistically outperform the original SURF detector in two of the most challenging scenarios: the LIS and the LCC image collections. The experiments with the real world image set, though, exposed the main problem of the approach, its high susceptible to image noise. In both of the two scenarios that the algorithm was tested (small and large illumination changes), the repeatability rates dropped considerably below the rates presented by the original SURF algorithm.

Our second approach (LSAC SURF) achieves photometric invariant feature responses using the Local Space Average Color descriptor as working space to detect illumination invariant SURF features. The inclusion of this preliminary step adds a small computational load, but demonstrated to provide a valuable improvement in the feature detection invariance. Similar to the previous approach, LSAC SURF demonstrated a very high repeatability rate in all the controlled image sets tested. In addition, the algorithm was able to statistically outperform the original SURF detector in three out of the four scenarios: LIC, LIS and the LCC image collections. The experimental results with the real world image sets confirmed the theoretical invariance provided by the proposed algorithm and also demonstrated its robustness to image noise.

Chapter 5

Vision-based Localization

Vision-based localization methodologies have demonstrated to be an important alternative to traditional wheel odometry. Its advantages becomes even more salient when robots are subjected to uneven terrain, wheel-slippage, over acceleration, fast turning, interaction with external bodies and interaction with internal forces like the wheelchair castor wheels. Therefore, a localization methodology that is independent of the wheel-ground interaction is more adequate for intelligent wheelchairs. Here we present a visual odometry approach based on inexpensive RGB-D cameras. The proposed algorithm localizes visually salient points, and uses the depth information of each pixel to estimate the robot translation and rotation updates at each frame. Experimental results showed an absolute trajectory error of around 2%, demonstrating the applicability of the localization algorithm in the navigation system of intelligent wheelchairs.

5.1 Introduction

In this chapter, we consider the problem of vision-based localization for mobile robots. In order to obtain fully autonomous robots, an accurate localization of the robot in the world is an essential capability. Missions to be achieved by the robot are often expressed in localization terms, such as "reach that position" or "return to the initial position". The correct execution of the trajectories provided by the planners relies on the precise knowledge of robot motions. In addition, if an accurate localization is estimated in real-time, the remaining computational resources can be allocated to perform other important robotic tasks such as planning, object recognition and visual perception. Indeed, many robotics applications can benefit from an accurate and fast localization. Two methodologies have become predominant in vision-based localization systems: one is the incremental trajectory estimation of the Visual Odometry (VO), and the other is the global position estimation of the Visual Simultaneous Localization and Mapping (VSLAM).

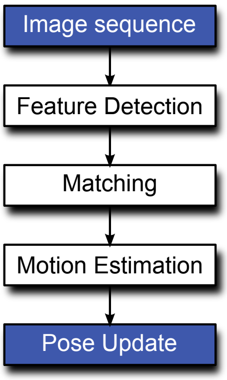


Figure 5.1: Typical visual odometry decomposition.

Visual odometry is the process of estimating the change in position and attitude of an agent (vehicle, human, robot, etc.) using data provided by a single or multiple cameras. The key idea is to determine the 3D camera motion by solving the transformation between a selection of image features extracted from consecutive pairs of frames [8, 252]. The methodology has been used in a wide variety of applications, including robotics, wearable computing, augmented reality and automotive.

Similar to the classic wheel odometry, VO estimates the agent current pose based on a previously known position and the accumulation of small fractions of motion. The main advantage of VO methods, with respect to wheel odometry, is that they are not affected by wheel slippage, uneven terrain or other harsh conditions. Since VO provides only relative localization, and due to its inherent accumulation of errors, the methodology may not be implemented as the only localization methodology, but instead combined with other global positioning system to periodically reset the accumulated error. Although the several implementations presented in the literature, visual odometry can be decomposed into three steps: feature detection, matching and motion estimation (Figure 5.1).

The feature detection step consists of selecting the interest image features over the camera frame. This is probably the most time consuming step of the whole algorithm, since each pixel of the image frame is compared with its neighborhood to check whether it is distinct or not. Therefore, besides good repeatability, this step may take into account the computer complexity of the desired feature detector algorithm.

Following, the matching[1] step compares features from consecutive frames in order to find, if possible, their correspondences. Thus, all features detected in the frame I_k are compared to every feature detected in the frame I_{k-1} within a fixed threshold distance. The evaluation of potential matches varies according to the detection algorithm. It can be from simply computing a normalized cross correlation over a square window[2], up to comparing the distance between the

[1] To establish matches when several unknown changes occur in the image, one must consider features that are as much invariant as possible with respect to any image transformation.

[2] In related work, the size of the window usually varies from 3x3 up to 11x11 pixels.

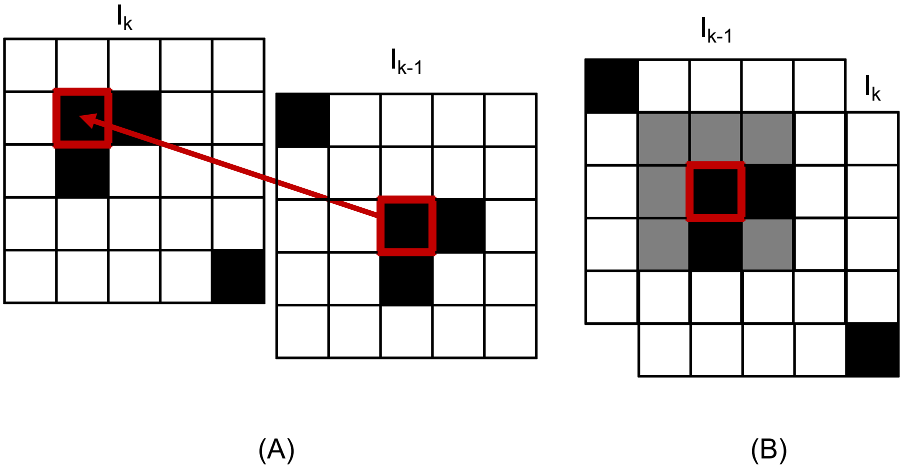


Figure 5.2: Feature matching: (A) Same feature detected in consecutive images I_k and I_{k-1}. (B) Evaluation of potential matches through normalized cross correlation with a 3x3 window.

feature descriptors. Figure 5.2 illustrates the matching and evaluation of a feature point through normalized cross correlation.

The next step, motion estimation, is the central computation step for every image in a VO system. It is responsible to compute the geometric relation between the previous frame I_{k-1} to the current frame I_k. There are three methods to estimate the camera motion, which vary according to dimension that feature correspondences are specified:

- 2D-to-2D: feature correspondences from both the previous and the current frames are specified in 2D image coordinates. Rotation and translation are directly extracted from the essential matrix that can be computed from 2D-to-2D feature correspondences using the epipolar constraint. The minimal case solution involves five 2D-to-2D correspondences.

- 3D-to-3D: feature correspondences from both the previous and the current frames are specified in 3D. This method require features to be triangulated using a stereo or RGB-D camera system. The camera motion can be computed by determining the aligning transformation of the two 3D feature sets. The minimal case solution involves three 3D-to-3D noncollinear correspondences.

- 3D-to-2D: features from the previous frame are specified in 3D, while from the current frame are expressed in 2D. the minimal case involves three 3D-to-2D correspondences. This method minimizes the image reprojection error instead of the 3D-to-3D feature position error. The minimal case involves three 3D-to-2D correspondences, and is referred in the literature as perspective from three points (P3P).

For a more detailed review of these methods we may refer to [253] and [211]. Both these methods result in estimations of the relative rotation and translation from the the previous camera position to its current position. Through the concatenation of all these relative movements, the full trajectory of the camera, and thus of the robot, can be recovered.

Independently if using classic wheel odometry or vision-based odometry, dead reckoning techniques are intrinsically subjected to motion drifts due to its inability to detect revisited locations. An alternative methodology consists on incrementally build and maintain a map of the environment by estimating the locations of the robot and of the mapped features. This task of computing camera motion from measurements of a continuously expanding set of visual features is referred in the literature as Simultaneous Localization and Mapping (SLAM).

The problem of doing so is that both the feature's observation and the robot localization are corrupted by noise. In [210] Lemaire *et al.* say that "in the absence of an a priori map of the environment, the robot is facing a kind of "chicken and egg problem": it makes observations on the environment that are corrupted by noise, from positions which estimates are also corrupted with noise". In other words, the errors in the robot's pose have an influence on the estimations of the observed feature's locations. Similarly, the use of observations of previously detected features to locate the robot, provide pose estimations that inherits from both errors. Because stochastic approaches explicitly handle sensor noise, they have demonstrated to deal with the SLAM problems in a consistent way. The implementation of a typical feature-based SLAM approach encompasses the following four basic steps:

- Feature detection. It consists in detecting in the perceived data, features of the environment that are salient, easily observable and whose relative position to the robot can be estimated. This process depends on the kind of environment and on the sensors the robot is equipped with: it is a perception process, that represents the features with a specific data structure.

- Prediction. Estimation of the robot motion between two feature observations. This estimate can be provided by sensors, by a dynamic model of robot evolution fed with the motion control inputs, or thanks to simple assumptions, such as a constant velocity model.

- Observation. Estimation of the feature location relatively to the robot pose from which it is observed.

- Estimation. This is the core of the solution to SLAM: it consists in integrating the various relative measurements to estimate the robot and landmarks positions in a common global reference frame. The stochastic approaches incrementally estimate a posterior probability distribution over the robot and landmarks positions, with all the available relative estimates up to the current time.

Besides these essential functionalities, one must also consider the map management issues. To ensure the best position estimates as possible and to avoid high computation time due to the algorithmic complexity of the estimation process, an active way of selecting and managing the various landmarks among all the detected ones is desirable. First the robot detects and initialize new landmarks on the map. In the second step, the robot predicts its motion with the associated increase of its position uncertainty. In the third step, the robot observes the previously mapped landmarks from a new (unknown) position. Finally, the robot corrects of landmark positions and estimates its localization, with the associated decrease of both robot and map uncertainties.

In a large sense, the main difference between VO and VSLAM is that the later concerns with obtaining a global, consistent estimate of the robot path. In order to obtain global consistency, VLSAM needs to keep track of a map of the environment and detect when the robot has returned to a previously visited location. When a loop closure is detected, this information is used to reduce the drift in both the map and the estimated trajectory. This makes VLSAM more complex and computationally more expensive than VO. Visual odometry, on the other hand, concerns with the local consistency of the trajectory. The path is computed incrementally, pose after pose, and potentially optimized only over the last n poses.

The choice between VSLAM and VO is a trade-off between precision and performance. While sometimes the VSLAM globally consistent trajectory is desirable, other times the VO real-time performance is a requirement. Finally, despite the successful results that have been obtained using VLSAM systems, most of them have been limited to small indoor workspaces, and only a few have recently been designed for large-scale areas.

Despite the several implementations proposed in the literate, vision systems can be classified in stereo and monocular approaches. Stereo approaches encompass the algorithms in which the 3D state of the observed features can readily be estimated from a single observation. The vast majority of existing vision-based localization approaches rely on data that directly convey the landmark 3D state, for example by matching points in the stereoscopic image pair. If the robot is endowed with a single camera, on the other hand, only the bearings of the features can be readily observed. The bearings-only problem is an instance of the more general partially observable system, in which the sensor does not give enough information to compute the full state of a landmark from a single observation.

Monocular cameras have the ability to measure the bearing of image features, but are not able to estimate depth. However, given an image sequence of a rigid 3D scene taken from a moving camera, it is possible to compute both a scene structure and a camera motion up to a scale factor. To infer the 3D position of each feature, the moving camera may observe it repeatedly each time, capturing a ray of light from the feature to its optic center. The measured angle between the captured rays from different viewpoints is the feature's parallax, and allows feature's depth to be estimated. Landmarks Initialization is a delicate task. Extended Kalman Filter requires Gaussian representations for all the random variables that form the map (the robot pose and all landmark's positions). Furthermore, their variances need to be small to be able to properly approximate all the non-linear functions with their linearized forms. From one bearing measurement, it is not possible to establish an estimate of the landmark position that satisfies this fundamental rule. Thus, it is only achieved through successive measurements from different points of view, when enough angular aperture has been accumulated. This reasoning leads to systems that have to wait for this angular aperture to be available before initializing the landmark in the SLAM map, which is known as delayed initialization.

Up to a recent past stereo rigs were the sensor of choice for obtaining visual and depth information. Dense 3D vision sensors were expensive and limited to just a few research groups. This reality changed, though, with the popularization of time-of-flight (e.g. CamBoard nano [254], SwissRanger SR4000 [255]) and RGB-D cameras (e.g. Kinect [256], Xtion [257], CARMINE [258]). Due to their small size, low power consumption, reliability and speed of the measurement, these sensors became the primary choice for 3D measuring in indoor robotics. In this context, RGB-D cameras are specially relevant since mass production made them broadly available at a very low cost.

Structured light RGB-D cameras are composite devices that consist of an RGB camera, an IR pattern projector, and an IR camera. The latest two are used in conjunction to triangulate points in space, working as a depth camera. Through a process called registration each pixel from the depth image is reprojected into the frame of the color image, so that the depth information correspondent to each RGB pixel is made available. For more information regarding the principles of operation of structured light RGB-D cameras and accuracy analysis we may refer to the works of Smisek *et al.* [259] and Khoshelham [260].

In this chapter, we describe a methodology to recover the trajectory of robotic devices. The proposed visual odometry algorithm makes use of affordable RGB-D cameras to provide motion estimations, and avoid the typical problems caused by the wheel-ground interaction that are encountered in the traditional wheel odometry. Through experimental analysis, we demonstrate that our RGB-D odometry can provide consistent localization over real world conditions.

The outline of the chapter is the following. Section 5.2 presents some related works in the area of visual odometry and visual simultaneous localization and mapping. Section 5.3 addresses the motion estimation algorithm proposed in this thesis. Section 5.4 describes the methodology used to evaluate the performance of the egomotion algorithm, as well as the results obtained from the estimation of a real world trajectory. Finally, the summary and conclusions of this chapter are presented in SectionSection 5.5.

5.2 Literature Review

The basic idea of estimating mobile robot's motion using on-board cameras can be traced back to 1980, when Moravec [261] presented his PhD thesis. In his work, the author describes a correspondence-based approach designed to track distinctive features over pairs of frames and build a 3D world model. The central idea of his algorithm was to match features detected in the recently acquired images with those from the world model, and so find their relative position to the robot pose. Later, Matthies and Shafer described a system that evolved from Moravec's work. In [7], they proposed the use of a three-dimensional Gaussian distribution, instead of relying on scalar models, to deal with error modeling in triangulation. According to their work, 3D Gaussians could reduce the variance in the robot position estimates.

Following Matthies work with some minor variations, stereo visual odometry is presented as an alternative for localization in the slippery rock surface and steep slopes present in mars terrain.

In [262], VO was applied during the operation of rovers in the mars planet. Due to computational constrains, the algorithm demanded 2-3 min to estimate each step of the rover motion. It reduced the use of visual odometry just for complex environments, like steep slopes and in soils propitious to wheel dragging. Despite of few instances in which estimation did not converge (i.e. due to too large motion or lack of features in the image), VO was able to provide the rover's motion estimate in more than 95% of the time. Among the several benefits provided by visual odometry the authors cite the vehicle safety (achieved by having the rover over the planned drives), the improved accuracy in new and mixed soil terrains (and so a greater number of science observation), and the reduction in the time needed to make targets reachable by the robot's instruments. Another work involving exploration rovers was presented by Helmick *et al.* [263]. Here, visual odometry is used to continuously compensate the wheel slippage of a Mars rover. At the same time the rover estimates its motion measuring the wheel rates (vehicle kinematics), through visual odometry and using an on-board Inertial Measurement Unit (IMU). Then visual odometry and the IMU estimates are merged through a Kalman filter, providing a motion estimate which is independent of the vehicle's interaction with the environment. Finally, the motion estimate from the Kalman filter is compared with the motion estimate from the vehicle kinematics to determine if any significant slippage has occurred. Thus, in case no slippage has occurred the kinematic estimate contribute to the Kalman filter estimate, otherwise, an "slip vector" is computed to compensate the rover trajectory.

Milella *et al.* [264] claim that a reduction of false matches can significantly improve the visual odometry accuracy. Therefore, they propose a 3 step match rejection solution based on the integration of nearest-neighbor-ratio and mutual consistency check with the iterative reckoning of 3D Euclidean transformation to remove outliers from the sample. Interest points in sequence images are compared through the Euclidean distance to the closest and second-closest neighbors – if the distance to the closest neighbor is significantly closer, then the match is accepted. Later, pairing processes are applied for both current and previous frames, and matches accepted only when they are mutually the preferred mate. Finally, an iterative process computes the rotation and translation matrix, computes the error of each match and selects (for refinement) those under a threshold.

For Olson *et al.* [265, 266], despite of presenting an accurate solution to estimate motion in short runs, the incremental nature of visual odometry algorithms expose it to the accumulation of errors over long distances. In his work, Olson describes several mechanisms to improve stereo ego-motion estimative over long distance navigations. Besides the techniques for increasing the robustness of feature selection, outlier removal and tracking, the author has demonstrated that even robust systems tend to accumulate errors. According to his research, the positioning errors tend to grow with the square root of the distance traveled, as well as with the integral of the orientation error - implying in a super-linear contribution that grows at $O(d^{\frac{3}{2}})$. With this in mind, he proposes the use of absolute orientation sensors (accelerometers, compass, and IMUs) to provide periodic updates to the orientation estimate, eliminating the super-linear error growth. Similarly, Levin and Szeliski [267] sustain that complementary sources of information (containing global positioning information) are needed to compensate VO inaccuracies. For this reason, she proposed a novel approach that matches up visual data with a hand-drawn map of the environment to correct global pose estimations. In addition, the algorithm concerns about path crossing detection – in which path-crossing points are used to constrain the possible matches between the visual odometry and the map, improving the quality of the pose estimations. In this paper, Howard [12] proposes a frame-to-frame algorithm for estimating a stereo camera's motion. Comparing to previous works, the main distinction of his research occurs in the algorithm for detecting inliers. After features are detected, the algorithm check its consistency using the assumption that a pair of any two features shall present the same distance (measured through 3D world coordinates) in the current frame $f(i)$ and in the previous frame $f(i-1)$. Thus, authors claim that the algorithm can cope with frames containing even 90%, which would severely compromise the results of other inliers detection algorithms – and so the VO estimates. However, after his results, Howard support that due to the limitations of pure visual odometry even the most minimal robot should have some form of proprioceptive sensing, and so, their algorithm is intended to extend instead of replace these sensors.

Besides those methodologies using stereo cameras, researches using monocular cameras have also presented promising results. For instance, Nistér *et al.* [268, 269] dealt with a stereo head, but have also presented a full structure from motion (SFM) algorithm to estimate ego-motion with a single camera. First, the algorithm detects interesting points in each frame using Harris corners detector. Then feature points are matched between pairs of sequential frames. After that, features are tracked over a certain number of frames, and relative poses estimated using the 5-point algorithm and preemptive random sample consensus (RANSAC). Finally, the observed feature tracks are triangulated into 3D points using the first and the last observations on each track. Making use of a learning methodology, Royer *et al.* [8] presents a different approach for outdoor robot navigation using only monocular vision. In his work, firstly the robot makes use of a learning phase - in which it is manually driven to record a video sequence of the predefined path (video reference). Later, from the video reference, a set of key frames are detected, and the camera motion computed. Then, a set of interest points are reconstructed in 3D, serving as landmarks for the localization process. Finalized the learning process, the robot can perform the same path, acquiring new images, detecting interest points on it and correlating them with the previous 3D reconstruction.

Another approaches based on the monocular perspective propose solutions for the motion estimation using omnidirectional cameras. In [270], Corke *et al.* proposed two alternatives for computing the monocular visual odometry of a planetary rover. In the first approach, they used optical flow computation with planar motion assumption, while in the second they computed an unconstrained SFM. Their results have demonstrated that the optical flow method is more robust to estimate the vehicle velocity, while the SFM provides better accuracy (with larger computational cost). Scaramuzza and Siegwart [271] also have chosen to deal with catadioptric cameras to compute real-time ego-motion. The main innovation of his work consisted in the application of two different trackers for the vehicle motion estimation. The first tracker was based on the coplanar correlation between two different views of the same plane (homography-based), and is employed to update the magnitude of the translation. The second tracker used an appearance-based approach (known as visual compass) to provide high-resolution estimates of the vehicle's rotation. According to his algorithm, every omnidirectional image is unwrapped into cylindrical panoramas and compared with the previous image. A column-wise shift of the best match between successive images is used to directly compute the rotation angle. His methodology can be summarized into five steps: acquisition of consecutive frames, extraction and matching of SIFT features, reckoning of the rotation angle through the appearance-based methodology, rejection of the outliers (through RANSAC), and finally estimation of the camera motion. Later, Scaramuzza *et al.* [216] exploits the planar motion and the nonholonomic constraints of the Ackerman steering geometry of automotive vehicles. These constraints reduce the vehicle's degrees of freedom to only two: rotation and radius of curvature. Thus, only one feature correspondence is necessary to compute the epipolar geometry, simplifying the motion reckoning and increasing its robustness in scenes in few structures. Similar to Scaramuzza, Civera *et al.* [272] also proposes a solution for visual odometry that requires just one matching for computing structure from motion. However, while in Scaramuzza's extra information comes from the application of restrictive motion models, in [272] the extra information comes from the probability distribution estimated by the EKF. Actually, as features are kept alive just while inside the camera's field of view, they are not re-localized the uncertainty of the camera pose computed by EKF estimators would always grow with respect to world reference frame. For this reason, the authors have used a sensor-centered EKF estimator, which represents features locations and camera position in a local reference frame.

One of the first works using vision as input for SLAM approaches is the work of Davidson [273], later published in [274]. This paper describes one of the first applications of real-time robot localization within a SLAM framework to use a stereo head as input. Davidson support that without building and maintaining a map of the environment (features that could be re-detected are treated as new) a progressive error accumulates proportionally to the distance traveled. Thus, a SLAM approach is used to propagate first-order approximations of probability density functions, representing uncertain estimates of the robot, the features and the relation between these estimates. Over the image, features are detected using the Shi and Tomasi variation of the Harris Corner detector. Then, regions where features are most likely to lie are computed and matches searched within these regions using normalized sum-of-squared-differences. Finally, robot motion is measured and the map managed in order to keep only a sparse set of reliable and distinct features. The previous work of Davison evolved to a more comprehensive approach. In [13], and more recently in [9], he proposes the use of EKF for solving the real-time motion estimation of a single camera. To deal with the lack of metric scale of the monocular cameras, Davison assume that the camera starts at rest, in front of a known object (with known dimensions and position). Then, the algorithm predicts the camera movement and searches, in a likely region, for features already in the SLAM map. Once feature depth can not be estimated from a single measure, for the feature initialization in the SLAM map Davison proposes a particle filter approach - creating a set of depth hypothesis, which is refined up to convergence. Despite handling with depth initialization, this approach did not solve entirely the problem. Actually, it established a significant limitation once the algorithm is able to deal just with features located in a small predefined range (from 0.5 to 5.0m) - limiting its application to room-scale scenes. Further, such "delayed" of initialization meant that observations of features were not used to update the camera pose estimate until their conversion into fully initialized features. Another problem of this approach consists in the use of the template matching technique for feature correspondence. Such technique, which uses filter predictions to solve matching ambiguity, has demonstrated a lack of stability in situations like camera shake, occlusion and erratic motion. Similarly, as features are viewed from wider angles, surrounding regions deviate from the templates and matching becomes unreliable, again resulting in failure.

Most of the monocular SLAM approaches rely on prior knowledge of the robot position to determine which features of the map select for matching with those from the current view and to reduce search area of each feature. Nevertheless, events like rapid camera motions, occlusions and motion blur, present in real applications, violate such assumptions and often cause tracking to fail, loss of camera pose, and even map corruption. Therefore, Williams *et al.* [10] presented a robust re-localization method which operates in parallel to Davison's MonoSLAM system [9]. His approach intends to increase the robustness against camera shakes and occlusions relying on the image-to-landmark matches yielded by randomized trees to recover the camera pose. Despite randomized trees breaks down the problem into several classes, the cost regarding class training and storage is an obstacle when dealing with large maps with hundreds of features.

Jensfelt *et al.* [275] has presented an approach for online mapping that combines Harris-Laplace and SIFT descriptor for feature detection. However, due to the delayed output of the tracking module current estimations of the robots position have to be predicted from the last pose of the SLAM module using odometry or dead-reckoning sensors. With robot moving further, more features are accumulated. Thus, to avoid the heavy one-to-all matching strategy, the author adopts a kd-tree to construct SIFT features descriptors. Nevertheless, with continuously added feature descriptors, kd-tree becomes larger and unbalanced, leading to a rapid decrease in the search efficiency. In the work presented by Lameire *et al.* [276], the main innovation concerns with the initialization of new features. According to his approach, the initial probability density of a feature is approximated by a weighted sum of Gaussians. Subsequent observations are used to compute the probability of each Gaussian and prune bad hypothesis (those with low likehood). Then, when only a single Gaussian remains, a new check verifies the feature consistency, which can definitely accept or reject the feature. Later, Lemaire at al. [210] compare the advantages of bearing-only and stereo approaches in a traditional SLAM framework. As in [276], authors support the use of delayed feature initialization in order to avoid useless computations of unstable features. Finally, the comparison between monocular and stereo remarks that bearing-only SLAM has demonstrated to be more sensitive to the prediction input, and so, incurring more errors in the position estimates.

A severe drawback of several monocular approaches regards to the fact that they can only cope with features located near the camera. In bearing-only approaches, the feature's depth is initialized measuring the angular difference between the light rays from the feature to the camera's optic center in different viewpoints (feature parallax). However, the depth of features with low parallax (features distant to the camera) is very difficult to be measured, so these features are usually rejected. According to Civera *et al.* [272], while features with high uncertain depths do not provide much information about the camera translation, they are very useful reference for the camera rotation. Thus, they propose a novel parameterization for point features represented by a 6D state vector that encode uncertainty up to infinity with one Gaussian. In addition, with the inverse depth parameterization (IDP), features can be immediately used to improve camera estimations through an immediate initialization process. However, the use of IDP is more computational consuming, because instead of the three dimensions of a Euclidean representation each point is represented by the 6D state vector.

Current visual SLAM systems based on EKF framework suffers from a severe limitation concerning the size of the environment they can self-localize. Actually, real-time is mostly achieved in maps with up to 100 features. Thus, Paz *et al.* [277] proposes to divide a large environment into a set of several small maps. This way, their algorithm (called Divide and Conquer – DεC) is able to compute the covariance of probabilistic maps from large environments in a reduced amount of time, improving consistency of the resulting estimate. However, this approach requires local maps to be statistically independent, and consequently, it is not possible to share important information, such as the camera velocity, or information about features currently being tracked. Later, in [11] Paz *et al.* proposed an alternative for improving the DεC approach. Rather than dealing with statistically independent maps, this novel approach extent the previous research working with conditionally independent maps. As a result, worth information is shared between maps, without increasing computational costs or loosing accuracy.

A further novelty of this work consists in combining of stereo and monocular approaches. In doing so, it is possible to extract the best of each paradigm. The stereo approach is able to provide depth information for features at a close range, and thus compute the robot translation. In addition, with stereo it is possible to observe the true environment scale, eliminating the intrinsic scale unobservability of monocular system – scale does not need to be initialized using external sensors, nor through a priori knowledge as the size of a known object or the initial camera speed. However, monocular approaches are able to deal with points much more distant to the camera (even points at the infinity). Distant features act as bearing references, providing better estimation of the camera rotation. They are also very important in outdoor environments, which features are dismissed by stereo heads due to their inability to estimate depth with reasonable accuracy. To take advantage of both types of information, their system combines in the map the 3D points provided by the stereo par (defined by Cartesian coordinates) with Inverse Depth (ID) points [272] provided by the bearing-only algorithm.

In [278], Chekhlov *et al.* discuss about the problems regarding feature descriptors used for motion estimation. They observe that despite minimizing computational efforts, template-matching techniques are affected by the size of search region (which are effective only for small search regions) and by the viewing angle (large variations infer matching problems). Although warping template techniques propose a solution for improving the robustness over large viewing angle variations, it suffers from a widespread mismatch in situations of sudden erratic motion and camera occlusion. For this reason, the authors have proposed a novel methodology to operate over a large range of views by using SIFT-like feature descriptors. Estimates of camera motion (using unscented Kalman filter) are computed to guess the scale and to speed up the computation of the features. Also interested in alternatives for detecting interest feature, Tomono [279] proposes a SLAM approach based on the Iterative Closest point (ICP) algorithm and edge detection to estimate the motion of stereo pair. The author discusses the stability of corner-like approaches in non-structured scenes. According to his research, corner-like algorithms can frequently detect an insufficient amount of features for robust localization in man-made environments, and so, other alternatives like the lines and edges may be explored. The camera motion is estimated in two steps. In the first, motion is estimated with visual odometry. Then, a key frame adjustment is employed to re-estimate motion and reduce the accumulated errors. Despite the dense map resulted from this approach, the huge amount of edge points detected on each frame (up to 10,000 according to the author) demands special care with processing efficiency. Another related to edges is the low distinguishability of edge points, and the large number of multiple matching candidates.

Since 2010, with the advent of the low cost RGB-D cameras, a new set of localization methodologies have been proposed in the literature. Henry *et al.* [280] presented one of the first works using RGB-D sensors for localization and mapping. Their system used both sparse visual features and dense point clouds for frame-to-frame alignment and loop closure detection. First it extracts SIFT features and, through RANSAC, estimate the image transformation. This transformation is than used in the initialization of the ICP dense estimation. The biggest limitation of the system, high computational cost, is evidenced in their experiments. With a reported computational time close to 1 second per frame, the approach in the presented configuration is not able to provide real-time pose estimations.

Endres *et al.* [281] presented the VSLAM approach referred in the literature as RGB-D SLAM. The algorithm uses a SIFT to extract visual key points from the color images. Next, the algorithm uses FLANN to match features between consecutive frames, and the depth images to localize them in the 3D space. Finally, the algorithm estimate the transformation between the two frames with RANSAC, and optimize the pose graph using non-linear optimization. With the elimination of the ICP step of Henry's approach and the use of a parallelized GPU implementation of SIFT, Endres could significantly reduce in the computational time. Experimental results using TUM RGB-D dataset [282] demonstrated that, on average, the frame processing time was of 0.35s.

Paton and Kosecka [283] suggested an RGB-D localization system (referred to as Adaptive RGB-D) that selects the motion estimation method according to the estimation reliability. The algorithm is mainly based on the detection and matching of sparse SIFT features. Though, when the RANSAC motion estimations fail or present a high residual error, it takes advantage of the dense point clouds provided by the RGB-D depth images to refine the motion estimation with ICP. Experimental results using TUM RGB-D dataset demonstrated that, when compared to RGB-D SLAM, the algorithm provided inferior results in datasets with many loops and rich in visual features (reported to be due to its lack of a global optimization), and better results in datasets with limited matching features and absence of loops.

Recently, Kerl *et al.* [284] presented an approach that minimizes the photometric error, and

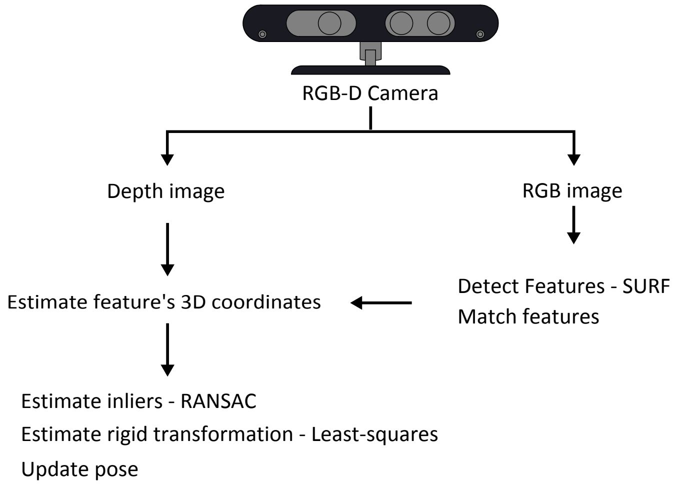

RGB-D Camera
Depth image
RGB image
Detect Features - SURF
Match features
Estimate feature's 3D coordinates
Estimate inliers - RANSAC
Estimate rigid transformation - Least-squares
Update pose

Figure 5.3: Decomposition of the proposed RGB-D odometry.

estimates the camera motion using non-linear minimization. A great advantage of eliminating the sparse feature detection and matching step is the reduction in the computational cost, allowing the algorithm to run in real-time (30 Hz) on a single CPU core of a Intel i5 (3.46GHz). Experiments performed on the TUM RGB-D dataset reported a reduction in the root mean square error when compared to the RGB-D SLAM approach.

5.3 RGB-D Based Motion Estimation

Our motion estimation methodology relies on the data provided by RGB-D sensors to estimate the robot motion and localization. A general flow of the proposed algorithm is presented in Figure 5.3. The algorithm starts with extraction and description of SURF image features. SURF was chosen not only due to its invariance to changes in scale, rotation and point of view, but specially because its computational efficiency allows real-time implementations.

Features from the current and previous scenes are then evaluated in order to find the appropriate correspondences. Our implementation uses a brute force matcher, in a process that compares all the features from the current set with every feature from the previous one. The matcher can, therefore, find the pairs of features that contain the closest descriptors. In addition, the algorithm performs a cross check in order to return only consistent pairs. In other words, it only returns pairs of descriptor (i, j) whose $i-th$ element is the nearest descriptor of $j-th$ if the $j-th$ element is also the nearest descriptor of $i-th$.

After the matching step, we project the 2D feature locations from the image to 3D using the pixel depth information. The RGB-D sensors have a factory calibration stored on-board, based on a high level polynomial warping function. The OpenNI [285] driver uses this calibration to compensate radial and tangential lens distortion, and for registering the depth images (taken by the IR camera) to the RGB images. The conversion from the 2D images to 3D point clouds are defined as follows:

$$Z = D(v,u) \tag{5.1}$$

$$X = \frac{(u - cx)Z}{fx} \tag{5.2}$$

$$Y = \frac{(v - cy)Z}{fy} \tag{5.3}$$

where $D(v,u)$ is the pixel value of the depth image, (v,u) the 2D image coordinate, (X,Y,Z) the 3D image coordinate, cx and cy the camera optical center, and fx and fy the camera focal length.

In theory, the transformation of the camera pose between two frames could be computed in closed form from the 3D point correspondences [286]. However, due to noise in the 2D feature localization and mismatch in feature pairs it is not possible to assure a perfect reliability with respect to repeatability and false positives. Noise in the estimation of the features 3D location also contributes to make the robust estimation of transformations highly non-trivial. Inconsistencies between depth data and the RGB image are a common issue due to a lack of synchronization between the shutters of the infrared and the color camera. Another important cause of inconsistencies rely on the interpolation at depth jumps, which occur since visually salient points often lie at object borders.

In order to deal with such noise data and remove the outliers we make use of the Random Sample Consensus (RANSAC) algorithm. After matching the feature descriptors of two frames, we randomly select three matched feature pairs (minimal number of points from which a rigid transformation in SE(3) can be computed). Outliers are than avoided by refusing sample sets for which the pairwise Euclidean distances do not match. When the samples pass this test, they are used to estimate the camera rigid transformation. This transformation is applied to all matched features, and the features within a fixed threshold distance are counted as inliers. The choice of the threshold is related with the random error of depth measurements. In our experiments, we considered a threshold of 4cm since it is the standard deviation at the maximum range of the sensor [260]. These steps are iterated and the transformation with most inliers is kept. The number of iterations, 35 RANSAC hypothesis in our experiments, was defined according to the Equation (2.52), considering the minimal set of data points $s = 3$, a confidence level $p = 0.99$, and an inlier ratio $\omega = 50\%$.

Next, all the inliers are used to compute a refined transformation. For that, we relied on the Umeyama's methodology [286] to estimate the transformation parameters between the pairs of 3D points (Y_i and X_i). The algorithm is based on the analysis of the covariance matrix, and estimates the rotation R and translation T that minimizes the mean square error e^2 of the input point sets:

$$e^2(R,t) = \frac{1}{n}\sum_{i=1}^{n} ||Y_i - (R\,X_i + T)||^2 \tag{5.4}$$

Algorithm 3 RGB-D based motion estimation

Definitions:

rgbImg: 3 layer matrix containing the red, green and blue values of image pixels
depthImg: matrix containing the depth information of every pixel of the RGB image
pose : vector containing the camera's x, y and z position in the world reference frame
orient : vector containing the camera's rotation quaternions in the world reference frame

```
1: function RGBODOMETRY($pose, orient, prev2Dkpts, prev3Dkpts, rgbImg, depthImg$)

2:     $2Dkpts \leftarrow LSACSurf(rgbImg)$          ▷ Detection of visual salient points

3:     $matches \leftarrow bruteForceMatch(2Dkpts, prev2Dkpts)$
4:     $3Dkpts \leftarrow convert2D23D(matches, 2Dkpts, depthImg)$

5:     $inliers \leftarrow RANSAC(matches, 3Dkpts, prev3Dkpts)$
6:     $trans \leftarrow Umeyama(inliers)$     ▷ Estimate the camera's relative translation and rotation

7:     $pose \leftarrow updatePose(trans)$
8:     $orient \leftarrow updateOrient(trans)$
         return $pose, orient, 2Dkpts, 3Dkpts$

9: end function
```

where $i = 1, 2, ..., n$, n is the number of point pairs. Finally, from the rotation matrix and the translation vector, the camera pose $X(k) = [x, y, z]^T$ can be updated at each time step k:

$$X(k) = R\,X(k-1) + T; \tag{5.5}$$

The pseudo-code for the RGB-D based motion estimation algorithm is detailed in the Algorithm 3. First the algorithm receives the previous camera's position and orientation, previous 2D and 3D key points, as well as the current RGB and depth images (line 1). From the RGB image, it extracts 2D salient key points with LSAC SURF (line 2). These key point are than matched with the key points from the previous frame by brute force (line 3). The 3D position of the key points that were successfully matched are estimated using their 2D location and the depth image (line 4). Following, we use RANSAC to find the 3D inliers (line 5) and Umeyama's methodology to provide a refined estimation of the camera's rotation and translation (line 6). The final step is to update the camera's pose and orientation (lines 7 and 8). The algorithm is called every time a new RGB-D image pair is available.

5.4 Experiments and Results

In this section, we characterize the performance of our RGB-D odometry algorithm on a large image dataset. In addition, we investigate the influence of different choices of feature detectors over the accuracy of trajectory estimated by our odometry algorithm.

5.4.1 Dataset

The RGB-D odometry approach proposed in this Chapter has been tested on a dataset of real world images captured by an RGB-D camera. The TUM RGB-D dataset [282] contain sequences of pre-registered color and depth images along with the ground truth trajectory of the camera. Both images were captured by a Microsoft Kinect at 30 Hz with a full resolution of 640x480 pixels. The ground truth data was collected from a high-accuracy motion-capture system at 100 Hz, and contains the translation and orientation of the optical center of the color camera with respect to a fixed coordinate system. In addition, the dataset provides the intrinsic parameters of the color and infrared cameras (focal length and optical center), a correction factor for the depth values, and a tool to evaluate the accuracy of an estimated trajectory.

For evaluation, we chose the FR1 xyz and FR2 xyz sequences because they contain a typical real world indoor environment. The main characteristics of each sequence are summarized in the Table 5.1. As can be noted from this table, the average camera velocities range from 1.7 deg/s to 8.9 deg/s and from 0.05 m/s to 0.24 m/s.

Table 5.1: Characteristics of the TUM RGB-D dataset sequences.

Sequence	Length	Duration	Avg. Ang. Speed	Avg. Trans. Speed	N^o Frames
FR1 xyz	7.112m	30.00s	8.920^o/s	0.244m/s	796
FR2 xyz	7.029m	122.74s	1.716^o/s	0.058m/s	3665

5.4.2 Evaluation Metrics

The evaluation regarding the quality of the estimated trajectory was performed using two common evaluation metrics: relative pose error (RPE) and absolute trajectory error (ATE) [282]. The relative pose error measures the local accuracy of the trajectory. It computes the error in the relative motion between subsequent pairs of frames, making it well-suited for estimating the drift of visual odometry systems. The end result of the RPE evaluation method is the root mean squared error (RMSE) of the relative pose errors summed over the entire trajectory. Results of this evaluation method can be separated into translational and rotational errors, which are respectively defined as:

$$RPE_{Trans} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}||T_i - \hat{T}_i||^2} \tag{5.6}$$

$$RPE_{Rot} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}||R_i - \hat{R}_i||^2} \tag{5.7}$$

where n is the number of camera poses, T_i and R_i are the camera relatives translation and orientation at time step i, and $\hat{T}_i$ and $\hat{R}_i$ are the camera truth translation and orientation given by the

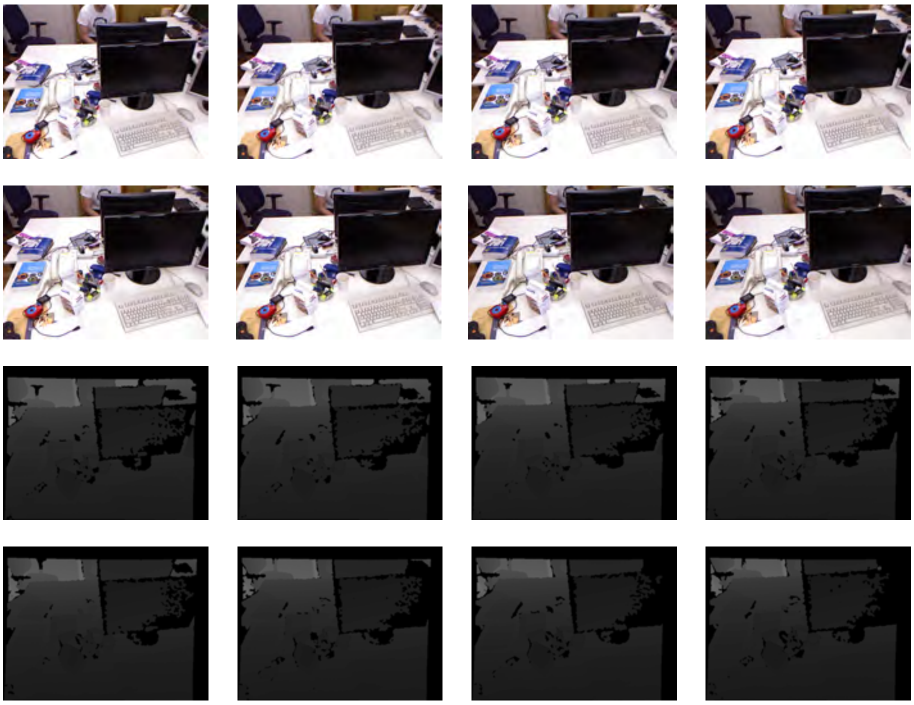


Figure 5.4: Sample RGB (upper rows) and depth (bottom rows) images from the FR1 xyz sequence.

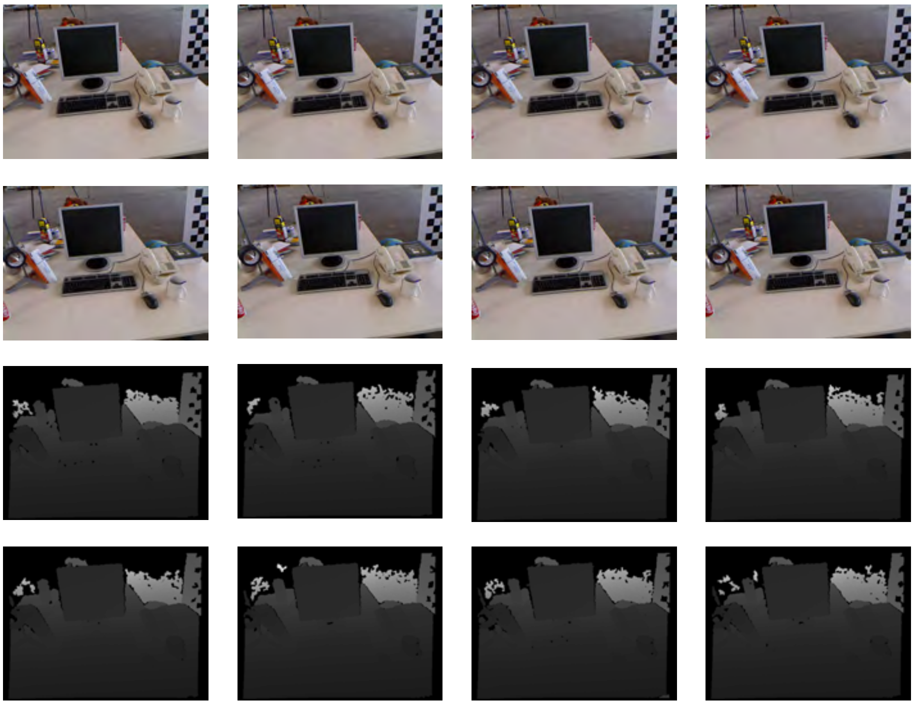


Figure 5.5: Sample RGB (upper rows) and depth (bottom rows) images from the FR2 xyz sequence.

ground truth. Note that, since rotational errors also manifest themselves in wrong translations, some researchers find it sufficient to evaluate RPE translational component.

The absolute trajectory error (ATE), on the other hand, measures the difference between poses given by the estimated trajectory and the ground truth. Since ATE evaluates the global consistency of the estimated trajectory, it specially suited for measuring the performance of visual SLAM systems. The end result of the ATE evaluation method is the root mean squared error (RMSE) of the global pose errors summed over the entire trajectory. The absolute trajectory error can be computed as:

$$ATE_{Trans} = \sqrt{\frac{1}{n}\sum_{i=0}^{n} ||P_i - \hat{P}_i||^2} \tag{5.8}$$

where n is the number of camera poses, P_i is the estimated camera pose at time step i, and $\hat{P}_i$ is the camera truth pose given by the ground truth. The literature report that these two metrics are strongly correlated. Nevertheless, from a practical perspective, ATE method has an intuitive visualization which facilitates visual inspection [282].

5.4.3 Experiment Results

In the first round of experiments, we evaluated the accuracy of our system on all sequences using SURF feature extraction. All results were obtained on a single core of a PC with Intel i7 2630(2.00GHz) and 4GB RAM. The ATE and RPE results from the FR1 xyz and FR2 xyz sequences are summarized in the Table 5.2. On these sequences, we obtain respectively 6.3cm and 10.7cm RMSE error. The mean runtime per frame were of 0.217s and 0.220s. Further, we analyzed if the feature detector proposed in the previous Chapter could improve the accuracy of the estimated trajectory. For that, we repeat the same experiment (round two) using LSAC SURF to extract visual salient points.

The ATE and RPE results from the FR1 xyz and FR2 xyz sequences are summarized in the Table 5.3. With LSAC SURF we improved the RMSE error in 1.8cm and 3.5cm, respectively. Yet, our approach did not compromise the algorithm computational cost. The mean runtime per frame with LSAC SURF were respectively 0.222s and 0.224s, only 2.3% and 1.9% slower than with the original SURF algorithm. Figures 5.6 and 5.7 show the translational error for each frame of the FR1 xyz sequence for the trajectories estimated with SURF and LSAC SURF respectively. In Figure 5.8, the trajectory is split into separate plots for the x, y and z component with a graph for the ground truth position (red), the position estimated with SURF(blue) and the position estimated with LSAC SURF (green). From that, it is clear that the algorithm is able to recover from bad motion estimations. This is only possible because the sequence contains several loops, and thus, the camera truth position is able to eventually meet the estimated position.

Figure 5.9 shows the translational error for the trajectory estimated with SURF on the FR2 xyz sequence. We can note that estimations close to frames 250 and 1000 presented a significant translational error (0.085m and 0.06m). A closer inspection indicated that such errors occurred due

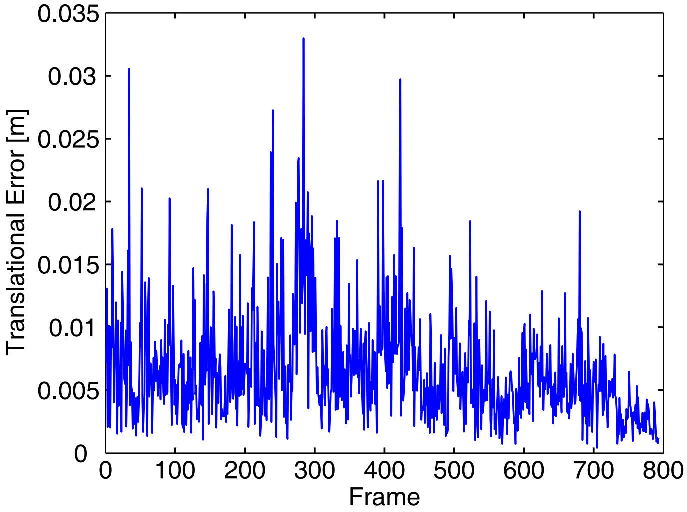


Figure 5.6: Translational relative pose error of the estimated trajectory in FR1 xyz sequence with the original SURF detector.

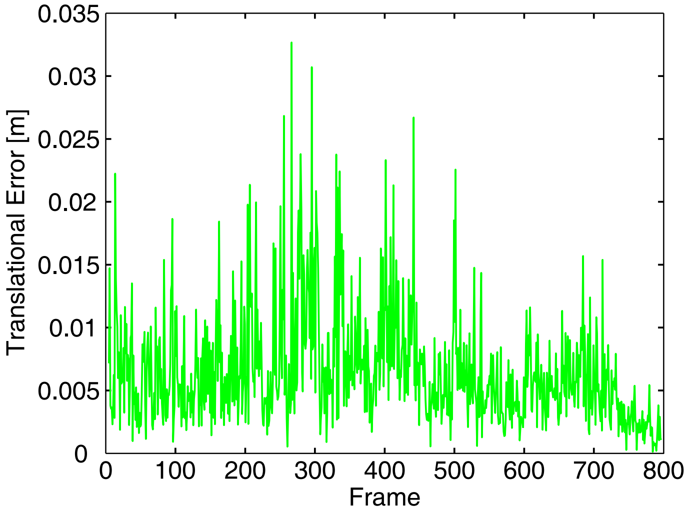
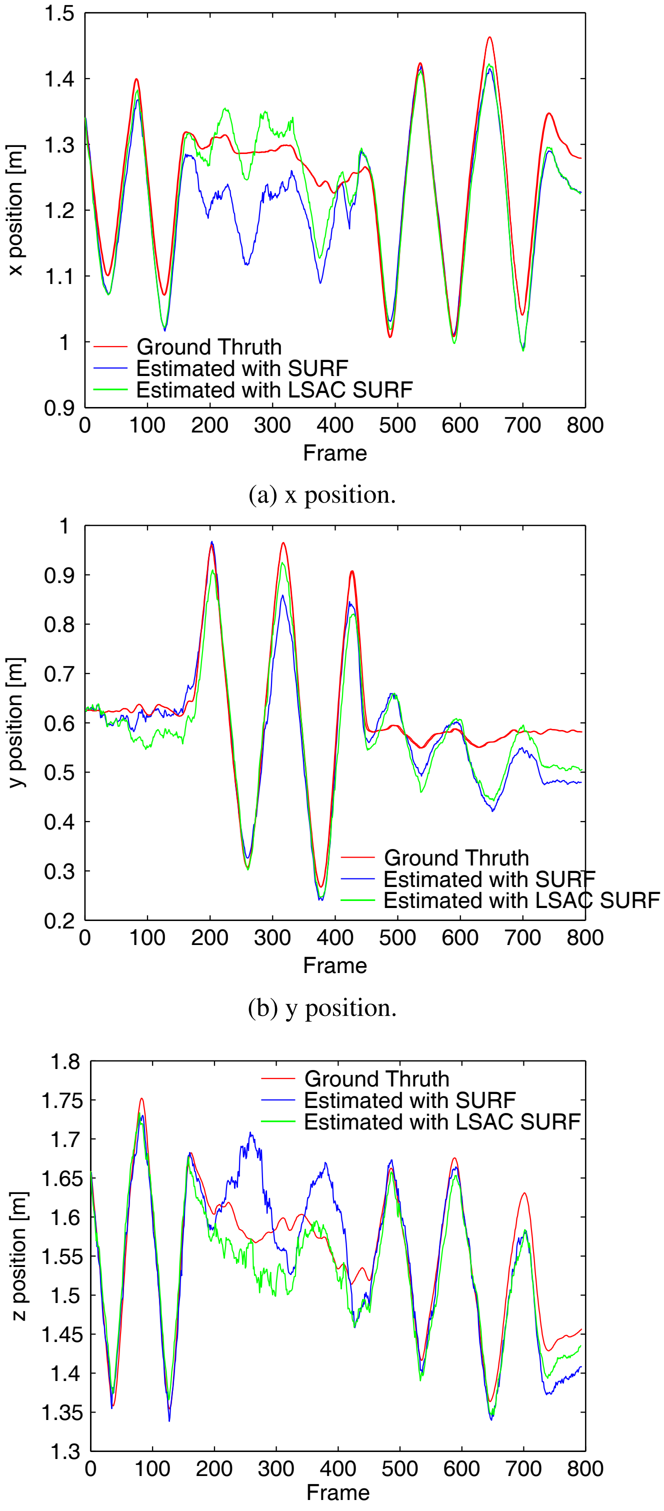


Figure 5.7: Translational relative pose error of the estimated trajectory in FR1 xyz sequence with the LSAC SURF detector.



(a) x position.

(b) y position.

(c) z position.

Figure 5.8: Estimated vs. ground truth trajectories of the FR1 xyz sequence. The red line represents the truth position of the camera, while the blue and green represent respectively the trajectory estimated with SURF and LSAC SURF detectors.

Table 5.2: Evaluation of the proposed visual odometry approach. On the two test sequences, the system feature detection was performed with the original SURF detector.

Sequence	Absolute trajectory error	Relative pose error		Total
	Trans. RMSE	Trans. RMSE	Rot. RMSE	Runtime
FR1 xyz	0.063166m	0.007434m	1.046866°	173.184s
FR2 xyz	0.107393m	0.005533m	1.337283°	806.078s

Table 5.3: Evaluation of the proposed visual odometry approach. On the two test sequences, the system feature detection was performed with the LSAC SURF detector.

Sequence	Absolute trajectory error	Relative pose error		Total
	Trans. RMSE	Trans. RMSE	Rot. RMSE	Runtime
FR1 xyz	0.044879m	0.005581m	1.003705°	177.167s
FR2 xyz	0.072615m	0.004454m	1.068909°	821.978s

to noise in the estimation of the 3D points, leading to a very low number of inliers and consequently a bad estimation of the rigid transformation. Figure 5.10, on the other hand, shows the translational error for the trajectory estimated with LSAC SURF. As in the previous sequence, Figure 5.11 split the camera trajectory into separate plots for the x, y and z component. The red line represents the truth position of the camera, while the blue and green represent respectively the trajectory estimated with SURF and LSAC SURF detectors. Similar to the previous observation, the visual odometry algorithm is able to recover from bad motions estimations due to the loops of the camera trajectory.

5.5 Conclusions

In this Chapter we described the most relevant methodologies in the area of feature-based visual localization. Visual odometry systems have the advantage of coping with hundreds (or even thousands) of features in each frame. Furthermore, in trajectories in which the robot continually explores new regions without returning, visual odometry can obtain as much precision as typical SLAM systems, but with a significant reduction in computational cost. However, just like the conventional dead-reckoning, VO is affected by the accumulation of errors over the time, and thus eventual drift in is inevitable. In addition, recovering a forward motion still presents a challenge to existing VO algorithms, partly due to the limited lifetime of feature tracks [3].

Visual SLAM systems, on the other hand, allow repeatable long-term localization through naturally occurring landmarks. Using cameras with a wide angular range ensures that persistent features re-detected after lengthy neglect can also be re-matched, even if the area is passed through along a different trajectory or in a different direction. This is key to reduce the effect of motion drift: in VSLAM the drift depends on the distance from the origin (in the world reference frame)

[3] The most informative features are at the boundaries of an image. Thus, they quickly move out of the camera's FOV.

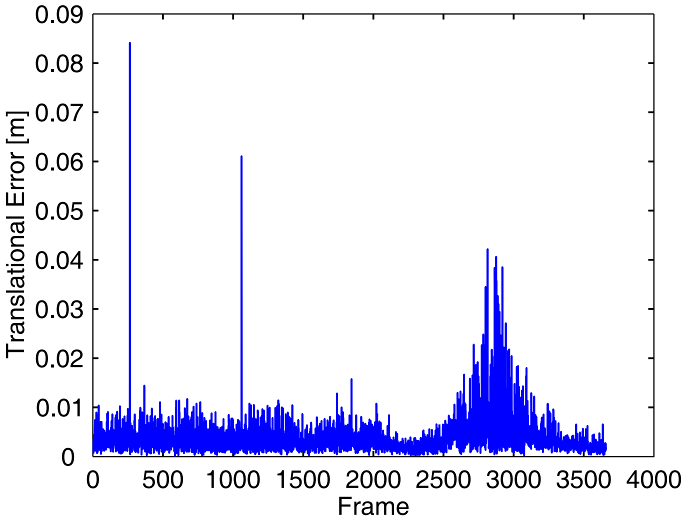


Figure 5.9: Translational relative pose error of the estimated trajectory in FR2 xyz sequence with the SURF detector.

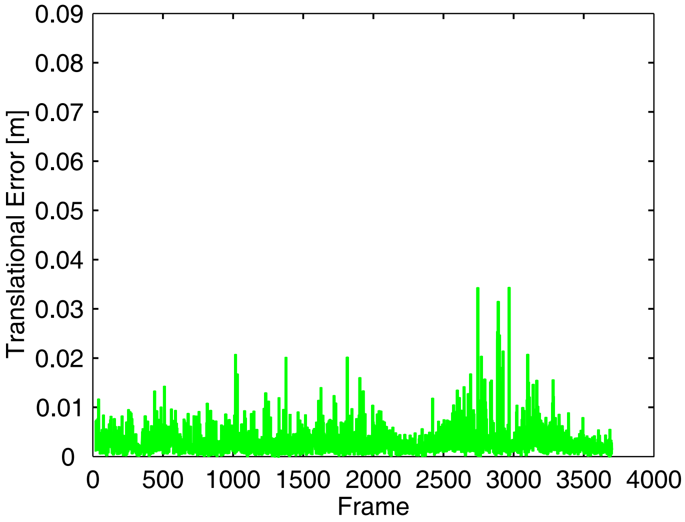
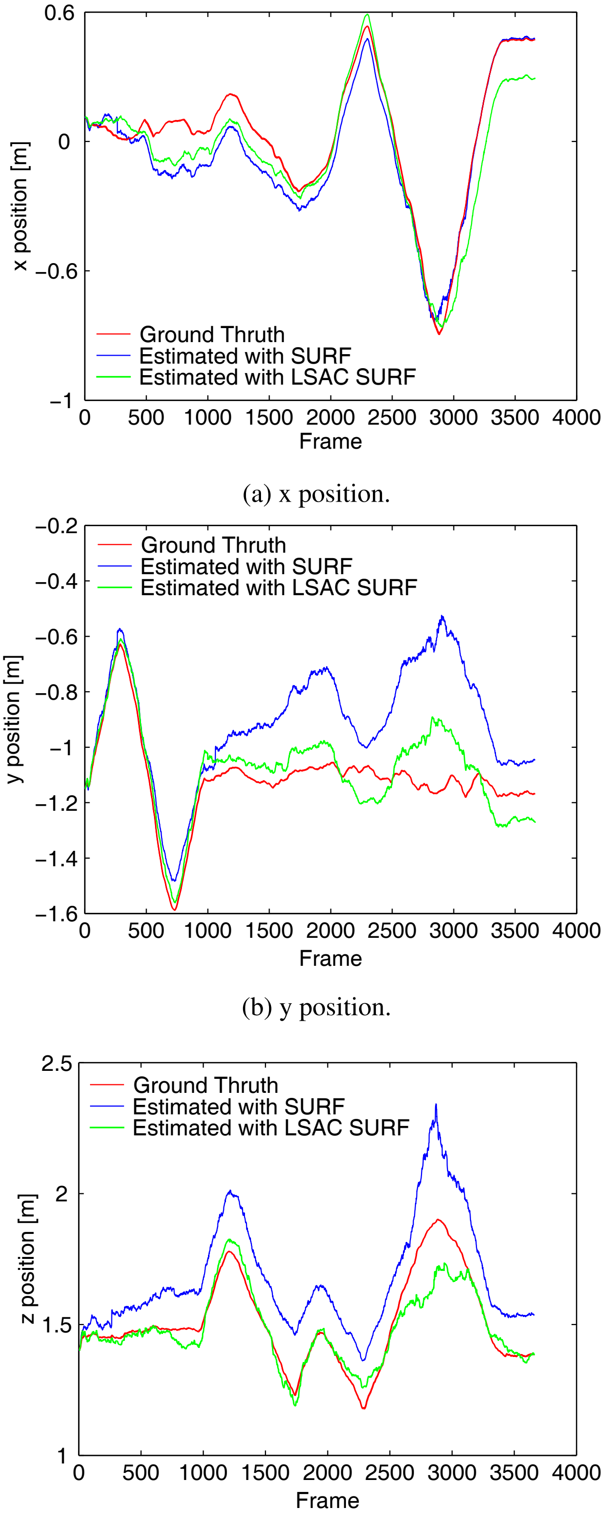


Figure 5.10: Translational relative pose error of the estimated trajectory in FR2 xyz sequence with the LSAC SURF detector.



(a) x position.

(b) y position.

(c) z position.

Figure 5.11: Estimated vs. ground truth trajectories of the FR2 xyz sequence. The red line represents the truth position of the camera, while the blue and green represent respectively the trajectory estimated with SURF and LSAC SURF detectors.

and not on the total distance traveled by the robot. Furthermore, VSLAM is able to derive on-the-fly probabilistic estimation of the camera and features, and benefit from this to improve processing efficiency.

We have also seen that, when dealing with vision-based localization systems, comparisons between stereo versus monocular approaches seems inevitable. A significant advantage of the stereo scheme is that it can operate correctly even with slow and no camera motion. This is also an indication of its greater stability, since many of the difficulties in monocular ego-motion estimation are caused by small motions. Stereo vision gives more information, because the overall scale of the motion is immediately known through the baseline of the stereo head. It makes easier to integrate visual odometry with information from other sources and also to preserve the correct scale of motion past breakdowns in the motion estimations. In the case of monocular approaches, an important limiting factor is that scale is not observable. Thus, scale must be initialized using some a priori knowledge about such as the size of a known object visible at the start or the initial speed of the camera. However, in large environments, unless scale information is injected on the system periodically, the scale of the map can suffer from a slow but continuous drift. Conversely, monocular approaches can cope with features on any depth, once even features at infinity can improve the orientation estimates. In addition, monocular cameras are less restrictive and easier to integrate in robots and portable devices.

Further, we proposed a odometry system based only on visual information. Our system relies on the color and depth data provided by inexpensive RGB-D cameras to compute the robot relative motion between pairs of frames and recover its full trajectory. Our algorithm starts by localizing visually salient points on the gray scale image. After matching these features over consecutive images, the algorithm uses the depth information to estimate their 3D position on the space. Through a probabilistic approach (RANSAC), it removes features whose 3D were corrupted by noise and computes the robot translation and rotation that minimizes the mean square error.

A set of experiments with a real-world image dataset were performed to evaluate the proposed approach. The results of such tests resulted in an absolute trajectory error of around 2% for 3D trajectories of 7m. In face of that, we believe on the applicability of the localization algorithm in the navigation system of intelligent wheelchairs (and mobile robots in general). Further we investigated the influence of the feature detector in the motion estimation. The accuracy of the estimated trajectory was superior for LSAC SURF in the two sequences tested, indicating that the hypothesis that the wider illumination invariance of LSAC SURF could improve the motion estimation is indeed verified. It is interesting to salient that these results were achieved on a dataset with low illumination challenges, and thus LSAC SURF is likely to provide even better results in other scenarios. We also believe that LSAC SURF tend be even more relevant to VSLAM systems. Since in VSLAM features may be revisited and re-matched after long periods, it is likely that variations in the scene illumination are more significant. However, as previously discussed, odometry is inherently affected by the accumulation of errors an so it is not expected to work as the only localization methodology. Instead, it designed to integrate a wider pose estimation approach that fuses the visual odometry data with other sources of information such as wheel odometry, global navigation systems, inertial navigation systems, etc.

Chapter 6

Conclusions

> "All truths are easy to understand once they are discovered; the point is to discover them."
>
> – Galileo Galilei

This chapter summarizes the contributions of this thesis, its limitation and future directions of research. For wheelchairs to be able to work for people with special mobility requirements, they need to be able to acquire knowledge through perception. In other words they need to collect sensor measurements from which they extract meaningful information. This thesis covered some of the essential components of a intelligent wheelchair system, adapting some methodologies developed for traditional mobile robotics to assistive devices:

- hardware framework: how to provide sensing and processing capabilities to regular powered wheelchair preserving the wheelchair ergonomics and its normal operation.

- obstacle avoidance: how to prevent collisions without full navigation autonomy (sharing the wheelchair control with the user).

- feature extraction and matching: how to extract and robustly match distinctive features that are suitable for robot motion estimations.

- RGB-D odometry: how to recover the trajectory of a camera/wheelchair using RGB-D cameras as the only input.

6.1 Main Contributions

The main contributions of this thesis are concentrated in two areas, respectively assistive robotics and localization. In the area of assistive robotics, an initial literature review revealed that the user welfare (normal wheelchair operation, ergonomics, accessibility, etc) is not considered in the design of most intelligent wheelchair projects. Since providing assistance to impaired people should be always the major objective to any intelligent wheelchair project, we designed our solution with a user-centered perspective. This work contributes to the conceptualization and development of a modular platform for the development of intelligent wheelchairs that mitigates the visual and ergonomic impacts caused by sensors and other computational devices. Our approach also provide compatibility with the hardware of multiple models and manufactures of powered wheelchairs, facilitating the conversion of regular powered wheelchair into intelligent wheelchairs.

Another work described here, the assessment of robotic simulators, provided an important contribution for the implementation of the IntellWheels simulator. Based on this study, a very realistic model of the wheelchair prototype was created. Throughout the project, the simulated environment was used to test IntellWheels flexible multimodal interface not only in able-bodied individuals, but also in patients with cerebral palsy. The experiments also allowed concluding that the simulator was an important tool to test and train the users to drive the intelligent wheelchair.

The literature reports that, in some obstacle avoidance methodologies, users preferred the manual control even when the shared control effectively reduced their number of collisions. A possible explanation to this observation is that some obstacle avoidance algorithms present a behavior that diverges from the user expectations. In this sense, higher acceptance rates are important, otherwise users tend to abandon the shared control methodology and increase the risk of collision. This was a concern when designing the shared control algorithm proposed in this thesis. Our algorithm demonstrated that it is able not only to reduce the number of collisions, but also to improve the user perception of assistance. An additional advantage concerns with the algorithm constant and low computational complexity, which allows real-time operation in embedded systems with limited computational capabilities.

Regarding the robot localization, this work focused on the extension of some methodologies used in visual-based localization systems. First, we can point the LN and the LSAC extensions of the SURF detector. Our contribution consisted in the combination of SURF feature detection algorithm with other computer vision techniques (respectively the local normalization and the local space average color) in order to improve its photometric invariance properties. In this context, the LSAC SURF algorithm showed to be particularly interesting due to its significant increase in feature illumination invariance and robustness to image noise. In addition, our RGB-D odometry demonstrated to be a viable option to deal with motion estimation tasks.

6.2 Limitations

The main limitation of this work is probably related to the inner nature of odometry, since the assumption that the robot path can be recovered through the integration of small consecutive estimations is not fully verified in practice. The noise present in real world sensors inevitably lead to noisy estimations which, in turn, leads to motion drifts and global inconsistencies. Nevertheless, the proposed RGB-D odometry algorithm can provide relevant pose estimations that can be fused with other global localization methodologies, or serve as base for a future visual SLAM approach.

Another limitation is related to our feature-based motion estimation approach. For the algorithm to work effectively, it should be able to extract a sufficient number of visual features. In this context illumination plays an important role. Saturation and interference with IR projected pattern lead to a severe reduction in the number of features detected, which consequently affects the robustness of the estimation. The scene must be predominantly static, with enough texture to allow apparent motion to be extracted. Furthermore, consecutive frames should be captured by ensuring that they have sufficient scene overlap, to allow proper matching. Finally, only those features located within the depth range of the RGB-D camera can have their 3D position estimated. For some frames, many detected visual features are out of range of the depth sensor, so those features have no associated 3D points and do not participate in the motion estimation procedure. Also, when the majority of the features lie in a small region of the image, they do not provide very strong constraints on the motion.

6.3 Final Remarks

Throughout this thesis we presented a series of methodologies and experiments to allow us test two hypotheses regarding the development of intelligent intelligent wheelchairs. The first hypothesis concerned with the design of intelligent wheelchairs, and stated that *"It is possible to design an intelligent wheelchair to assist severely handicapped individuals using low cost off-the-shelf devices without interfering with the wheelchair normal operation, and with reduced visual impact"*. The second hypothesis, on the other hand, concerned with the use of vision algorithms as means to improve the wheelchair localization, and stated that *"The use and extension of current vision-based methodologies can provide robust localization for intelligent wheelchairs"*.

Revisiting the conclusions of Chapter 3 it is possible to verify the first hypothesis. The intelligent wheelchair prototype was indeed only designed with off-the-shelf devices, which were physically disposed to avoid any interference with the wheelchair ergonomics. In order to increase the access to this technology, we kept the cost of the hardware framework at 2.000,00€, which is equivalent to the cost of ordinary powered wheelchairs. In addition, results of a public survey suggested that our design can mitigate the visual impact of the additional devices assembled in the wheelchair.

Despite the encouraging results presented in the Chapters 4 and 5, there is not sufficient evidence to completely verify the second hypothesis. The vision-based localization approach demonstrated to provide good localization estimations, but further improvements are still required to compute motion estimations purely based on computer vision. Indeed, the results and discussions presented in the previous Chapters suggest that a truly robust localization approach can only be achieved through the combination of multiple sensors and methodologies.

6.4 Recommendations For Future Work

Inspired by the contributions made in this thesis, a number of interesting open issues are worth investigating. In the following we suggest several recommendations for future work, filtering them by application domain:

Intelligent Wheelchair: The placement of an RGB-D camera have been contemplated in the IntellWheels hardware architecture and also during the development of the sensor bars. Despite the specification of how to position and fix such sensors in the wheelchair, the manufacture of such mounting was not included in the scope of this thesis and should be seen as future work. Our design places the sensor in the front part of the wheelchair, facing forward. In order to keep the access to the wheelchair seat clear, ant thus better preserve the wheelchair ergonomics, the sensor should be rotated 90 degrees from its standard horizontal operating position. It may be fixed on the wheelchair sensor bar through a special tip, which should replaces one of the current rounded tip that holds two ultrasound sensors. In addition the obstacle avoidance proposed is very sensor-depended, and does not overcome by itself the intrinsic sonar shortcomings. Therefore, further improvements should include some probabilistic analysis to increase robustness and reduce measure oscillations. Future work could also encompass new experiments in order to evaluate the proposed obstacle avoidance algorithm not only with able-bodied individuals, but also with patients with cerebral palsy, Parkinson and Alzheimer, as well as other possible end users.

Feature detection algorithm: The algorithm proposed for feature detection was designed to increase the feature repeatability without adding too much computational complexity to the original SURF implementation. A further development to reduce the algorithm computational time could include a code optimization and subdivision of tasks in multiple simultaneous threads. In the experimental section, a deeper analysis could include a comparison of the proposed method with other feature detection illumination invariant methods.

Localization Approach: One of the motivations of this research was to tackle the localization problems faced by intelligent wheelchairs. For this reason we proposed a vision-based odometry methodology, and tested it on a RGB-D dataset of a real-world scenario. In this sense, a following work would consist of integrating the proposed localization methodology in the IntellWheels perception agent, allowing the wheelchair to use the motion estimations for navigation purposes. Despite these encouraging results, our system has several shortcomings that deserve future effort. The mean runtime per frame, currently around 200ms, could certainly be speeded up by an efficient implementation based on modern GPU hardware. In addition, further research should be done to develop/implement a bundle adjustment algorithm to improve the estimated pose and assure a local consistency of the estimated path. Another promising research line would be to develop a RGB-D SLAM system based on the current RGB-D odometry approach. This way the wheelchair could identify revisited places and assure a global consistency to the estimated trajectory. Finally, another interesting avenue for research is the extraction of object representations from the rich information contained in dense 3D maps.

All these perspectives for future development are interesting in several areas of engineering, specially in the fields of the computer vision and robotics. The development of these future works could lead to more robust assistive robotic devices, and thus, improve the potential of people with motor disabilities.

Appendix A

Questionnaire: Assessment of IntellWheels Local Obstacle Avoidance

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

Motivation

Powered wheelchairs can provide a means of locomotion for people with mobility impairments. However, a broad variety of health conditions limit or even completely prevent several individuals from properly controlling ordinary powered wheelchairs. With the current reduction in the costs of computational systems, sensors and actuators, they start to be integrated in the wheelchairs. By endowing capacities of perception, decision and action, this new concept of wheelchairs can provide a greater assistance, independence and safety to the user.

Presentation

The goal of this study is to compare safety and the perception of safety of the wheelchair user in two distinct situations: in the regular driving of a powered wheelchair, and when aided by an autonomous obstacle avoidance methodology.

About the Questionnaire

This questionnaire is composed of qualitative questions were the respondents are invited to express their atitudes, opinions and satisfaction regarding the methodologies developed, thus there are no right or wrong answers. The evaluation of the answers will be governed according to the principles of ethical conduct,guaranteeing the right to privacy of the respondent as well as the anonymity and confidentiality of information provided

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

Questions

Instructions: Mark the most appropriate alternative for each of the questions below.

1. Please, inform your age in years: ___________

2. Please select your gender: **□ M** **□ F**

3. I frequently drive powered wheelchairs.

Strongly Disagree □ 1	Somewhat Disagree □ 2	Neither Agree nor Disagree □ 3	Somewhat Agree □ 4	Strongly Agree □ 5

4. In comparison to ordinary powered wheelchairs, the extra hardware assembled in the IntellWheels prototype caused a big visual impact.

Strongly Disagree □ 1	Somewhat Disagree □ 2	Neither Agree nor Disagree □ 3	Somewhat Agree □ 4	Strongly Agree □ 5

Comments:

Subject ID:	Date:

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO LIACC INESCTEC TECHNOLOGY & SCIENCE ASSOCIATE LABORATORY FCT Fundação para a Ciência e a Tecnologia

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

Manual Control Experiment

Instructions: Mark the most appropriate alternative for each of the questions below.

Questions	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat Agree	Strongly agree
I feel comfortable driving the wheelchair.	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
I feel that I am in control of the wheelchair.	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
The wheelchair behaves as expected.	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
It is ease to drive the wheelchair in cluttered environments.	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
Driving the wheelchair requires little attention.	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
The wheelchair presents the same behaviour in both real and simulated environments.	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5

Comments:

Subject ID:	Date:	Nº collisions:

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

Shared Control Experiment

Instructions: Mark the most appropriate alternative for each of the questions below.

Questions	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat Agree	Strongly agree
I feel comfortable driving the wheelchair.	□ 1	□ 2	□ 3	□ 4	□ 5
I feel that I am in control of the wheelchair.	□ 1	□ 2	□ 3	□ 4	□ 5
The wheelchair behaves as expected.	□ 1	□ 2	□ 3	□ 4	□ 5
It is ease to drive the wheelchair in cluttered environments.	□ 1	□ 2	□ 3	□ 4	□ 5
Driving the wheelchair requires little attention.	□ 1	□ 2	□ 3	□ 4	□ 5
The wheelchair presents the same behaviour in both real and simulated environments.	□ 1	□ 2	□ 3	□ 4	□ 5
I believe that the wheelchair helped me in the navigation process.	□ 1	□ 2	□ 3	□ 4	□ 5

Comments:

Subject ID: Date: Nº collisions:

Appendix B

Questionnaire: Assessment of IntellWheels Visual Appearance

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

Motivation

A broad variety of health conditions limit or even completely prevent several individuals with mobility impairments from properly controlling ordinary powered wheelchairs. With the current reduction in the costs of computational systems, sensors and actuators, they start to be integrated in the wheelchairs. By endowing capacities of perception, decision and action, this new concept of intelligent wheelchairs can provide a greater assistance, independence and safety to the user. Nevertheless, the addition of cameras, laser scanners, computer and displays can deeply modify the visual appearance, comfort and ergonomics of ordinary wheelchairs. Often, such situation creates physical and psychological barriers that tends to alienate potential intelligent wheelchair users.

Presentation

The goal of this study is to evaluate the visual impact of the IntellWheels prototype comparing its current appearance with the original powered wheelchair it was based on, and with the intelligent wheelchair prototypes of other research projects.

About the Questionnaire

This questionnaire is composed of qualitative questions were the respondents are invited to express their opinions regarding the visual appearance of intelligent wheelchair prototypes, thus there are no right or wrong answers. The evaluation of the answers will be governed according to the principles of ethical conduct,guaranteeing the right to privacy of the respondent as well as the anonymity and confidentiality of information provided.

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO LIACC INESCTEC TECHNOLOGY & SCIENCE ASSOCIATE LABORATORY FCT Fundação para a Ciência e a Tecnologia

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

1. Please, inform your age in years: ___________

2. Please select your gender: ❏ M ❏ F

3. Please, indicate your highest level of education:

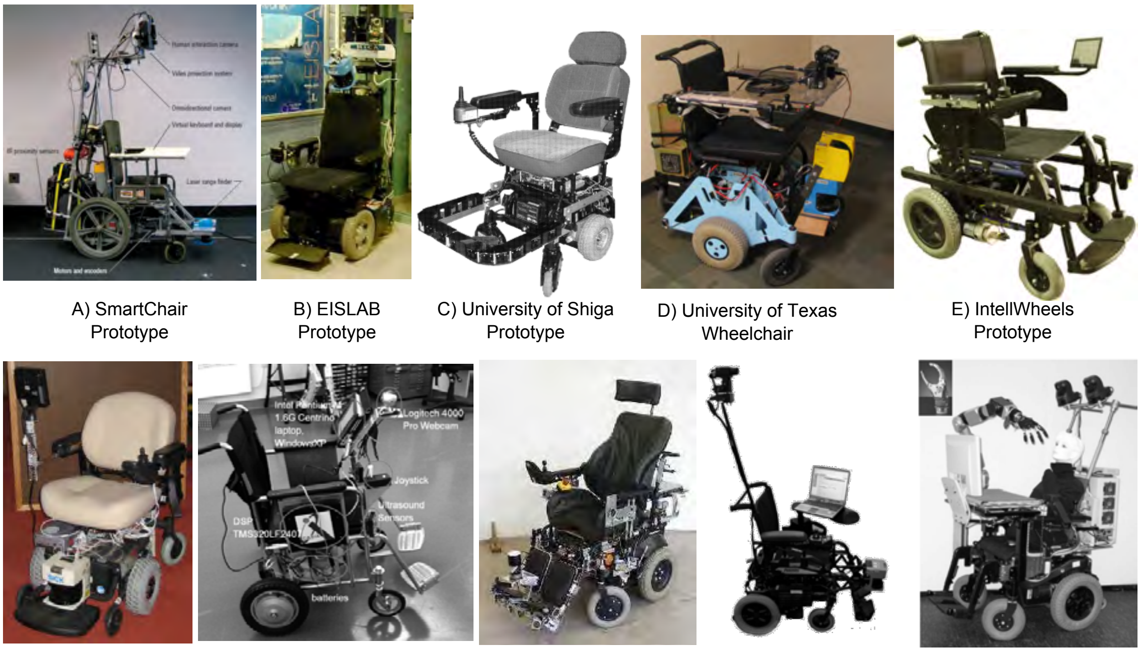
❏ I do not know
❏ No schooling
❏ 1 Cycle (1º - 4º years)
❏ 2 Cycle (5º - 6º years)
❏ 3 Cycle (7º - 9º years)
❏ High school
❏ Bachelors degree
❏ Masters degree
❏ Doctorates degree

4. Please indicate the number of years you are a wheelchair user (If applicable): ___________

Comments:

Subject ID:	Date:

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface



A) SmartChair Prototype

B) EISLAB Prototype

C) University of Shiga Prototype

D) University of Texas Wheelchair

E) IntellWheels Prototype

F) MIT Prototype

G) Robochair Prototype

H) Sharioto Prototype

I) SENA Prototype

J) FRIEND II Prototype

Figure B.1: Intelligent wheelchair prototypes

5. Based on the pictures of intelligent wheelchair prototypes (Fig. B.1), please indicate (for each prototype) your level of agreement with the following statement:

The addition of sensors and other hardware devices had visual/ergonomic impact on the wheelchair (e.g. changed the normal appearance/usage of the Wheelchair)

	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
A)SmartChair	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
B)EISLAB	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
C)University of Shiga	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
D)University of Texas	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
E)IntellWheels	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
F)MIT	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
G)Robochair	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
H)Sharioto	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
I)SENA	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5
J)FRIEND II	❑ 1	❑ 2	❑ 3	❑ 4	❑ 5

IntellWheels: Intelligent Wheelchair with Flexible Multimodal interface

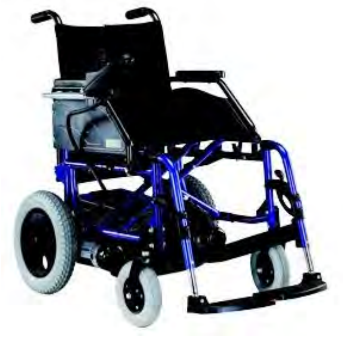


Figure B.2: Original powered wheelchair

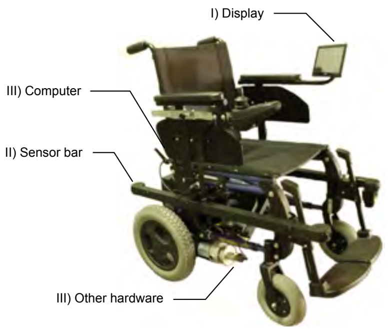


Figure B.3: Intellwheels prototype

6. For each statement below, please indicate your level of agreement:

In comparison with the original powered wheelchair (Fig. B.2), global visual changes of the IntellWheels prototype (Fig. B.3) are small.

Strongly Disagree ❑ 1	Somewhat Disagree ❑ 2	Neither Agree nor Disagree ❑ 3	Somewhat Agree ❑ 4	Strongly Agree ❑ 5

In comparison with the original powered wheelchair (Figure B.2), visual changes introduced by the display (Fig. B.3.I) are small.

Strongly Disagree ❑ 1	Somewhat Disagree ❑ 2	Neither Agree nor Disagree ❑ 3	Somewhat Agree ❑ 4	Strongly Agree ❑ 5

In comparison with the original powered wheelchair (Figure B.2), visual changes introduced by the sensor bars (Fig. B.3.II) are small.

Strongly Disagree ❑ 1	Somewhat Disagree ❑ 2	Neither Agree nor Disagree ❑ 3	Somewhat Agree ❑ 4	Strongly Agree ❑ 5

In comparison with the original powered wheelchair (Figure B.2), visual changes introduced by the PC and other hardware (Fig. B.3.III) are small.

Strongly Disagree ❑ 1	Somewhat Disagree ❑ 2	Neither Agree nor Disagree ❑ 3	Somewhat Agree ❑ 4	Strongly Agree ❑ 5

References

[1] I. Adelola, S. Cox, and A. Rahman, "Adaptable virtual reality interface for powered wheelchair training of disabled children," in *4th International Conference on Disability, Virtual Reality and Associated Technologies - ICDVRAT*, Veszprém, Hungary, 18-20 September 2002, pp. 173–180.

[2] M. A. Jones, I. R. McEwen, and L. Hansen, "Use of power mobility for a young child with spinal muscular atrophy," *Physical Therapy*, vol. 83, no. 3, pp. 253–262, 2003.

[3] L. Montesano, M. Diaz, S. Bhaskar, and J. Minguez, "Towards an intelligent wheelchair system for users with cerebral palsy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 193–202, 2010.

[4] R. C. Simpson, E. F. LoPresti, and R. A. Cooper, "How many people would benefit from a smart wheelchair?" *Journal of Rehabilitation Research and Development*, vol. 45, no. 1, pp. 53–71, 2008.

[5] R. Murphy, *Introduction to AI robotics*, ser. Intelligent robotics and autonomous agents. Cambridge, Mass.: MIT Press, 2000.

[6] J. Borenstein and L. Q. Feng, "Correction of systematic odometry errors in mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems: Human Robot Interaction and Cooperative Robots*, vol. 3, Pittsburgh, PA, USA, 5-9 August 1995, pp. 569–574.

[7] L. Matthies and S. A. Shafer, "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 239–248, June 1987.

[8] E. Royer, J. Bom, M. Dhome, B. Thuilot, M. Lhuillier, and F. Marmoiton, "Outdoor autonomous navigation using monocular vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, vol. 3, Edmonton, Canada, 2-6 August 2005, pp. 3395–3400.

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[10] B. Williams, G. Klein, and I. Reid, "Real-time slam relocalisation," in *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 14-21 October 2007, pp. 2244–2251.

[11] L. M. Paz, P. Pinies, J. D. Tards, and J. Neira, "Large-scale 6-dof slam with stereo-in-hand," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 946–957, October 2008.

[12] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IEEE/RSJ International Conference on Robots and Intelligent Systems - IROS*, Nice, France, 22-26 September 2008, pp. 3946–3952.

[13] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *IEEE International Conference on Computer Vision*, vol. 2, Nice, France, 13-16 October 2003, pp. 1403–1410.

[14] R. Sim, P. Elinas, and J. J. Little, "A study of the rao-blackwellised particle filter for efficient and accurate vision-based slam," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 303–318, September 2007.

[15] R. A. M. Braga, M. Petry, A. P. Moreira, and L. P. Reis, "Concept and design of the intellwheels development platform for intelligent wheelchairs," in *Informatics in Control, Automation and Robotics*, ser. Lecture Notes in Electrical Engineering, J. A. Cetto, J.-L. Ferrier, and J. Filipe, Eds. Heidelberg: Springer Berlin Heidelberg, 2009, vol. 37, pp. 191–203.

[16] R. A. Braga, M. Petry, L. P. Reis, and A. P. Moreira, "Intellwheels: Modular development platform for intelligent wheelchairs," *Journal of Rehabilitation Research and Development*, vol. 48, no. 9, pp. 1061–1076, December 2011.

[17] B. M. Faria, L. Ferreira, L. P. Reis, N. Lau, M. Petry, and J. Couto, "Manual control for driving an intelligent wheelchair: A comparative study of joystick mapping methods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - Progress, challenges and future perspectives in navigation and manipulation assistance for robotic wheelchairs workshop*, Vila Moura, Portugal, 7-12 October 2012, pp. 1–7.

[18] B. M. Faria, L. Ferreira, L. P. Reis, N. Lau, and M. Petry, "Intelligent wheelchair manual control methods: A usability study by cerebral palsy patients," in *Progress in Artificial Intelligence - EPIA 2013 / Lecture Notes in Computer Science*, ser. Lecture Notes in Computer Science, L. Correia, L. P. Reis, and J. Cascalho, Eds. Berlin: Springer Berlin Heidelberg, 2013, vol. 8154, pp. 271–282.

[19] R. A. M. Braga, M. Petry, A. P. Moreira, and L. P. Reis, "Intellwheels - a development platform for intelligent wheelchairs for disabled people," in *International Conference on Informatics in Control, Automation and Robotics*, J. A. Cetto and J. L. Ferrier, Eds. Funchal, Portugal: INSTICC-Inst Syst Technologies Information Control and Communication, 11-15 May 2008, pp. 115–121.

[20] M. Petry, A. P. Moreira, B. M. Faria, and L. P. Reis, "Intellwheels: Intelligent wheelchair with user-centered design," in *IEEE 15th International Conference on e-Health Networking, Applications and Services - Healthcom*, Lisbon, Portugal, 9-12 October [To be published] 2013, pp. 1–5.

[21] M. Petry, A. P. Moreira, L. P. Reis, and R. Rossetti, "Intelligent wheelchair simulation: Requirements and architectural issues," in *International Conference on Mobile Robots and Competitions*, Lisbon, Portugal, 6 April 2011, pp. 102–107.

[22] M. Petry, A. P. Moreira, R. A. M. Braga, and L. P. Reis, "Shared control for obstacle avoidance in intelligent wheelchairs," in *IEEE Conference on Robotics Automation and Mechatronics - RAM*, Singapore, 28-30 June 2010, pp. 182–187.

[23] M. Petry, A. P. Moreira, and L. P. Reis, “Increasing illumination invariance of surf feature detector through color constancy,” in *Progress in Artificial Intelligence - EPIA 2013 / Lecture Notes in Computer Science*, ser. Lecture Notes in Computer Science, L. Correia, L. P. Reis, and J. Cascalho, Eds. Berlin: Springer Berlin Heidelberg, 2013, vol. 8154, p. 259–270.

[24] ——, “Lsac surf: a surf feature detector with large photometric invariance,” [Submitted] 2013.

[25] M. R. Petry, A. P. Moreira, and L. P. Reis, “Rgb-d based motion estimation for mobile robots,” [Submitted] 2013.

[26] S. A. Shafer, “Using color to separate reflection components,” *Color Research & Application*, vol. 10, no. 4, pp. 210–218, 1985.

[27] H. D. Cheng, X. H. Jiang, Y. Sun, and J. L. Wang, “Color image segmentation: Advances and prospects,” *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, December 2001.

[28] F. Perez and C. Koch, “Toward color image segmentation in analog vlsi - algorithm and hardware,” *International Journal of Computer Vision*, vol. 12, no. 1, pp. 17–42, February 1994.

[29] A. Munsell, *A Color Notation*. Baltimore: Munsell Color Company, 1939.

[30] A. R. Smith, “Color gamut transform pairs,” in *5th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH*, vol. 12, Atlanta, Ga, USA, 23 - 25 August 1978, pp. 12–19.

[31] G. H. Joblove and D. Greenberg, “Color spaces for computer graphics,” in *5th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH*, vol. 12. Atlanta, Ga, USA: ACM, 23 - 25 August 1978, pp. 20–25.

[32] M. W. Schwarz, W. B. Cowan, and J. C. Beatty, “An experimental comparison of rgb, yiq, lab, hsv, and opponent color models,” *ACM Transactions on Graphics*, vol. 6, no. 2, pp. 123–158, 1987.

[33] K. McLaren, “Xiii—the development of the cie 1976 (l* a* b*) uniform colour space and colour-difference formula,” *Journal of the Society of Dyers and Colourists*, vol. 92, no. 9, pp. 338–341, September 1976.

[34] A. Hanbury, “Constructing cylindrical coordinate colour spaces,” *Pattern Recognition Letters*, vol. 29, no. 4, pp. 494–500, March 2008.

[35] S. D. Buluswar and B. A. Draper, “Color recognition in outdoor images,” in *International Conference on Computer Vision*, Bombay , India, 04 - 07 January 1998, pp. 171–177.

[36] J. v. Kries, “Influence of adaptation on the effects produced by luminous stimuli,” in *Sources of color science*, D. L. MacAdam, Ed. Cambridge, USA: MIT Press, 1970.

[37] D. A. Forsyth, “A novel algorithm for color constancy,” *International Journal of Computer Vision*, vol. 5, no. 1, pp. 5–36, August 1990.

[38] G. D. Finlayson, S. D. Hordley, and R. Xu, “Convex programming colour constancy with a diagonal-offset model,” in *IEEE International Conference on Image Processing - ICIP*, vol. 3, Genova, Italy, 11-14 September 2005, pp. III–948–51.

[39] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, September 2010.

[40] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*, vol. 1, Kauai, HI, USA, 08-14 December 2001, pp. 511–518.

[41] D. A. Forsyth and J. Ponce, *Computer vision a modern approach.* New Jersey: Prentice-Hall, 2003.

[42] J. J. Koenderink, “The structure of images,” *Biological Cybernetics*, vol. 50, no. 5, pp. 363–370, August 1984.

[43] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, “Uniqueness of the gaussian kernel for scale-space filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 1, pp. 26–33, January 1986.

[44] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, November 1998.

[45] M. A. Fischler and R. C. Bolles, “Random sample consensus : a paradigm for model-fitting with applications to image-analysis and automated cartography,” *Communications of ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[46] E. Vazquez, T. Gevers, M. Lucassen, J. van de Weijer, and R. Baldrich, “Saliency of color image derivatives: a comparison between computational models and human perception,” *Journal of the Optical Society of America A*, vol. 27, no. 3, pp. 613–621, February 2010.

[47] J. van de Weijer, T. Gevers, and A. D. Bagdanov, “Boosting color saliency in image feature detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 150–156, Jan 2006.

[48] A. E. Abdel-Hakim and A. A. Farag, “Csift: A sift descriptor with color invariant characteristics,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR*, vol. 2. New York, NY, USA: IEEE Computer Society, 17-22 June 2006, pp. 1978–1983.

[49] 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, July 2008.

[50] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, Manchester, UK, 1988, p. 147–151.

[51] K. Mikolajczyk and C. Schmid, “Indexing based on scale invariant interest points,” in *IEEE International Conference on Computer Vision*, vol. I, Vancouver, Canada, 07-14 Jul 2001, pp. 525–531.

[52] P. J. Burt and E. H. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, Apr 1983.

[53] D. G. Lowe, “Object recognition from local scale-invariant features,” in *International Conference on Computer Vision*, vol. 2. Kerkyra, Greece: IEEE Computer Society, 20-27 September 1999, pp. 1150–1157.

[54] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European Conference on Computer Vision*, vol. 3951, Graz, Austria, 7-13 May 2006, pp. 404–417.

[55] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[56] M. Brown and D. Lowe, “Invariant features from interest point groups,” pp. 656–665, September 2002.

[57] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts, “Color invariance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1338–1350, December 2001.

[58] G. J. Burghouts and J. M. Geusebroek, “Performance evaluation of local colour invariants,” *Computer Vision and Image Understanding*, vol. 113, pp. 48–62, January 2009.

[59] M. A. Ruzon and C. Tomasi, “Edge, junction, and corner detection using color distributions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1281–1295, November 2001.

[60] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, “A standard default color space for the internet-srgb,” Hewlett-Packard and Microsoft, Tech. Rep., 1996.

[61] J. A. Worthey, “Limitations of color constancy,” *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 2, no. 7, pp. 1014–1026, 1985.

[62] J. A. Worthey and M. H. Brill, “Heuristic analysis of von kries color constancy,” *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 2, no. 13, pp. P13–P14, 1985.

[63] G. D. Finlayson, M. S. Drew, and B. V. Funt, “Diagonal transforms suffice for color constancy,” in *Fourth International Conference on Computer Vision*, Berlin, Germany, 11-14 May 1993, pp. 164–171.

[64] ——, “Color constancy: enhancing von kries adaption via sensor transformations,” in *Human Vision, Visual Processing, and Digital Display IV*, J. P. Allebach and B. E. Rogowitz, Eds., vol. 1913, San Jose, CA, USA, 31 January 1993, pp. 473–484.

[65] ——, “Color constancy: generalized diagonal transforms suffice,” *Journal of the Optical Society of America A*, vol. 11, no. 11, pp. 3011–3019, January 1994.

[66] G. D. Finlayson, “Coefficient color constancy,” PhD Thesis, School of Computing, Simon Fraser University, April 1995.

[67] K. Barnard, L. Martin, A. Coath, and B. Funt, “A comparison of computational color constancy algorithms - part ii: Experiments with image data,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 985–996, Sep 2002.

[68] G. D. Finlayson, M. S. Drew, and B. V. Funt, “Spectral sharpening: sensor transformations for improved color constancy,” *Journal of the Optical Society of America A*, vol. 11, no. 5, pp. 1553–1563, May 1994.

[69] K. Barnard and B. Funt, “Experiments in sensor sharpening for color constancy,” in *IS&T/SID 6th Color Imaging Conference: Color Science, Systems and Applications*, vol. 6, Scottsdale, Arizona, USA, November 1998, pp. 43–46.

[70] M. S. Drew and G. D. Finlayson, "Spectral sharpening with positivity," *Journal of the Optical Society of America A*, vol. 17, no. 8, pp. 1361–1370, August 2000.

[71] R. Simpson, E. LoPresti, S. Hayashi, I. Nourbakhsh, and D. Miller, "The smart wheelchair component system," *Journal of Rehabilitation Research and Development*, vol. 41, no. 3B, pp. 429–442, 2004.

[72] P. Jia, H. Hu, T. Lu, and K. Yuan, "Head gesture recognition for hands-free control of an intelligent wheelchair," *Journal of Industrial Robot*, vol. IV, no. 1, pp. 60–68, 2007.

[73] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey," *Journal of Rehabilitation Research and Development*, vol. 37, no. 3, pp. 353–360, May/June 2000.

[74] R. L. Madarasz, L. C. Heiny, R. F. Cromp, and N. M. Mazur, "The design of an autonomous vehicle for the disabled," *IEEE Journal of Robotics and Automation*, vol. 2, no. 3, pp. 117–126, September 1986.

[75] P. Nisbet, J. Craig, P. Odor, and S. Aitken, "Smart wheelchairs for mobility training," *Technology and Disability*, vol. 5, no. 1, pp. 49–62, May 1996.

[76] P. D. Nisbet, "Who's intelligent? wheelchair, driver or both?" in *International Conference on Control Applications*, vol. 2, Glasgow, Scotlank, UK, 18-20 September 2002, pp. 760–765.

[77] S. Rehab, "Smile rehab ltd," [Online]. Available: http://www.smilerehab.com/, [Accessed on March 2013].

[78] H. Hoyer and R. Hoelper, "Open control architecture for an intelligent omnidirectional wheelchair," *Rehabilitation Technology : Strategies for the European Union*, vol. 9, pp. 93–97, 1993.

[79] U. Borgolte, H. Hoyer, C. Buhler, H. Heck, and R. Hoelper, "Architectural concepts of a semi-autonomous wheelchair," *Journal of Intelligent & Robotic Systems*, vol. 22, no. 3-4, pp. 233–253, 1998.

[80] R. Simpson, S. P. Levine, D. A. Bell, L. Jaros, Y. Koren, and J. Borenstein, *NavChair: An Assistive Wheelchair Navigation System with Automatic Adaptation*. Springer-Verlag, 1998, vol. 1458.

[81] S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, and J. Borenstein, "The navchair assistive wheelchair navigation system," *IEEE Transactions on Rehabilitation Engineering*, vol. 7, no. 4, pp. 443–451, December 1999.

[82] D. P. Miller and M. Slack, "Design and testing of a low-cost robotic wheelchair prototype," *Autonomous Robots*, vol. II, no. 1, pp. 77–88, March 1995.

[83] E. Prassler, J. Scholz, and P. Fiorini, "A robotic wheelchair for crowded public environments," *IEEE Robotics & Automation Magazine*, vol. 8, no. 1, pp. 38–45, March 2001.

[84] P. Wellman, V. Krovi, and V. Kumar, "An adaptive mobility system for the disabled," in *IEEE International Conference on Robotics and Automation*, vol. 3, San Diego, CA, USA, 8-13 May 1994, pp. 2006–2011.

[85] B. Borgerding, O. Ivlev, C. Martens, N. Ruchel, and A. Gräser, "Friend-functional robot arm with user friendly interface for disabled people," in *5th European Conference for the Advancement of Assistive Technology*, 1999.

[86] C. Martens, N. Ruchel, O. Lang, O. Ivlev, and A. Graser, "A friend for assisting handicapped people," *Robotics & Automation Magazine, IEEE*, vol. 8, no. 1, pp. 57–65, 2001.

[87] I. Volosyak, O. Ivlev, and A. Graser, "Rehabilitation robot friend ii-the general concept and current implementation," in *9th International Conference on Rehabilitation Robotics - ICORR*. Chicago, USA: IEEE, 28 June-1 July 2005, pp. 540–544.

[88] O. Prenzel, J. Feuser, and A. Graser, "Rehabilitation robot in intelligent home environment - software architecture and implementation of a distributed system," in *9th International Conference on Rehabilitation Robotics - ICORR*, Chicago, USA, 28 June-1 July 2005 2005, pp. 530–535.

[89] S. P. Parikh, V. Grassi Jr, V. Kumar, and J. Okamoto Jr, "Incorporating user inputs in motion planning for a smart wheelchair," in *IEEE International Conference on Robotics and Automation - ICRA*, vol. 2. New Orleans, USA: IEEE, 26 April - 1 May 2004, pp. 2043–2048.

[90] S. P. Parikh, V. Grassi, V. Kumar, and J. Okamoto, "Usability study of a control framework for an intelligent wheelchair," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18-22 April 2005, pp. 4745–4750.

[91] S. P. Parikh, V. Grassi Jr, V. Kumar, and J. Okamoto Jr, "Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform," *Intelligent Systems, IEEE*, vol. 22, no. 2, pp. 33–41, April 2007.

[92] FIPA, "Foundation for intelligent physical agents," [Online]. Available: http://www.fipa.org, [Accessed on November 2010].

[93] J. Gonzalez, A. J. Muaeoz, C. Galindo, J. A. Fernandez-Madrigal, and J. L. Blanco, "A description of the sena robotic wheelchair," in *IEEE Mediterranean Electrotechnical Conference - MELECON*, Malaga, Spain, 16-19 May 2006, pp. 437–440.

[94] T. Hamagami and H. Hirata, "Development of intelligent wheelchair acquiring autonomous, cooperative, and collaborative behavior," in *IEEE International Conference on Systems, Man & Cybernetics*, vol. 1-7, Hague, Netherlands, 10 -13 October 2004, pp. 3525–3530.

[95] S. Rönnbäck, J. Piekkari, K. Hyyppä, L. Haakapää, V. Kammunen, and S. Koskinen, "Mica-mobile internet connected assistant," in *International Conference in Lifestyle, Health and Technolgy*, Luleå, Sweden, 1-3 June 2005, pp. 1–5.

[96] S. Rönnbäck, J. Piekkari, K. Hyyppä, T. Berglund, and S. Koskinen, "A semi-autonomous wheelchair towards user-centered design," in *Computers Helping People with Special Needs*, ser. Lecture Notes in Computer Science, K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, Eds. Berlin: Springer Berlin Heidelberg, 2006, vol. 4061, pp. 701–708.

[97] S. Hemachandra, T. Kollar, N. Roy, and S. Teller, "Following and interpreting narrated guided tours," in *IEEE International Conference on Robotics and Automation - ICRA*. Changhai, China: IEEE, 9-11 May 2011, pp. 2574–2579.

[98] D. Vanhooydonck, E. Demeester, M. Nuttin, and H. Van Brussel, "Shared control for intelligent wheelchairs: an implicit estimation of the user intention," in *International Workshop on Advances in Service Robotics - ASER'03*. Bardolino, Italy: Citeseer, 13-15 March 2003, pp. 176–182.

[99] T. Yasuda, K. Nakamura, A. Kawahara, and K. Tanaka, "Neural network with variable type connection weights for autonomous obstacle avoidance on a prototype of six-wheel type intelligent wheelchair," *International Journal of Innovative Computing, Information and Control*, vol. 2, no. 5, pp. 1165–1177, October 2006.

[100] N. L. Katevas, N. M. Sgouros, S. G. Tzafestas, G. Papakonstantinou, P. Beattie, J. M. Bishop, P. Tsanakas, and D. Koutsouris, "The autonomous mobile robot senario: A sensor-aided intelligent navigation system for powered wheelchairs," *IEEE Robotics & Automation Magazine*, vol. 4, no. 4, pp. 60–70, December 1997.

[101] G. Bourhis, O. Horn, O. Habert, and A. Pruski, "An autonomous vehicle for people with motor disabilities," *Robotics & Automation Magazine, IEEE*, vol. 8, no. 1, pp. 20–28, 2001.

[102] A. Lankenau and T. Rofer, "A versatile and safe mobility assistant," *IEEE Robotics & Automation Magazine*, vol. 8, no. 1, pp. 29–37, 2001.

[103] S. Gulati and B. Kuipers, "High performance control for graceful motion of an intelligent wheelchair," in *IEEE International Conference on Robotics and Automation - ICRA'08*. Pasadena, USA: IEEE, 19-23 May 2008, pp. 3932–3938.

[104] M. Mazo, "An integral system for assisted mobility [automated wheelchair]," *Robotics & Automation Magazine, IEEE*, vol. 8, no. 1, pp. 46–56, 2001.

[105] P. C. Ng and L. C. de Silva, "Head gestures recognition," in *Proceedings International Conference on Image Processing*, vol. III, 2001, pp. 266–269.

[106] Y. Adachi, Y. Kuno, N. Shimada, and Y. Shirai, "Intelligent wheelchair using visual information on human faces," in *IEEE/RSJ Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 1-3, Victoria, BC, Canada, 13 -17 October 1998, pp. 354–359.

[107] H. Lakany, "Steering a wheelchair by thought," in *IEEE International Workshop on Intelligent Environments*, Colchester, UK, 29 June 2005, pp. 199 – 202.

[108] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. L. Teo, Q. Zeng, C. Laugier, and M. H. Ang Jr., "Controlling a wheelchair indoors using thought," *IEEE Intelligent Systems and Their Applications*, vol. 22, no. 2, pp. 18–24, March-April 2007.

[109] RADHAR, "Radhar - robotic adaptation to humans adapting to robots," [Online]. Available: https://www.radhar.eu/, [Accessed on March 2013].

[110] E. Demeester, E. EB Vander Poorten, A. Hüntemann, and J. De Schutter, "Wheelchair navigation assistance in the fp7 project radhar: Objectives and current state," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS, workshop on Navigation and Manipulation Assistance for Robotic Wheelchairs*, Vilamoura, Portugal, 12 October 2012, pp. 1–8.

[111] LURCH, "Lurch – the autonomous wheelchair," [Online]. Available: http://airlab.elet.polimi.it/index.php/LURCH_-_The_autonomous_wheelchair, [Accessed on March 2013].

[112] H. Soh and Y. Demiris, "Towards early mobility independence: An intelligent paediatric wheelchair with case studies," in *IEEE/RSJ International Conference on Intelligent Robots and Systems. Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs*, Vila Moura, Portugal, 2012, pp. 1–7.

[113] I. Iturrate, J. M. Antelis, A. Kubler, and J. Minguez, "A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 614–627, 2009.

[114] I. Iturrate, J. Antelis, and J. Minguez, "Synchronous eeg brain-actuated wheelchair with automated navigation," in *IEEE International Conference on Robotics and Automation - ICRA'09*. Kobe, Japan: IEEE, 12-17 May 2009, pp. 2318–2325.

[115] F. Ribeiro, "Enigma: Cadeira de rodas omnidireccional," *Revista Robótica*, no. 6, pp. 50–51, 2007.

[116] L. Figueiredo, "Projecto magickey," [Online]. Available: http://www.magickey.ipg.pt, [Accessed on March 2013].

[117] G. Pires and U. Nunes, "A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller," *Journal of Intelligent and Robotic Systems*, vol. 34, no. 3, pp. 301–314, July 2002.

[118] G. Pires, U. Nunes, and A. de Almeida, "Robchair a semi autonomous wheelchair for disabled people," in *3rd IFAC Symposium on Intelligent Autonomous Vehicles - IAV'98*, Madrid, Spain, 25-27 March 1998, pp. 648–652.

[119] R. Solea and U. Nunes, "Robotic wheelchair control considering user comfort: modeling and experimental evaluation," in *International Conference on Informatics in Control, Automation and Robotics - ICINCO*, Funchal, Portugal, 11-15 May 2008, pp. 1–8.

[120] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, March 1986.

[121] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of Rehabilitation Research and Development*, vol. 42, no. 4, pp. 423–435, 2005.

[122] H. Wakaumi, K. Nakamura, and T. Matsumura, "Development of an automated wheelchair guided by a magnetic ferrite marker lane," *Journal of Rehabilitation Research and Development*, vol. 29, no. 1, pp. 27–34, January 1992.

[123] R. Ouiguini and B. Belloulou, "Navigation of an autonomous wheelchair in a structured environment," in *IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 2, Taipei, Taiwan, 5-10 August 1996, pp. 749–754.

[124] R. Gelin, J. M. Detriche, J. P. Lambert, and P. Malblanc, "The sprint of coach," in *International Conference on Systems, Man and Cybernetics. Systems Engineering in the Service of Humans.*, vol. 3, 17-20 October 1993, pp. 547–552.

[125] J. D. Yoder, E. T. Baumgartner, and S. B. Skaar, "Initial results in the development of a guidance system for a powered wheelchair," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 3, pp. 143–151, September 1996.

[126] C. Hon Nin, X. Yangsheng, and S. K. Tso, "Learning human navigational skill for smart wheelchair," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Lausanne, Switzerland, 30 September - 4 October 2002, pp. 996–1001.

[127] E. S. Boy, C. L. Teo, and E. Burdet, "Collaborative wheelchair assistant," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Lausanne, Switzerland, 30 September - 4 October 2002, pp. 1511–1516 vol.2.

[128] R. C. Simpson, D. Poirot, and F. Baxter, "The hephaestus smart wheelchair system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 118–122, June 2002.

[129] S. Rao and R. Kuc, "Inch: an intelligent wheelchair prototype," in *15th Annual Northeast Bioengineering Conference*, Boston, USA, 27-28 March 1989, pp. 35–36.

[130] K. Schilling, H. Roth, R. Lieb, and H. Stützle, "Sensors to improve the safety for wheelchair users," *Improving the quality of life for the European citizen. Technology for inclusive design and equality, Assistive Technology Research Series*, vol. 4, pp. 331–335, 1998.

[131] Y. Murakami, Y. Kuno, N. Shimada, and Y. Shirai, "Collision avoidance by observing pedestrians' faces for intelligent wheelchairs," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4, Maui, USA, 29 October - 3 November 2001, pp. 2018–2023.

[132] H. Seki, S. Kobayashi, Y. Kamiya, M. Hikizu, and H. Nomura, "Autonomous/semi-autonomous navigation system of a wheelchair by active ultrasonic beacons," in *IEEE International Conference on Robotics and Automation - ICRA*, vol. 2, San Francisco, USA, 24-28 April 2000, pp. 1366–1371.

[133] R. C. Luo, H. Chi-Yang, C. Tse Min, and L. Meng-Hsien, "Force reflective feedback control for intelligent wheelchairs," in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, Kyongju, South Korea, 17-19 October 1999, pp. 918–923.

[134] J. Connell and P. Viola, "Cooperative control of a semi-autonomous mobile robot," in *IEEE International Conference on Robotics and Automation*, vol. 2, Cincinnati, USA, 13-18 May 1990, pp. 1118–1121.

[135] M. Inhyuk, J. Sanghyun, and K. Youngkwang, "Safe and reliable intelligent wheelchair robot with human robot interaction," in *IEEE International Conference on Robotics and Automation - ICRA'02*, vol. 4, Washington, USA, 11-15 May 2002, pp. 3595–3600.

[136] L. Xueen, Z. Xiaojian, and T. Tieniu, "A behavior-based architecture for the control of an intelligent powered wheelchair," in *9th IEEE International Workshop on Robot and Human Interactive Communication - RO-MAN'00*, Osaka, Japan, 27–29 September 2000, pp. 80–83.

[137] N. Sgouros, "Qualitative navigation for autonomous wheelchair robots in indoor environments," *Autonomous Robots*, vol. 12, no. 3, pp. 257–266, May 2002.

[138] J. D. Crisman, M. E. Cleary, and J. C. Rojas, "The deictically controlled wheelchair," *Image and Vision Computing*, vol. 16, no. 4, pp. 235–249, April 1998.

[139] Y. Kuno, N. Shimada, and Y. Shirai, “Look where you’re going [robotic wheelchair],” *IEEE Robotics & Automation Magazine*, vol. 10, no. 1, pp. 26–34, March 2003.

[140] J.-A. Fernández-Madrigal, C. Galindo, and J. González, “Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment,” *Integrated Computer-Aided Engineering*, vol. 11, no. 4, pp. 309–322, December 2004.

[141] A. Civit-Balcells, F. Diaz Del Rio, G. Jimenez, J. L. Sevillano, C. Amaya, and S. Vicente, “Sirius: improving the maneuverability of powered wheelchairs,” in *International Conference on Control Applications*, vol. 2, Glasgow, UK, 18-20 September 2002, pp. 790–795.

[142] D. L. Jaffe, H. L. Harris, and S. K. Leung, “Ultrasonic head controlled wheelchair/interface: A case study in development and technology transfer,” in *13th Annual International Conference on Assistive Technology for People with Disabilities - RESNA*, Washington, DC, USA, 15–20 June 1990, pp. 23–24.

[143] R. Frisch, S. Guo, R. Cooper, E. LoPresti, R. Simpson, S. Hayashi, and W. Ammer, “Hardware design of the smart power assistance module for manual wheelchairs,” in *27th International Annual Conference on Assistive Technology for People with Disabilities - RESNA*, Orlando, USA, 20–22 June 2004, pp. 20–22.

[144] D. Ding, E. Lopresti, R. Simpson, and R. Cooper, “Interpreting joystick signals for wheelchair navigation,” in *26th International Annual Conference on Assistive Technology for People with Disabilities - RESNA*, Atlanta, USA, 19–23 June 2003.

[145] T. Gomi and A. Griffith, “Developing intelligent wheelchairs for the handicapped,” in *Assistive Technology and Artificial Intelligence*, ser. Lecture Notes in Computer Science, V. Mittal, H. Yanco, J. Aronis, and R. Simpson, Eds. Springer Berlin Heidelberg, 1998, vol. 1458, pp. 150–178.

[146] D. Cagigas and J. Abascal, “Hierarchical path search with partial materialization of costs for a smart wheelchair,” *Journal of Intelligent and Robotic Systems*, vol. 39, no. 4, pp. 409–431, April 2004.

[147] W. Gribble, R. Browning, M. Hewett, E. Remolina, and B. Kuipers, “Integrating vision and spatial reasoning for assistive navigation,” in *Assistive Technology and Artificial Intelligence*, ser. Lecture Notes in Computer Science, V. Mittal, H. Yanco, J. Aronis, and R. Simpson, Eds. Springer Berlin Heidelberg, 1998, vol. 1458, pp. 179–193.

[148] H. Kitagawa, T. Kobayashi, T. Beppu, and K. Terashima, “Semi-autonomous obstacle avoidance of omnidirectional wheelchair by joystick impedance control,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS’01*, vol. 4, Maui, HI, USA, 29 October - 03 Novemeber 2001, pp. 2148–2153.

[149] S. Fioretti, T. Leo, and S. Longhi, “A navigation system for increasing the autonomy and the security of powered wheelchairs,” *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 490–498, December 2000.

[150] D. A. Sanders, I. J. Stott, and M. J. Goodwin, “Assisting a disabled person in navigating an electric vehicle through a doorway,” in *IEE Colloquium on New Developments in Electric Vehicles for Disabled Persons*, London, UK, 17 March 1995, pp. 5/1–5/6.

[151] G. Bugmann, K. Koay, N. Barlow, M. Phillips, and D. Rodney, "Stable encoding of robot trajectories using normalised radial basis functions: Application to an autonomous wheelchair," in *International symposium on robotics*, Birmingham, UK, 27–30 April 1998, pp. 1–6.

[152] S. Chauhan, P. Sharma, H. R. Singh, A. Mobin, and S. S. Agrawal, "Design and development of voice-cum-auto steered robotic wheelchair incorporating reactive fuzzy scheme for anti-collision and auto routing," in *IEEE Region 10 Conference - TENCON'00*, vol. 1, Kuala Lumpur, Malaysia, 24-27 September 2000, pp. 192–195.

[153] P. Mallet and G. Schoner, "Wad project where attractor dynamics aids wheelchair navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS'02*, vol. 1, Lausanne, Switzerland, 30 September - 5 October 2002, pp. 690–695 vol.1.

[154] Y. Adachi, K. Goto, A. Khiat, Y. Matsumoto, and T. Ogasawara, "Estimation of user's attention based on gaze and environment measurements for robotic wheelchair," in *12th IEEE International Workshop on Robot and Human Interactive Communication - ROMAN'03*, Milbrae, CA, USA, 31 October - 2 November 2003, pp. 97–102.

[155] H. Yanco, "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface," in *Assistive Technology and Artificial Intelligence*, ser. Lecture Notes in Computer Science, V. Mittal, H. Yanco, J. Aronis, and R. Simpson, Eds. Springer Berlin Heidelberg, 1998, vol. 1458, pp. 256–268.

[156] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, Jun 1991 1991.

[157] ——, "Obstacle avoidance with ultrasonic sensors," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 213–218, Apr 1988.

[158] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 249–265, June 1987.

[159] T. Z. Wang and J. Yang, "Certainty grids method in robot perception and navigation," in *IEEE International Symposium on Intelligent Control*, Monterey, CA, 27-29 Aug, 1995 1995, pp. 539–544.

[160] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, July/August 1988 1988.

[161] J. Borenstein and Y. Koren, "Histogramic in-motion mapping for mobile robot obstacle avoidance," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 535–539, 1991.

[162] J. Andrews and N. Hogan, "Impedance control as a framework for implementing obstacle avoidance in a manipulator," p. 243–251, November 1983.

[163] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, March 1986.

[164] B. H. Krogh, "A generalized potential field approach to obstacle avoidance control," p. 1, 1984.

[165] H. Seki, S. Shibayama, Y. Kamiya, and M. Hikizu, "Practical obstacle avoidance using potential field for a nonholonmic mobile robot with rectangular body," in *IEEE International Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany, 15-18 Sept. 2008 2008, pp. 326–332.

[166] M. Khatib and R. Chatila, "An extended potential field approach for mobile robot sensor-based motions," in *International Conference on Intelligent Autonomous Systems*, Karlsruhe, Germany, March 1995 1995, pp. 490–496.

[167] E. Bicho, P. Mallet, and G. Schoner, "Using attractor dynamics to control autonomous vehicle motion," in *Annual Conference of the IEEE Industrial Electronics Society - IECON*, vol. 2, Aachen, Germany, 31 Aug-4 Sep 1998 1998, pp. 1176–1181.

[168] Y. Koren and J. Borenstein, "Potential-field methods and their inherent limitations for mobile robot navigation," in *IEEE International Conference on Robotics and Automation - ICRA*, Sacramento, CA, USA, 9-11 Apr 1991 1991, pp. 1398–1404.

[169] R. A. M. Braga, "Plataforma de desenvolvimento de cadeiras de rodas inteligentes," PhD Thesis, Department of Informatics Engineering, Faculty of Engineering, University of Porto, 2010.

[170] L. P. Reis, R. A. M. Braga, M. Sousa, and A. P. Moreira, "Intellwheels mmi: A flexible interface for an intelligent wheelchair," in *Robocup 2009: Robot Soccer World Cup XIII*, ser. Lecture Notes in Artificial Intelligence, T. S. J.Baltes, M.Lagoudakis, Ed. Heidelberg: Springer, 2009, vol. 5949, pp. 296–307.

[171] P. M. Faria, R. A. M. Braga, E. Valgode, and L. P. Reis, "Interface framework to drive an intelligent wheelchair using facial expressions," in *IEEE International Symposium on Industrial Electronics - ISIE*, Vigo, Spain, June 4-7 2007, pp. 1791–1796.

[172] P. M. Faria, R. A. M. Braga, E. Valgôde, and L. P. Reis, "Platform to drive an intelligent wheelchair using facial expressions," in *International Conference on Enterprise Information Systems - Human-Computer Interaction*, Funchal, Madeira, Portugal, 2007, pp. 164–169.

[173] B. M. Faria, L. P. Reis, and N. Lau, "Cerebral palsy eeg signals classification: Facial expressions and thoughts for driving an intelligent wheelchair," in *International Conference on Data Mining - Biological Data Mining and its Applications in Healthcare Workshop*, Brussels, Belgium, 10-13 December 2012, pp. 33–40.

[174] B. M. Faria, S. Vasconcelos, L. P. Reis, and N. Lau, "Evaluation of distinct input methods of an intelligent wheelchair in simulated and real environments: A performance and usability study," *Assistive Technology: The Official Journal of RESNA*, vol. 25, no. 2, pp. 88–98, September 2012.

[175] M. Petry, "Desenvolvimento de um protótipo e de metodologias de controlo de uma cadeira de rodas inteligente," Master Thesis, Department of Electrical and Computers Engineering, Faculty of Engineering, University of Porto, 26 February 2008.

[176] R. A. M. Braga, M. Petry, E. Oliveira, and L. P. Reis, "Multi-level control of an intelligent wheelchair in a hospital environment using a cyber-mouse simulation system," in *5th International Conference on Informatics in Control, Automation and Robotics - ICINCO*,

J. Filipe, J. A. Cetto, and J. L. Ferrier, Eds., vol. II. Funchal, Portugal: Insticc-Inst Syst Technologies Information Control and Communication, 11-15 May 2008, pp. 179–182.

[177] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, May 2001.

[178] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics.* Cambridge, MA, USA: MIT Press, 2005.

[179] R. A. M. Braga, M. Petry, L. P. Reis, and A. P. Moreira, "Plataform for intelligent wheelchairs using multi-level control and probabilistic motion model," pp. 833–838, July 21–23 2008.

[180] M. Fox and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, no. 1, pp. 61–124, December 2003.

[181] K. Fowler, "Missio-critical and safety-critical developent," *IEEE Instrumentation & Measurement Magazine*, vol. 7, no. 4, pp. 52– 59, December 2004.

[182] CENELEC, "En 50159-2 railway applications - communication, signalling and processing systems," 2001.

[183] F. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*, 1st ed. Wiley, 2007.

[184] S. Oviatt, "Multimodal interfaces," in *The human-computer interaction handbook*, A. J. Julie and S. Andrew, Eds. L. Erlbaum Associates Inc., 2003, pp. 286–304.

[185] M. Sousa, R. A. M. Braga, and L. P. Reis, "Multimodal interface for an intelligent wheelchair," in *International Workshop on Intelligent Robotics*, L.P.Reis, N. L.Correia, and R.Bianchi, Eds., Lisbon, 14 October 2008 2008, pp. 107–118.

[186] B. M. Faria, S. Vasconcelos, L. P. Reis, and N. Lau, "A methodology for creating intelligent wheelchair users' profiles," in *International Conference on Agents and Artificial Intelligence - ICAART.* Vila Moura, Portugal: ICAART, 6-8 February 2012, pp. 171–179.

[187] M. Friedmann, K. Petersen, and O. von Stryk, "Simulation of multi-robot teams with flexible level of detail," in *Simulation, Modeling, and Programming for Autonomous Robots*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5325/2008, pp. 29–40.

[188] C. Pepper, S. Balakirsky, and C. Scrapper, "Robot simulation physics validation," pp. 97–104, 2007.

[189] N. Lau, A. Pereira, A. Melo, A. Neves, and J. Figueiredo, "Ciber-rato: Um ambiente de simulaçao de robots móveis e autónomos," *Revista do DETUA*, vol. 3, no. 7, pp. 647–650, September 2002.

[190] N. Lau, A. Pereira, A. Melo, J. Neves, and J. Figueiredo, "Ciber-rato: Uma competição robótica num ambiente virtual," *Revista do DETUA*, vol. 3, no. 7, pp. 647–650, September 2002.

[191] P. Malheiro, R. Braga, and L. Reis, "Intellwheels simulator: A simulation environment for intelligent wheelchairs," in *3rd International Workshop on Intelligent Robotics - IROBOT 2008*, Lisbon, Portugal, 14-17 October 2008, pp. 95–106.

[192] R. A. Braga, P. Malheiro, and L. P. Reis, "Development of a realistic simulator for robotic intelligent wheelchairs in a hospital environment," in *RoboCup Symposium 2009*, ser. LNAI, T. S. J.Baltes, M.Lagoudakis, Ed. Heidelberg: Springer, 2009, vol. 5949, pp. 23–34.

[193] J. Craighead, R. Murphy, J. Burke, and B. Goldiez, "A survey of commercial & open source unmanned vehicle simulators," in *IEEE International Conference on Robotics and Automation*, Rome, Italy, 10-14 April 2007, pp. 852–857.

[194] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Usarsim: a robot simulator for research and education," in *IEEE International Conference on Robotics and Automation - ICRA*, Rome, Italy, 10-14 April 2007, pp. 1400–1405.

[195] S. Balaguer, S. Balakirsky, S. Carpin, M. Lewis, and C. Scrapper, "Usarsim: A validated simulator for research in robotics and automation," in *Workshop on Robot simulators: available software, scientific applications and future trends - IEEE/RSJ IROS*, 2008, p. 6.

[196] Microsoft, "Microsoft robotics developer studio 2008," [Online]. Available: http://www.microsoft.com/robotics/, [Accessed on November 2010].

[197] O. Michel, "Webotstm: Professional mobile robot simulation," *International Journal of Advanced Robotics Systems*, vol. 1, pp. 39–42, 2004.

[198] P. Costa, J. Gonçalves, J. Lima, and P. Malheiros, "Simtwo realistic simulator: A tool for the development and validation of robot software," *Theory and Applications of Mathematics & Computer Science*, vol. 1, no. 1, pp. 17–33, April 2011.

[199] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: A versatile 3d robot simulator," in *Simulation, Modeling, and Programming for Autonomous Robots*, ser. Lecture Notes in Computer Science, N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, and O. Stryk, Eds. Springer Berlin Heidelberg, 2010, vol. 6472, pp. 51–62.

[200] Gazebo, "Gazebo user manual," [Online]. Available: http://gazebosim.org/, [Accessed on May 2013].

[201] U. Engine, "Game engine technology by unreal," [Online]. Available: http://www.unrealengine.com/, [Accessed on March 2013].

[202] M. Lewis, J. Wang, and S. Hughes, "Usarsim: Simulation for the study of human-robot interaction," *Journal of Cognitive Engineering and Decision Making*, vol. 1, no. 1, pp. 98–120, 2007.

[203] U. Engine, "Unreal game editor," [Online]. Available: http://www.unrealengine.com/features/editor/, [Accessed on March 2013].

[204] K. L. Murdock, *3ds Max 2011 Bible*. Indianapolis, USA: Wiley, 2010.

[205] Autodesk, "3ds max - 3d modeling, animation and rendering software," [Online]. Available: http://usa.autodesk.com/3ds-max/, 2013.

[206] G. Bourhis and Y. Agostini, "The vahm robotized wheelchair: System architecture and human-machine interaction," *Journal of Intelligent & Robotic Systems*, vol. 22, no. 1, pp. 39–50, 1998.

[207] D. A. Bell, J. Borenstein, S. P. Levine, Y. Koren, and L. Jaros, "An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance," in *IEEE International Conference on Robotics and Automation*, vol. 3, San Diego, CA, USA, 08 - 13 May 1994, pp. 2018–2022.

[208] G. Corder and D. Foreman, *Nonparametric statistics for non-statisticians: a step-by-step approach*. Wiley, 2009.

[209] M. Ma, M. McNeill, S. McDonough, J. Crosbie, and L. Oliver, "Physics fidelity of virtual reality in motor rehabilitation," pp. 35–41, April 2006.

[210] T. Lemaire, C. Berger, I. K. Jung, and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, September 2007.

[211] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2003.

[212] J. Philip, "A non-iterative algorithm for determining all essential matrices corresponding to five point pairs," *The Photogrammetric Record*, vol. 15, no. 88, pp. 589–599, October 1996.

[213] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.

[214] H. Stewénius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, June 2006.

[215] D. Ortín and J. M. M. Montiel, "Indoor robot motion based on monocular images," *Robotica*, vol. 19, no. 03, pp. 331–342, May 2001.

[216] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *IEEE International Conference on Robotics and Automation - ICRA*, Kobe, Japan, 12-17 May 2009, pp. 488–494.

[217] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-point ransac for ekf-based structure from motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS 2009*, St. Louis, MO, USA, 10-15 October 2009, pp. 3498–3504.

[218] C. Ancuti and P. Bekaert, "Sift-cch: Increasing the sift distinctness by color co-occurrence histograms," in *International Symposium on Image and Signal Processing and Analysis*, Istanbul, Turkey, 27-29 September 2007, pp. 130–135.

[219] P. Fan, A. D. Men, M. Y. Chen, and B. Yang, "Color-surf: A surf descriptor with local kernel color histograms," in *IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, China, 6-8 November 2009, pp. 726–730.

[220] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via plsa," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin / Heidelberg, 2006, vol. 3954, pp. 517–530.

[221] G. D. Finlayson, S. D. Hordley, L. Cheng, and M. S. Drew, “On the removal of shadows from images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, January 2006.

[222] A. Gijsenij, L. Rui, and T. Gevers, “Color constancy for multiple light sources,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 697–707, February 2012.

[223] J. van de Weijer, T. Gevers, and A. Gijsenij, “Edge-based color constancy,” *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, September 2007.

[224] S. Zeki, *A Vision of the Brain*. Oxford: Blackwell Scientific Publications, 1993.

[225] B. Horn, *Robot vision*, mit press ed., ser. MIT electrical engineering and computer science series. Cambridge, Massachusetts: MIT Press, 1986.

[226] M. Ebner, “How does the brain arrive at a color constant descriptor?” in *Advances in Brain, Vision, and Artificial Intelligence*, ser. Lecture Notes in Computer Science, F. Mele, G. Ramella, S. Santillo, and F. Ventriglia, Eds. Springer Berlin / Heidelberg, 2007, vol. 4729, pp. 84–93.

[227] ——, “Why color constancy improves for moving objects,” in *International Conference on Bio-inspired Systems and Signal Processing - BIOSIGNALS 2012*, S. V. Huffel, C. M. B. A. Correia, A. L. N. Fred, and H. Gamboa, Eds. Vilamoura, Portugal: SciTePress, 1-4 February 2012, pp. 193–198.

[228] G. D. Finlayson, “Color in perspective,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1034–1038, October 1996.

[229] K. Barnard, G. D. Finlayson, and B. Funt, “Color constancy for scenes with varying illumination,” *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 311–321, February 1997.

[230] G. Finlayson and S. Hordley, “Improving gamut mapping color constancy,” *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1774–1783, October 2000.

[231] A. Gijsenij, T. Gevers, and J. van de Weijer, “Generalized gamut mapping using image derivative structures for color constancy,” *International Journal of Computer Vision*, vol. 86, no. 2-3, pp. 127–139, Jan 2010.

[232] M. Mosny and B. Funt, “Cubical gamut mapping colour constancy,” in *CGIV2010 IS&T Fifth European Conference on Color in Graphics, Imaging and Vision*, vol. 5. Joensuu, Finland: Society for Imaging Science and Technology, June 2010, pp. 466–470.

[233] G. D. Finlayson, S. D. Hordley, and I. Tastl, “Gamut constrained illuminant estimation,” *International Journal of Computer Vision*, vol. 67, no. 1, pp. 93–109, April 2006.

[234] A. Gijsenij, T. Gevers, and J. van de Weijer, “Computational color constancy: Survey and experiments,” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2475–2489, 2011.

[235] K. Barnard, “Improvements to gamut mapping colour constancy algorithms,” in *Computer Vision - ECCV 2000*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, vol. 1842, pp. 390–403.

[236] E. H. Land, "The retinex theory of color vision," *Scientific American*, vol. 237, no. 6, pp. 108–28, December 1977.

[237] V. Agarwal, B. Abidi, A. Koschan, and M. Abidi, "An overview of color constancy algorithms," *Journal of Pattern Recognition Research*, vol. 1, no. 1, pp. 42–54, 2006.

[238] E. Provenzi, C. Gatta, M. Fierro, and A. Rizzi, "A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1757–1770, October 2008.

[239] G. Buchsbaum, "A spatial processor model for object color-perception," *Journal of the Franklin Institute-Engineering and Applied Mathematics*, vol. 310, no. 1, pp. 1–26, July 1980.

[240] M. Ebner, "Color constancy based on local space average color," *Machine Vision and Applications*, vol. 20, no. 5, pp. 283–301, Jul 2009.

[241] R. Gershon, A. Jepson, and J. Tsotsos, "From [r;g;b] to surface reflectance: Computing color constant descriptors in images," *Perception*, vol. 17, pp. 755–758, 1988.

[242] M. Ebner, "Combining white-patch retinex and the gray world assumption to achieve color constancy for multiple illuminants," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, B. Michaelis and G. Krell, Eds. Springer Berlin Heidelberg, 2003, vol. 2781, pp. 60–67.

[243] S. Zeki and L. Marini, "Three cortical stages of colour processing in the human brain," *Brain*, vol. 121, pp. 1669–1685, Sep 1998.

[244] J. Herault, "A model of colour processing in the retina of vertebrates: From photoreceptors to colour opposition and colour constancy phenomena," *Neurocomputing*, vol. 12, no. 2-3, pp. 113–129, Jul 31 1996.

[245] H. Helson, "Fundamental problems in color vision i the principle governing changes in hue, saturation, and lightness of non-selective samples in chromatic illumination," *Journal of Experimental Psychology*, vol. 23, pp. 439–476, 1938.

[246] X. D. Xie and K. M. Lam, "An efficient illumination normalization method for face recognition," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 609–617, Apr 15 2006.

[247] J. Ruiz-Del-Solar and J. Quinteros, "Illumination compensation and normalization in eigenspace-based face recognition: A comparative study of different pre-processing approaches," *Pattern Recognition Letters*, vol. 29, no. 14, pp. 1966–1979, Oct 15 2008.

[248] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms - part i: Methodology and experiments with synthesized data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 972–984, September 2002.

[249] G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *12th Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications*, 2004, pp. 37–41.

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

[251] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, January 2005.

[252] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part i - the first 30 years and fundamentals," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, December 2011.

[253] T. S. Huang and A. N. Netravali, "Motion and structure from feature correspondences: a review," *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, Frebruary 1994.

[254] PMD, "pmd[vision] - camboard nano," [Online]. Available: https://www.cayim.com/, [Accessed on May 2013].

[255] MESA, "Mesa imaging ag - swissranger sr4000," [Online]. Available: http://www.mesa-imaging.ch/prodview4k.php, [Accessed on May 2013].

[256] Microsoft, "Kinect for windows," [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/, [Accessed on May 2013].

[257] Asus, "Asus - xtion pro live," [Online]. Available: http://www.asus.com/Multimedia/Xtion_PRO_LIVE, [Accessed on May 2013].

[258] PrimeSense, "Sensors - primesense," [Online]. Available: http://www.primesense.com/solutions/sensor/, [Accessed on May 2013].

[259] J. Smisek, M. Jancosek, and T. Pajdla, "3d with kinect," in *Consumer Depth Cameras for Computer Vision*, ser. Advances in Computer Vision and Pattern Recognition, A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, Eds. Springer London, 2013, pp. 3–25.

[260] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb 2012.

[261] H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," PhD Thesis, Department of Computer Science, Stanford University, 1980.

[262] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," in *IEEE International Conference on Systems, Man and Cybernetics - SMC*, vol. 1, Hawaii, USA, 10-12 October 2005, pp. 903–910.

[263] D. M. Helmick, Y. Cheng, D. S. Clouse, M. Bajracharya, L. H. Matthies, and S. I. Roumeliotis, "Slip compensation for a mars rover," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, vol. 4, Edmonton, Canada, 2-6 August 2005, pp. 1419–1426.

[264] A. Milella, B. Nardelli, D. Di Paola, and G. Cicirelli, "Robust feature detection and matching for vehicle localization in uncharted environments," in *Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), IEEE International Conference on Intelligent Robots Systems - IROS*, St Louis, MO, USA, 11 October 2009, pp. 11–16.

[265] C. F. Olson, L. H. Matthies, M. Schoppers, and M. V. Maimone, "Robust stereo ego-motion for long distance navigation," in *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*, vol. 2. Hilton Head Island, SC, 13 -15 June 2000, pp. 453–458.

[266] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Stereo ego-motion improvements for robust rover navigation," in *IEEE International Conference on Robotics and Automation - ICRA*, vol. 2, Seoul, Korea, 21-26 May 2001, pp. 1099–1104.

[267] A. Levin and R. Szeliski, "Visual odometry and map correlation," in *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*, vol. 1, Washington, DC, USA, 27 June - 2 July 2004, pp. 611–618.

[268] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*, vol. 1. Washington, DC: IEEE Computer Society, 27 June - 02 July 2004, pp. 652–659.

[269] ——, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, January 2006.

[270] P. I. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, vol. 4, Sendai, Japan, 28 September- 2 October 2004, pp. 4007–4012.

[271] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1015–1026, October 2008.

[272] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, October 2008.

[273] A. J. Davison, "Mobile robot navigation using active vision," PhD Thesis, Faculty of Physical Sciences, University of Oxford, 1998.

[274] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, August 2002.

[275] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, "A framework for vision based bearing only 3d slam," in *IEEE International Conference on Robotics and Automation - ICRA*, Orlando, F, USA, 15-19 May 2006, pp. 1944–1950.

[276] T. Lemaire, S. Lacroix, and J. Sola, "A practical 3d bearing-only slam algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, Edmonton, Alberta, Canada, 2-6 August 2005, pp. 2757–2762.

[277] L. M. Paz, J. D. Tardos, and J. Neira, "Divide and conquer: Ekf slam in o(n)," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1107–1120, October 2008.

[278] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway, "Robust real-time visual slam using scale prediction and exemplar based feature description," in *IEEE Conference on Computer Vision and Pattern Recognition - CPVR*, Minneapolis, USA, 17-22 June 2007, pp. 430–436.

[279] M. Tomono, "Robust 3d slam with a stereo camera based on an edge-point icp algorithm," in *IEEE International Conference on Robotics and Automation - ICRA*, Kobe, Japan, 12-17 May 2009, pp. 4306 – 4311.

[280] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments,” in *International Symposium on Experimental Robotics - ISER*, Delhi, India, 18 - 21 December 2010, p. 15.

[281] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the rgb-d slam system,” in *IEEE International Conference on Robotics and Automation - ICRA12*, Vilamoura, Portugal, 14-18 May 2012, pp. 1691–1696.

[282] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS12*, Vilamoura, Portugal, 7-12 October 2012, pp. 573–580.

[283] M. Paton and J. Kosecka, “Adaptive rgb-d localization,” in *9th Conference on Computer and Robot Vision - CRV12*, Toronto, Canada, 28-30 May 2012, pp. 24–31.

[284] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for rgb-d cameras,” in *IEEE Conference on Robotics and Automation - ICRA*, Karlsruhe, Germany, 6-10 May 2013.

[285] OpenNI, “Openni sdk,” [Online]. Available: http://www.openni.org/openni-sdk, [Accessed on May 2013].

[286] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, April 1991.