Rui Pedro da Silva Nóbrega

Mestre em Engenharia Informática

Interactive Acquisition of Spatial Information from Images for Multimedia Applications

Dissertação para obtenção do Grau de Doutor em Informática

Orientador : Professor Doutor Nuno Manuel Robalo Correia,
Professor Catedrático,
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa

Júri:

Presidente: Professor Doutor Pedro Manuel Corrêa Calvente Barahona

Arguentes: Professor Doutor Hartmut Seichter
Professor Doutor João António Madeiras Pereira

Vogais: Professor Doutor Nuno Manuel Robalo Correia
Professora Doutora Maria Beatriz Alves de Sousa Santos
Professor Doutor Manuel João Toscano Próspero dos Santos

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

December, 2013

Interactive Acquisition of Spatial Information from Images for Multimedia Applications

Aos meus pais
e querida irmã

Acknowledgements

The last years were very intense and fulfilling, learned many things, made friends and discovered the world. That said, there are many people that helped me in this adventure and to whom I have to sincerely thank.

I would like, first of all, to thank my supervisor Professor Doutor Nuno Manuel Robalo Correia for all the guidance and friendship through this journey. Without his valuable input and support this thesis would not be possible. I also would like to thank, the valuable contribution of my thesis committee composed by Professor Doutor João António Madeiras Pereira and Professor Doutor Manuel Próspero dos Santos. Special thanks for everyone at my research group IMG, research center CITI-FCT-UNL and to the Fundação para a Ciência e Tecnologia (FCT research grant SFRH/BD/47511/2008).

All these years of research could not be possible without the undeniable effort of my family and friends. For that reasons I would like to thank to my parents Guida Silva e João Nóbrega, to my sister Cláudia Nóbrega, to my grandparents, uncles and cousins (specially my thesis partner João Teixeira), which supported me all the way.

To all my friends for their craziness, support and deviance: André Sabino, Sara Gancho, Sérgio Braz, Bruna Oliveira, Fátima Bernardes, Filipa Peleja, Bruno Cardoso, Ricardo Cebola, Pedro Gonçalo, Joana Martinho and Sandra António.

I have to thank to my colleagues, professors and friends at IMG, CITI and TU-Graz: Diogo Cabral, Rossana Santos, Carmen Morgado, Rui Madeira, Sofia Reis, Luís Silva, Carlos Nobre, Bárbara Teixeira, Guida Piriquito, Hugo Vieira, Duarte Gonçalves, Rui Jesus, Leonor Oliveira, Hartmut Seichter, Raphael Grasset, Eduardo Veas, João Magalhães, Armanda Rodrigues, Teresa Romão, Sofia Cavaco, all the FCT staff and everyone I may have forgotten. Thank you!

Abstract

This dissertation addresses the problem of creating interactive mixed reality applications where virtual objects interact in a real world scenario. These scenarios are intended to be captured by the users with cameras. In other words, the goal is to produce applications where virtual objects are introduced in photographs taken by the users. This is relevant to create games and architectural and space planning applications that interact with visual elements in the images such as walls, floors and empty spaces. Introducing virtual objects in photographs or video sequences presents several challenges, such as the pose estimation and the visually correct interaction with the boundaries of such objects. Furthermore, the introduced virtual objects should be interactive and respond to the real physical environments. The proposed detection system is semi-automatic and thus depends partially on the user to obtain the elements it needs. This operation should be significantly simple to accommodate the needs of a non-expert user. The system analyzes a photo captured by the user and detects high-level features such as vanishing points, floor and scene orientation. Using these features it will be possible to create virtual mixed and augmented reality applications where the user takes one or more photos of a certain place and interactively introduces virtual objects or elements that blend with the picture in real time. This document discusses computer vision, computer graphics and human-computer interaction techniques required to acquire images and information about the scenario involving the user. To demonstrate the framework and the proposed solutions, several proof-of-concept projects are presented and studied. Additionally, to validate the solution several system tests are described and each case-study interface was subject of different user-studies.

Keywords: Mixed and Augmented Reality, Computer Vision, Computer Graphics, Human-Computer Interaction.

Resumo

Esta dissertação aborda o problema da criação de aplicações interactivas de realidade mista e aumentada onde objetos virtuais interagem num cenário real. Por forma a capturar o cenário os utilizadores deverão utilizar dispositivos com câmaras. Por outras palavras, o objectivo é a produção de aplicações onde os objectos virtuais são introduzidos em fotografias tiradas pelos utilizadores. Isso pode ser interessante para criar jogos e aplicações de planeamento de espaços e arquitectura que interagem com elementos visuais das imagens, tais como paredes, chão e espaços vazios. Mostrar objectos virtuais em fotografias ou sequências de vídeo apresenta vários desafios como a estimativa de pose e a detecção visualmente correcta dos limites de interacção do cenário. Além disso, os objetos virtuais introduzidos devem ser interactivos e responder naturalmente aos ambientes físicos reais. O sistema de detecção proposto é semi-automático e, portanto, depende do utilizador para obter os elementos de que necessita. Esta operação deve ser significativamente simples para acomodar as necessidades de um utilizador não-especialista. O sistema analisa uma foto tirada pelo utilizador e detecta características de alto nível, tais como pontos de fuga, piso e orientação de cena. Usando esses recursos, é possível a criação de aplicações de realidade mista e aumentada, onde o utilizador tira uma ou mais fotos de um determinado lugar e de forma interactiva introduz objectos virtuais ou elementos que se misturam com a imagem em tempo real. Este documento discute várias técnicas de visão por computador, computação gráfica e de interacção pessoa-máquina necessários para adquirir imagens e informações sobre o cenário. Para demonstrar as soluções propostas, várias aplicações são apresentadas e estudadas. Além disso, para validar os vários conceitos são apresentados testes de sistema bem como estudos de usabilidade para cada interface proposta.

Palavras-chave: Realidade Mista e Aumentada, Visão por Computador, Computação Gráfica, Interacção Pessoa-Máquina.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

Symbols

α axis rotation.

δ low-level features.

θ rotation of r.

λ eigenvalue.

σ Gaussian scale parameter.

φ high-level features.

φ_1 vanishing points.

φ_2 scene orientation.

φ_3 floor.

C camera parameters.

d Euclidean distance.

DoG difference of Gaussians.

E energy function.

e number of Canny edges detected.

f focal length or function f.

g function which calculates φ.

G Gaussian kernel convolution.

H homography matrix.

h line sets detected for each edge.

I intensity or image.

M_v modelview matrix.

M logic model.

$m_{\varphi 3}$ floor mask.

m slope.

O virtual objects.

P projection model matrix.

R Harris score.

r normal vector connecting line to origin.

s score function.

T transpose.

v viewport matrix.

w weight function.

Abbreviations

ANOVA ANalysis Of VAriance.

AR Augmented Reality.

BRIEF Binary Robust Independent Elementary Features.

CAD Computer-Aided Design.

CG Computer Graphics.

CV Computer Vision.

FAST Features from Accelerated Segment Test.

FLANN Fast Library for Approximate Nearest Neighbors.

GPS Global Positioning System.

HCI Human-Computer Interaction.

MR Mixed Reality.

MSER Maximally Stable Extremal Region.

ORB Oriented BRIEF.

PTAM Parallel Tracking and Mapping.

RANSAC RANdom SAmple Consensus.

RQ Research Question.

SIFT Scale-Invariant Feature Transform.

SLAM Simultaneous Localization and Mapping.

SURF Speeded-Up Robust Features.

VR Virtual Reality.

1

Introduction

Mixing reality with virtual digital elements is an idea that evolved from the study of virtual environments and has improved with the increase in computer processing power and in image sensing devices. Another important factor that has contributed for the development of mixed and augmented reality systems [Azu97; Val98], is the availability of mobile devices, which allow taking cameras with processing power into the user's daily life.

Looking outside a window, an imaginative mind could wish to play a computer game in the real world, look into a picture and enter that picture's scenario or any other form of improved, enhanced, augmented, and mixed reality [TP12]. With the improvement of mobile processing power and the wide availability of cameras anywhere this is becoming possible to do outside laboratories or niche technical markets. To implement these systems the applications need to see and create a model of the world by understanding images through computer vision and image processing techniques [Aga+11].

The work presented in this dissertation is mainly concerned with the discussion of how images or video frames can be used as input for interactive applications. Questions to answer include, what value can this kind of input provide, what are the technical implications and what kind of problems arise for the users of such systems.

There are already several multimedia systems that use images and videos as input, or augmented reality systems that can present geo-referenced content in the context of a camera image [Ra+04]. Several other AR systems present virtual

content on top of special markers [Met13; Vuf13]. There are car applications that display the path on the window and detect traffic signs. Other systems detect faces, tags or other features, which will be later discussed in this thesis.

Multimedia computer vision based systems are increasingly entering commercial applications. With the help of several types of cameras (smartphone cameras, webcams, photographic cameras, camcorders) and additional sensors (depth sensors, accelerometers, GPS), it is possible to detect, track and capture different scene features, objects and elements. Taking advantage of this, systems can capture real-world scene elements and use them in interactive applications using as input real-time video, recorded video, and images [LHK09].

Cameras are everywhere with multiple purposes: camcorders are used to record memories, cell phone cameras take pictures and videos of spontaneous interesting moments, and webcams have many applications such as video chat. With the rise of the new interaction mechanisms in video consoles, cameras start to be used in the living room and other unexpected places. Tablets and smartphones have frontal and back cameras that can capture the world and feed an application with visual data.

How is it possible to use this visual acquisition power to integrate services in interactive applications, where the user can add visual content to the captured scene seamlessly and effortlessly? This would enable for a person to create new space designs, create augmented realities in virtual space or time, and run applications and games that mix reality with custom content.

This dissertation presents a mixed reality framework with algorithms and techniques to enable the acquisition of visual features from user-captured images and used to create interactive applications of virtual spaces. One of the applications of the virtual captured scene can be used in mixed reality, with virtual reality being superimposed on the captured real-world scenes (images and video). Mixed and augmented reality in applications and specifically in games has been increasing in recent years with several commercial examples in game consoles (e.g., Eye-toy, Microsoft Kinect) and tablet/smartphone applications. There are several toolkits (e.g., ARToolKit [ART03]) and frameworks to develop applications. Most Augmented Reality (AR) systems obey to certain principles, as they all use some form of computer vision and either have a fixed or a free-moving camera. In free-moving camera systems the augmented reality elements are usually associated with pre-defined marker (visual or GPS point). Fixed-camera applications usually use pre-built models of the scene.

The main motivation of this work is to create ubiquitous computer vision and

image processing systems that mix virtual content with real world scenarios without using predefined markers or objects. Instead, the improved reality should be based in image recognition and automatic scene reconstruction. Virtual objects should recognize real world elements and react accordingly to scene elements. It is exactly the combination of image analysis and recognition with the mixed and augmented reality concept that is being explored.

The main goal of this work is to create a framework that enables the user to virtually reshape a photographed scene, effectively choosing the scenario where the multimedia application will take place [KHF11; LHK09]. By adding virtual objects, or virtual content, the user can transform the scene into something completely different. In this dissertation the main focus will be given to the introduction of virtual objects that react to the scene as if they had physical properties such as mass or weight. The applications of this system could be the creation of furniture testing simulations in user spaces, virtual reshaping of old photos, or spatial-aware games. The next sections describe and formalize the main problems and proposed solutions as well as the contributions of this research work.

1.1 Research Questions

The main research question that is being addressed in this thesis is:

> ***How to create interactive mixed reality applications that use photographs taken by the users where virtual content takes advantage of visual features detected in the scenario***.

From this question it is possible to define the main addressed topics. These will be the use of *multimedia interactive applications*, *photos*, *mixed reality* and *image processing and analysis*.

From this broader problem several research questions arise. First, is it feasible to create interactive mixed reality applications were virtual objects, characters and elements interact in a real world scenario? Secondly, can the virtual content interact with the visual content using the properties of the real world? Finally, are common users (with no special expertise) capable and motivated enough to perform the necessary steps to initialize this virtual model environment?

For each research question there are several hypothesis that can possibly be presented as a solution for each problem. For completeness a null hypothesis is also provided.

Research Question 1 *Is it feasible to create interactive mixed reality applications where virtual objects interact in non pre-defined real world scenario?*

There are several examples in the state-of-the-art work to conclude that mixed reality is possible and feasible. The main question is if this is possible in a non pre-defined scenario without markers or special localization references. There are some examples of systems but there are also several open questions on what is the best way of initializing a mixed reality system and what features should be tracked or detected and how virtual content should be combined with reality. The goal is then to propose a new method to automatically create a geometrically coherent presentation of virtual objects in photographs.

Research Question 2 *Can the virtual content interact with the visual content using the properties of the real world in a non pre-defined scenario?*

Several examples in the literature present applications where virtual objects are combined with reality. There are a few projects that explore interactivity relating the physical properties of the scenario in the real world and the apparent properties of the rendered virtual objects. The goal is to identify properties that can be automatically detected in any scenario and apply them to the virtual 3D objects.

Research Question 3 *Are users capable and motivated enough to perform the necessary steps to use and initialize this virtual model environment?*

The goal is to identify situations where the proposed system makes sense and what kind of steps are users able and motivated to do without destroying the experience and the immersion with the mixed reality system. Several markerless systems have different initialization workflows. The goal here is to propose and evaluate different interaction paradigms to use and start the system.

Altogether, the main research questions can be divided into three sub problems: (1) *scenario acquisition*, (2) *visual processing* and (3) *interaction*.

In the (1) scenario acquisition, the challenges are all about how to capture the world. Which device (smartphone, tablet or webcam)? Which modality should be used (video input, image input, text input)? Has the image enough quality and visual content? What extra sensorial data should be acquired (GPS, accelerometer)? What extra information does the user have to provide? How is all the information gathered and stored?

The (2) visual processing of the scene presents several challenges. Which features should be detected? Which ones should be tracked? What can be inferred

from the detected features? How is the virtual model composition? Is there more than just 3D geometry information?

At the (3) interaction level it is important to help the user to create the virtual model and to explore interaction metaphors that can bring real world properties into the virtual applications (e.g., gravity and mass).

In the remaining part of this document, the research questions will be referred to as: **RQ**, for the main question, and **RQ1**, **RQ2** and **RQ3**, for the subquestions.

1.2 Research Overview

In order to address the research questions several solutions were designed, implemented and tested. The main novelty of the proposed solution is the proposal of a system that uses several techniques from different research areas to build multimedia applications with user-captured images. Besides the creation of virtual models, the interaction with the space takes into consideration real-world objects and their properties to create multimedia applications.

The different researched techniques are detailed in the third and fourth chapters. The initial research was very centered on the intrinsic properties and features of images and videos. The first prototypes implement solutions to detect a certain image in a collection of images, detect the relation between images of the same scene, describe an image through point descriptors and superimpose virtual objects using images as markers.

This was important to study what type of elements should compose the virtual scene model and led to more advanced systems such as the virtual exploration of spaces using panoramic images and the introduction of virtual content in video frames using a visual anchor for the objects.

A large portion of the research was spent studying point descriptors in images and their applications. This led to the creation of virtual scene models using stereovision to calculate virtual 3D depth.

The main solution proposed in this thesis is a mixed reality framework based on an initial image captured by the user. The image is analyzed to find the floor, vanishing points and room orientation. The mixed and augmented reality proposed solution introduces virtual objects with a guideline and snapping system to assist the user in the introduction of objects. This system is based on the analysis of the image lines (as in Figure 1.1), perspective and main visual segments and was implemented in an interactive system presented in Chapter 4. The main contributions of this work are the introduction of a semi-automatic method for

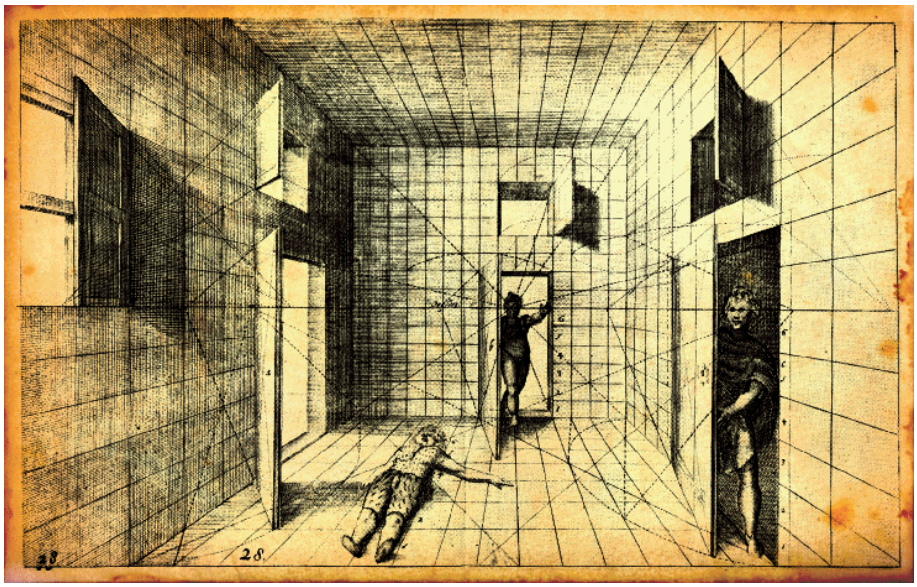


Figure 1.1: Perspective projection study by Hans Vredeman de Vries (1527-1607) architect, painter, and engineer. All parallel lines have the same vanishing point. (Source: [Perspective print by Vredeman de Vries], published in The Hague, 1604.)

high-level feature extraction and several use-case prototypes, which implement and make use of the given features. Figure 1.2 presents a preview of the created system where a virtual box is automatically added to the scene on the floor.

To illustrate the capabilities of the mixed reality system, a game was implemented to showcase the main key points of the framework. This is a classic two-player snake game where players must avoid at all costs bumping into the boundaries of the game or any of the snakes. The main novelty in this version is that the game can be played in any real-world flat surface that the players can imagine. This can be a table, the living room floor, the back yard or the main street. Figure 1.3 presents a preview of this game. Additionally, the users can build their own custom snake maze using sheets of paper or any other flat uniform material. This application is further discussed in section 5.2.

Additional prototypes were created to demonstrate different applications that can be take advantage of image and video input. These applications present alternative views from the main proposed solution. Figure 1.4 presents a case study of virtual exploration of a space from images, a website, created to explore a museum exhibition through homography of images obtained through feature extraction. This system will later be explained in detail in section 5.3.

Figures 1.2, 1.3 and 1.4 represent a preview of what are some of the results of this work.

Figure 1.2: Preview of the results from the virtual insertion of objects in photographs.

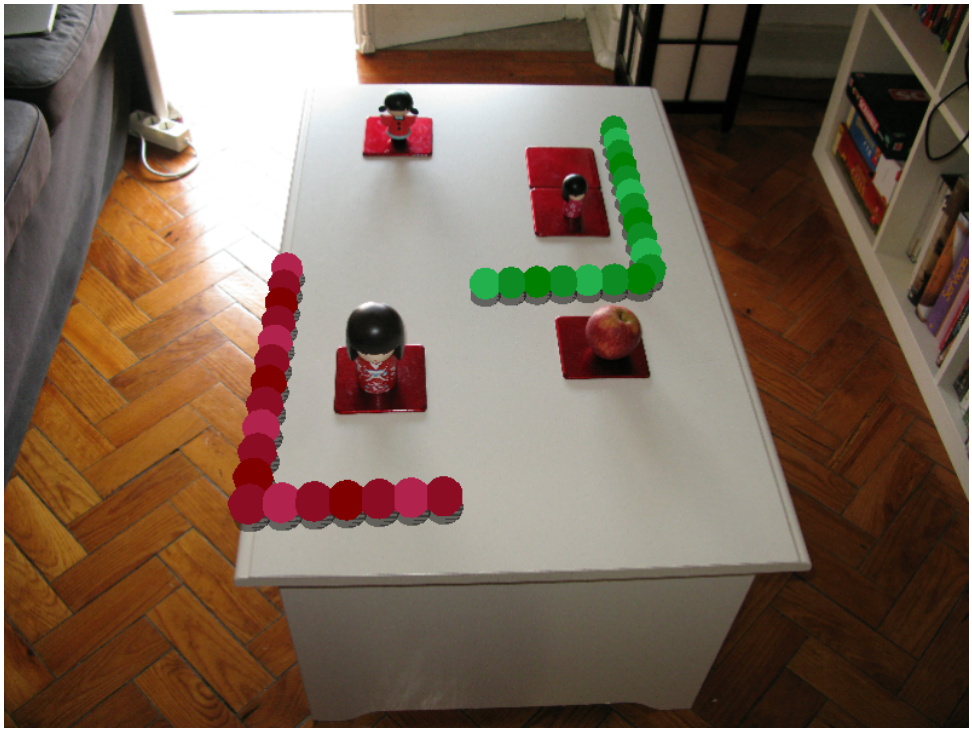


Figure 1.3: Preview of a virtual snake game running on a real-world maze made on a living room table.

Figure 1.4: Preview of a website where the navigation system was automatically created by analysing the relation between images.

1.3 Contributions

The main contribution of this work is the development of the previously mentioned solution for a markerless system intended to be used to build mixed reality applications based on photographs. The current work presents several improvements upon existing systems. These can be summarized in several key points: (1) the integration of several technologies from several fields as described in the previous section, (2) the integration of detection algorithms with a human assisted interface, (3) the proposal of a vanishing point detection algorithm for images with several redundancies (presented in Chapter 4) and (4) the proposal of several mixed reality interfaces that take advantage of the detected scene elements. Throughout this document the main innovations will be highlighted and compared, at each step, with related state-of-the-art systems.

Computer vision and interactive graphics integration: With the goal of building better multimedia systems for mixed reality applications concepts from different research fields were used. This led to an effort of integration where computer vision algorithms were researched and integrated with computer graphics to produce interactive applications.

Human assisted detection algorithms: One of the novelties of this work is that

it discusses and proposes several different strategies for the initialization of markerless mixed reality systems, especially systems where the user wants to acquire a non-predefined space without requiring that special markers or objects to be present in the world. This is done by detecting features from the environment and including the user in the process by asking simple questions about the scenario. This is further discussed in Chapter 4 and an example is presented in section 5.2.

Virtual object insertion in photos: Another development is the proposal of a framework to introduce virtual objects in photos. The algorithm presented in Chapter 4 is based on vanishing point detection using several levels of redundancy to accommodate different lightning and image quality conditions. The algorithm combines robust line detection with RANSAC, cluster detection, and 3D fitting for small rotations. This allows the introduction of virtual objects that are naturally oriented with the photographed scene layout and in the same perspective.

Mixed reality interfaces: The previous algorithm allows the construction of different multimedia applications with their own challenges and advantages. Using the proposed framework several different user interfaces are proposed to take advantage of the detected scenario from photographs. This can be seen in detail in Chapter 5.

1.4 Scope

This dissertation describes a multimedia solution (using cameras, user interaction and image processing) for applications that use real-world images. Using video or live cameras can be considered as a sub-problem of this. The extension to video of the proposed solutions is discussed but that is not the main focus of this work.

Computer vision and image processing algorithms were used and combined in different manners to create a detection system for several elements in photos. Many basic detected features (e.g., line segments) used previous implementations of such algorithms (e.g., Hough line transform [Hou62]) from libraries such as OpenCV [Ope13].

Throughout the text it is mentioned that the features are stored in the Model that describes the scenario. The use of the word Model should be understood as a logic Model with all the information and features required to create the mixed reality applications. It is a Model as in the Model-View-Controller paradigm and it

should not be confused with a 3D mesh reconstructed scene. The logic model can contain 2D information from the photo, point features as well as 3D information.

The proposed solution is meant to detect three-dimensional elements in the scene to superimpose 3D meshes on top of a photograph. There will be only one viewing angle, the angle of view of the photograph. The goal is not to reconstruct the 3D scene entirely and add extra 3D meshes. Although interesting, that is a different problem.

The study on interfaces focuses more on what should be presented to the users and in what order so that they can initialize the mixed reality correctly. In order to help the scene detection algorithms, the objective of the interface design is to discover what kind of input methods and annotations are users comfortable with.

1.5 Research Work

The work presented in this dissertation led to the development of several applications and documents. The contributions can be summarized as one mixed reality framework for photographs (Chapter 4), three proof-of-concept applications (Chapter 5) and several exploratory applications of key-concepts (Chapter 3).

The three applications (seen in Figures 1.2, 1.3 and 1.4) and the framework were subject to several user studies and performance studies. Finally all the experiments and proposed concepts were published in nine peer-reviewed publications:

[NC13a] Nóbrega, R., Correia, N. 2013. Dynamic Insertion of Virtual Objects in Photographs. In International Journal on Creative Interfaces and Computer Graphics (IJCICG), IGI-Global, To be published in 2013.

[NC13b] Nóbrega, R., Correia, N. 2013. Photo-based Multimedia Applications using Image Features Detection. In Proceedings of International Conference on Computer Graphics Theory and Applications (GRAPP'13). Barcelona, Spain, 298-307.

[NC12c] Nóbrega, R., Correia, N. 2012. Interactive Insertion of Virtual Objects in Photos and Videos. In Proceedings of Eurographics 2012 – Posters (EG'12). Eurographics Association, Cagliari, Sardinia, Italy, 7-8.

[NC12b] Nóbrega, R., Correia, N. 2012. Inserção Dinâmica de Objectos Virtuais no Contexto de Fotografias Tiradas por Utilizadores. In Proceedings of EPCG 2012 (EPCG '12). GPCG, Viana do Castelo, Portugal,103-106.

[Nó+12] Nóbrega, R., Correia, N., Nobre, C., Teixeira, A. B., Oliveira, L., and da Silva, R. H. 2012. Navigation in Past Museum Exhibitions using Multimedia Archives. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12). ACM, Capri, Italy, 778-779.

[NC12d] Nóbrega, R. and Correia, N. 2012. Magnetic Augmented Reality: Virtual Objects in Your Space. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12). ACM, Capri, Italy, 332-335.

[NC12a] Nóbrega, R. and Correia, N. 2012. Smart Interface for Reshaping Photos in 3D. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces (IUI '12). ACM, Lisbon, Portugal, 315-316.

[NC11] Nóbrega, R. and Correia, N. 2011. Design Your Room: Adding Virtual Objects to a Real Indoor Scenario. In Proceedings of CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11). ACM, Vancouver, Canada, 2143-2148.

[Nó11] Nóbrega, R. 2011. Modeling Places for Interactive Media and Entertainment Applications. In Proceedings of CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11). ACM, Vancouver, Canada, 1081-1084. (Doctoral Consortium of CHI'11)

All these papers represent different stages of the research. The most recent conference and journal papers present the most complete approach to the developed systems. Although not directly related with the current work, presented in this thesis, the following papers were produced in the early stages of the research:

[Cor+10] Correia, N., Mota, T., Nóbrega, R., Silva, L., Almeida, A. 2010. A Multitouch Tabletop for Robust Multimedia Interaction in Museums, In ACM International Conference on Interactive Tabletops and Surfaces (ITS '10). ACM, Saarbrücken, Germany, 117-120.

[NC09] Nóbrega, R. and Cavaco, S. 2009. Detecting Key Features in Popular Music: Case Study – Singing Voice Detection, In Procedings of Second International Workshop on Machine Learning and Music (MML '09) at ECML-PKDD'09, Bled, Slovenia, 7-12.

The first introduces [Cor+10] the concept of using image processing to detect fingers in multi-touch tables. Although the main focus is on tabletops it is an introduction to computer vision techniques. The second [NC09] is an essay on how to use machine learning to detect certain features, with an emphasis on sound.

Several parts of this work (including working demos [NC12a; Nó+12]), have been presented in these international conferences. Additional presentations and discussion have taken place in a doctoral consortium (CHI'11) [Nó11], a summer school (ICVSS'11)[1] and talks were given in other institutions such as Universidade da Madeira [2], IHA-FCSH-UNL[3] and Biblioteca da Fundação Calouste Gulbenkian[4].

This thesis work was discussed and presented in a short-term research visit to the Technical University of Graz in Austria[5]. In this stay at ICG, this dissertation was improved with the valuable input of computer graphics and computer vision experts. Additionally, the work was presented in the Winter Augmented Reality Meeting (WARM'13)[6] which was held at the same university.

The main contribution, as previously stated, is the creation of framework for the creation of a mixed reality application from high-level features detected from a single image. This framework is explained in detail in Chapter 4. It includes a capturing system based on FAST descriptors, it allows the creation of interactive applications in a space chosen by the user and it contains a semi-assisted detection where the user roughly sketches a line around the floor. The system provides automatic detection of high-level features such as floor, vanishing point and room orientation. The introduction of virtual objects can be done taking in consideration the floor and the orientation with the room. To evaluate the reliability of the detection system, a system study was conducted with the use of external databases as ground truth.

As stated before there are three main prototypes presented in Chapter 5: Magnetic Objects, Snake Game and Museum Navigation.

The magnetic augmented objects prototype uses the implemented framework with an interactive tool for positioning objects in 3D in a scenario chosen by the

[1]ICVSS'11, International Computer Vision Summer School 2011, `http://svg.dmi.unict.it/icvss2011/`.

[2]Universidade da Madeira, `http://www.uma.pt/`.

[3]IHA, Art History Institute at FCSH, NOVA University of Lisbon, `http://iha.fcsh.unl.pt`

[4]Calouste Gulbenkian Foundation Library, `http://www.biblarte.gulbenkian.pt/`.

[5]TUGraz, Technical University of Graz, `http://www.tugraz.at/` at ICG - Institute of Computer Graphics and Vision, `http://www.icg.tu-graz.ac.at/`

[6]WARM'11, Winter Augmented Reality Meeting, `http://studierstube.icg.tugraz.at/WARM13/`.

user. This prototype is presented in section 5.1. It presents the concept of magnetic virtual objects, which follow real-world properties, by bumping into the floor and aligning with the scenario. Guidelines assist the user to position the 3D objects by indicating that the objects have hit the floor and are oriented with the scene. There is a snapping system that when the objects are almost in position, they snap to the floor and to the orientation of the scene. Using this system it is possible to devise a furniture positioning application. A user study evaluates the performance gain of the guidelines and snapping system in the positioning of virtual objects in a mixed reality scene.

The Snake game showcases interaction concepts that can result from the proposed framework. This prototype is presented in section 5.2. It introduces the concept of playing a game in any surface and allows the creation of custom game levels using real objects. It has a semi-assisted initialization of the game to test the user introduction of images. The game, played by two players, has two snakes that move around in a user-defined space from a photograph. The goal is to survive as much time possible without hitting objects or snakes. A user-study evaluates the construction of the model and the engagement of the users with this type of systems.

The museum exhibition from the past website, featuring a navigation system based on the study of homographies between images to create an explorable virtual space, is presented in section 5.3. It introduces the concept of reconstructing an exhibition from the past, presenting a use-case for the digital reconstruction of the first modern art exhibition in Portugal organized by Calouste Gulbenkian Foundation in 1957. The panoramic navigation system is constructed semi-automatically. It is a collaboration with the Institute of Art History at FCSH-UNL and Biblioteca Fundação Calouste Gulbenkian. A system study was realized on the detection of an image in a larger image set and the comparison of every image with each other. A user study was also produced to evaluate the interface (including expert users).

Several prototypes were developed (some published in the online blog[7]) and are presented in Chapter 3: real time panorama builder using webcam or video (section 3.4), augmented reality in webcam or video using an image as a marker (section 3.2), distance measurement using stereo-vision and poster pinning to the scene with correct normals using stereo-vision (section 3.5).

Additional contributions and collaborations will be described throughout the document whenever they are relevant.

[7]Rui Nóbrega's Ph.D. blog, `http://doutorandoemfilosofia.blogspot.com`.

1.6 Summary and Document Structure

This chapter presents the motivation and research topics of this thesis. The goal was to introduce the main concepts that will be approached throughout the document. The first section presents the research questions and discusses the main hypothesis. The Research Overview section presents a preview of the proposed solutions in order to highlight the main research areas and motivate the reader for what will be presented in this dissertation. After that, section 1.3 describes the contributions and highlights the innovations. The intention of the scope section is to narrow the research fields that will be addressed, and discuss some of the base assumptions. The Research Work section (section 1.5) lists all the contributions of this work.

The document has the following structure:

- **Chapter 1. Introduction:** the main motivations of the research work, the problem statement, proposed solutions and contributions.
- **Chapter 2. Related work:** state-of-the-art solutions and projects that inspired the current work. Reference to several publications that are listed from four perspectives: generic computer vision, scene acquisition, mixed and augmented reality and interface and interaction design.
- **Chapter 3. Algorithms and Researched Techniques:** additional related work with detailed explanation of techniques and algorithms that were used in the next chapters. The goal is to present background details about each relevant algorithms to better introduce the proposed solution in the next chapters. Several proof-of-concept prototypes are also presented.
- **Chapter 4. Inserting Virtual Objects in Images:** presentation of the developed algorithms and techniques, with system evaluation. In this chapter the proposed solution for the research questions is presented in detail.
- **Chapter 5. Applications:** applications which apply the techniques developed in Chapter 4 and studied in Chapter 3. Several user studies are presented with critical discussion.

  Section 5.1. Magnetic Augmented Objects in Photos: Prototype of application with magnetic objects in photographs taken by the user.

  Section 5.2. Mixed Reality Snake: Game which runs on a photograph taken by the user.

Section 5.3. Past Museum Exhibition Navigation Through Overlapping Images: Website in which the navigation depends on the visual relation between images.

- **Chapter 6. Conclusions:** final conclusions with an evaluation of the research work, implications that result from it and future directions.
- **Appendix A. List of Websites:** list of websites: libraries and frameworks, products and other websites.
- **Appendix B. List of Videos:** videos of the applications.
- **Appendix C. List of Datasets:** list of image datasets.

2

Related Work

There are several projects and research proposals, which were important for the current work. This chapter presents a review of the state-of-the-art in relevant key areas related with this thesis. These include topics in image processing, computer vision, mixed and augmented reality, and human computer interaction. Throughout this chapter, the most influential proposals in the topics of the dissertation, will be presented. For each topic several related projects will be indicated and their relation with the current dissertation explained. Some of these projects contain techniques used in the current work, while others present different approaches to achieve similar goals. Each of the main topics is introduced along with several indications about its relevance to the proposed work in the later chapters.

The chapter is divided into four main sections, and each section approaches the related work from a different, sometimes overlapping, perspective. Section 2.1 presents a survey of several computer vision and image processing algorithms: how are images formed, what features describe them and what kind of applications exists. The survey is focused on image reconstruction, photo-stitching and features acquisition. Taking into account that the main research question **RQ**, section 1.1) is especially focused on scene acquisition from photographs a special section, section 2.2, was dedicated to the different techniques used for the acquisition of the spatial information from photos. The survey is divided by type of input: single image, several images and images with additional sensors. This was important to provide answers to the first research question

(**RQ1**).

Section 2.3, presents the related work from a mixed and augmented reality perspective, combining computer vision aspects with applicational and computer graphics concepts. Several systems and applications are presented categorized by type of approach. Mixed and augmented reality approaches include: GPS/sensor-based, marker-based, feature-based and others less common approaches. The relations between the virtual systems and reality are discussed in order to address **RQ2**.

Finally, these techniques require a specific language to communicate with the user. To answer **RQ3**, an engaging interface has to be created in order to help the user initialize the scene acquisition (section 2.2) and interact seamlessly with the mixed reality system (section 2.3). In section 2.4, several projects are presented from an interface and interaction point of view. The presented related work will be later revisited in Chapters 4 and 5 whenever relevant.

2.1 Computer Vision and Image Processing

In this section, the techniques, which depend on cameras or images, are detailed. Computer vision [FP02; Sze10] is the process of analyzing the images or frames from cameras and obtaining from them some form of measurable knowledge. It is *"an enterprise that uses statistical methods to disentangle data using models constructed with the aid of geometry, physics and learning theory"* [FP02]. It is usually quite dependent of image processing algorithms, although the defining line between the two areas is not clear. Most of the times, computer vision or machine vision is associated with the robotics field and 3D detection while image and signal processing is associated with computer graphics 2D image manipulation.

Before exploring the most important projects in this area, some concepts about cameras and the image formation are summarized in section 2.1.1. The most important features and descriptors, which are used to describe images are presented in section 2.1.2. Section 2.1.3 present some of the more influential projects related with images and photos with special focus in techniques such as photo-stitching, panorama creation and depth reconstruction from images. The same analysis was performed for projects using video (section 2.1.4).

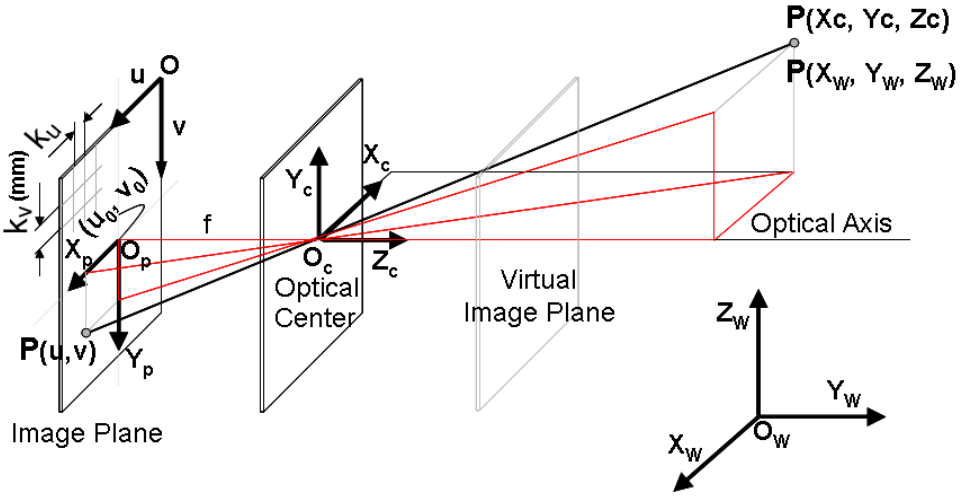


Figure 2.1: Pin-hole projection model.
(Source: Santolaria et al. 2009 [San+09])

2.1.1 Image Formation, Projection and Lens Calibration

Images are captured by the lenses of cameras and converted to pixels by a grid sensor. The process of image formation and the models, which describe the process can be seen in detail on the first chapter of Forsyth and Ponce's book [FP02] and in Hartley and Zisserman's book [HZ04]. The pin-hole projection model, seen in Figure 2.1, is one of the most used camera models. In this image the optical center represents the lens of the camera where the light is focused. The image plane is behind the lens and represents the image in pixels. To every point $P(X, Y, Z)$ in the real world, there is a projection $P(u, v)$ in the image plane. The image plane is an inverted representation of the real world. That is the reason why sometimes the image plane is represented by the virtual image plane seen in Figure 2.1.

Using the perspective model it is possible to convert real world coordinates (X, Y, Z) into plane coordinates (u, v) using

$$u = f\frac{X}{Z} \tag{2.1}$$

$$v = f\frac{Y}{Z} \tag{2.2}$$

where f is the focal length of the camera. This is an approximated simplified model of the perspective, but from it, is possible to perceive that farthest points (higher Z) will tend to have a small displacement from the origin (represented in Figure 2.1 by (u_0, v_0). A common form of representing the perspective model is

using the intrinsic and extrinsic parameters of the camera. The intrinsic parameters are the parameters which are related with the camera manufacture. They are represented below by a 3×3 matrix where f is the focal length and (u_0, v_0) is the principal point, the center of the image in pixel coordinates.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_1 \\ r_7 & r_8 & r_9 & t_1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

The extrinsic coordinates are represented by a 3×4 matrix (there are other models which use a 4×4 matrix), which represents the rotation and translation of a point. The variable s represents a possible scale factor. From these equations it can be observed that the focal length f is very important. If incorrect, it can distort the field-of-view and cause problems in scene detection or for the correct superimposition of objects in mixed reality applications.

Another problem that must be taken into consideration with cameras is lens distortion. Most cameras do not have a linear distribution between viewing angles and pixels. This distortion is larger in the corners of the image causing a fish-eye effect that can visually be observed when straight lines in the center start to present a slight curvature in the borders. Short focal length cameras (with high field-of-view) have a higher tendency to have a large distortion.

Many applications depend on the analysis of an undistorted version of the captured image. To find the images' focal lengths f_x and f_y, the principal point (u_0, v_0) and the distortion map several calibration algorithms have been presented [BDS10; PKG98; San+09].

One of the most common forms of calibration is used by the OpenCV [Ope13] library. It uses a known pattern such as a chessboard to calibrate the system. The board has a known physical size and a fixed number of repeating squares. When capturing several images of the board it is possible to map the distortion of the squares and approximate the focal length. Using a similar principle Baronti et al. [BDS10] describe a calibration process using Lego pieces.

There are numerous attempts [HZ04] to create methods that do not depend on special patterns. Many take advantage of structure from motion captured from short videos [PKG98], other use additional hardware such as lasers [San+09].

2.1.2 Image Features

After acquiring an image one of the main problems is to describe and compare images. As stated by Richard Szeliski, *"feature detection and matching are an essential component of many computer vision applications"* [Sze10]. Image features allow pinpointing similar points across different images of the same scene. One of the most explored image features is SIFT (Scale Invariant Feature Transform) developed by David Lowe [Low04]. Using this technique it is possible to describe a single image, saving only the most significant points of the image and their descriptors. Extracting the most important points can use algorithms such as the Harris points [HS88] or the more advanced Shi-Tomasi detector [ST94]. After that, for each point, the SIFT descriptor is extracted by evaluating several areas around the point at different scales. The method is described in detail in the next chapter in section 3.1.

Using SIFT descriptors for matching between images is much faster and reliable than comparing the pixels directly. SIFT is known for being a robust feature, invariant to scale and rotation (subsection 3.1.2). Many other features were proposed after SIFT, some provide improvements to this feature [CLY09; CM09; KS04], while others have different objectives such as extracting and matching the features faster [Bay+08; RD05; Rub+11; YC12]. The alternatives include PCA-SIFT [KS04], which includes a principal component analysis (PCA) to improve results, and HALF-SIFT, which improves localization of points. Finally, there are several surveys that compare robust feature detectors such as Mikolajczyk and Schmid [MS05], and Tuytelaars and Mikolajczyk [TM08].

The most common features, which improve performance against SIFT, are SURF, FAST and ORB. SURF [Bay+08] is able to find one image in a video in almost real-time with some scale invariance. FAST [RD05] is a feature that uses BRIEF descriptors [Cal+10] and was developed to work in real-time. ORB (Oriented BRIEF) is an improvement to the BRIEF descriptors adding additional reliability in rotated images. Although faster, both SURF and FAST are less accurate than SIFT. Both SIFT, SURF, FAST and ORB have an implementation in the popular computer vision framework OpenCV [Ope13]. These features will later be presented in detail in section 3.1.3.

Other features have different objectives besides tracking points. MSER, proposed by Matas et al. [Mat+04], is an affine invariant region detector which finds the Maximally Stable Extremal Region (MSER) between two images. It is mainly used in stereo vision applications.

Many applications can be created using the features presented in the previous section. The most common applications are image matching, as presented by Taylor et al. [TRD09], image tracking in real-time as in Wagner et al. [Wag+10a] and image photo-stitching, as proposed by Brown and Lowe [BL06]. These applications will be detailed in the next sections.

2.1.3 Photo Applications

There are several direct applications for computer vision and image processing techniques. Many applications use images as input. Images can be analyzed to detect specific features to be used in many applications. These include photo search applications [CLY09], photo-stitching [BL06], video manipulation [CLL06; DR08] or 3D reconstruction [Aga+11].

The image can be analyzed to detect special features, composite objects (faces, markers, polygons or other objects) or regions (segmentation, super-pixels or energy functions).

Several features were already discussed in the previous subsection. These can be points of interest as SIFT [Low04], stable regions such as MSER [Mat+04] or main lines [Hou62]. Using SIFT [Low04] or similar features, the main application is finding an image or image fragment in a large collection of unorganized photos. Brown and Lowe [BL06] proposed a system that automatically finds the spatial relation between a large set of unorganized images. Chu et al. [CLY09] propose a system to represent photo selection through the subset of features detected inside the selection. Photo Tourism [SSS06] is an example of an application that relates image features in large scale to create a 3D navigation system between them.

Object detection can include detecting faces, objects [VJ01] or markers [ART03] and is one of the main topics of computer vision with several applications in mixed and augmented reality (more in section 2.3). This is important for multiple tracking applications such as the surveillance application presented by Benfold [Ben11].

Another interesting way of looking into an image is through segmentation analysis. Images can be divided into several areas using energy functions that create aggregated super-pixel block regions [DB09]. A commonly used method is the Graph Cuts algorithm proposed by Boykov et al. [BVZ99]. Rother et al. [RK04] proposed an improved and interactive solution for the semi-automatic selection of an area in images.

There is a wide range of applications that can be achieved with image input. In this thesis the main focus will be given to applications for automatic discovery of

relation between images, automatic photo-stitching and panorama construction, 3D structure scanning and segmentation analysis.

The relation between images can be used to create a large panoramic image from the automatic photo-stitching of an unorganized set of photos, as explained by Brown and Lowe [BL06]. Automatic panorama [BL06; LHG08] construction requires a large set of images from the same scene. Liu et al. [LHG08] uses consecutive frames from videos to extract panoramic images. Wagner et al. [Wag+10b] presents a real-time mobile solution for this problem. The Unwrap Mosaics [RA+08] converts a video into a panoramic image. The image can then be altered with synthetic content. The results from that content are then re-rendered in the original video using image features to match the virtual content with the correct positions.

Some applications [DR08; RA+08] use the relations between images to create new video manipulation paradigms. Ay et al. [AZK08] use the cameras with orientation to create a geospatially navigable video stream. By analysing several images, sometimes the 3D relation between the cameras that took overlapping images can be extracted. This is the example of the Photo-tourism application [SSS06], which has the goal of modeling locations in 3D using large quantities of photos freely available on the Internet [SSS08]. This 3D structure is achieved through bundle adjustment of the features detected in each of the photos.

Segmentation analysis is important to isolate objects or large areas. This can be useful in selection techniques [RK04] or to recreate 2D scenes in 3D by isolating each object and inferring the object's depth as proposed by Saxena et al. [SSN09] in Make3D. Using an energy function the different areas can be summarized to achieve different goals. Pritch et al. [PKVP09] present a solution that reduces the size of an image while preserving the content meaning of the picture. Guo et al. [GDH11] present a shadow remover to create a shadow free image.

2.1.4 Camera and Video Applications

Numerous video based applications [CLL06; IAH95; Sze96] that use image features depend on some form of structure from motion [Cra+11; LK81] where the spatial relation of the features between frames are taken into consideration to acquire the structure of the scene. The difference between two images or frames is many times represented through a Homography matrix [LC09; Sze96]. This matrix contains the necessary components to distort one image into another. The matrix is usually inferred using the matching feature points of both images. To speed up the calculus many implementations use RANSAC (RANdom SAmple

Concensus) [FB81] to avoid processing all points at the same time. Many applications take advantage of scene homography for three-dimensional reconstruction [SSS08] or stereo vision [LC09].

Some of the more representative applications using cameras or video have already been presented in the previous subsection. There is a wide range of projects with different characteristics. Panorama construction from videos [CLL06; LHG08] is a common application because it is the most practical form of input for the user, with recent photographic cameras having an assisted panoramic mode based on video.

Cameras and video recognition are often used in real-time SLAM (Simultaneous Localization and Mapping) as shown in Gauglitz's et al. [Gau+12] work. With SLAM techniques and using videos from moving cameras it is possible to reconstruct in 3D a real world scenario [Rou+12]. Other possibility is to have a fixed camera and reconstruct the scene by changing the illumination source position [HS05].

Video summarization is also an important application allowing to quickly analyse a long video by eliminating repeated frames and merging non-concurrent areas of the video footage [CS08; RAPP06]. This is important mostly in still surveillance cameras [Ben11; Leo08]. Dragicevic et al. [DR08] present a video browsing technique in which the video motion is analyzed in order to make an interactive interface where certain objects in the video can be moved according to previous recordings of the video.

Camera recognition algorithms can also be used in conjunction with depth sensors (e.g., Microsoft Kinect camera[1]) to extract the scene structure or the human pose estimation [Sho+11].

2.2 Scene Acquisition

One of the main, topics which will be discussed in this thesis, is how to digitally acquire an unknown space using cameras (**RQ**, section 1.1). The previous section approached broad computer vision concepts and projects from several areas. This section presents specific projects and algorithms in scene acquisition using cameras divided by type of input: single image, several images and images with additional sensors.

As presented in the previous examples, the reconstruction of a scene using

[1]Microsoft Kinect camera,`http://www.microsoft.com/en-us/kinectforwindows/`.

images and videos is a recurrent research topic. Using several images PhotoSynth[2]/PhotoTourism can provide a 3D explorable point cloud environment. Using the same techniques several projects [Fur+09b; KCG11; SSS09] present different 3D visualizations systems for matching point clouds extracted from videos or images.

At this point it is important to separate the scene detection from the scene reconstruction. Many applications only require the acquisition of the 3D structure or layout [BSS11] of the scene (with more or less detail dependent on the application). Most of these applications use the detected 3D structure to superimpose virtual content [Del+11; Gup10; Gup+11].

In general, the scene detection techniques can be classified by the type of input that they use. It can be a single image (subsection 2.2.1), several images or video (subsection 2.2.2), or a conjunction of the first two with extra sensorial information (subsection 2.2.3).

2.2.1 Single Image Analysis

This subsection presents several approaches based on single image input. In the context of this thesis, this is especially important because the main proposed solution is itself based on a single image analysis, as stated in section 1.2. This proposed solution will be presented in detail in Chapter 4.

Using a single picture, as studied by Criminisi et al. [CRZ00] requires a thorough analysis of the image from different aspects. As previously stated, segmenting the scene into smaller problems is usually one of the solutions [GFK09; HEH07; RK04]. The Make 3D [SSN09] is a good example of 3D depth estimation from a single image using these techniques. Delong et al. [DB09] present a graph cut based technique to create a multiple level segmentation model. In the solution presented in Chapter 4, the Grabcut algorithm from Rother et al. [RK04] is used. This is an interesting example of an iterative segmentation technique which involves the user in the process.

Beyond segmentation some projects such as Liu et al. [LG10] use semantic information to access the content of the image.

Many projects focus on the analyses of the main lines that compose the scene [Gio+08; GJR07]. These main lines can be useful to extract the vanishing points [Li+12; Rot02] of the scene and to infer the main axis of the projection. From the main lines, vanishing points and super-blocks acquired from segmentation, several inferences can be made about the scene.

[2]PhotoSynth, 3D reconstruction from several photos, `http://photosynth.net`.

Lee et al. [LHK09; LGH10] have presented an example of structure detection based on indoor photos. It uses a technique in which the main lines of the scene are analyzed to discover vanishing points and test the line pattern against known models of indoor scenes.

Karsch et al. [KHF11] have successfully explored the visually correct detection of the scene structure from a single image producing very interesting results. For this, several assumptions are made to understand pictures of rooms, as stated in Forsyth's article [For13].

Using geometric analysis and pose estimation [SB02] of planar structures [Sim06] of a scene produces better results in Manhattan World [CY99; Fur+09a] environments where a series of perpendicular planes exist on the scene (e.g., floor, perpendicular walls and furniture).

Since this is a very important topic in this research, this related work will be revisited in the context of the presented solution in Chapter 4. The proposed solution will be compared with these solutions in section 4.5, with special attention to Lee et al. [LHK09] and Karsch et al. [KHF11].

2.2.2 Using Several Images and Video Analysis

Scene detection from cameras and video [Pol+07] or multiple images [DTM96; Ish+11; KCG11; PAR11; PKV98; WZ02] offers additional advantages because depth can be detected by triangulating the images.

Stereo vision or stereopsis [LC09] is a different approach where two video signals are used and processed at the same time. Its main advantage is that it is possible to calculate the depth of a perceived object or feature by comparing the two images.

Using stereo vision or several images it is possible to reconstruct in 3D an object [PH09] or scene. This is the basis for multiple projects [Fur+09a; GWCO09; HS05; Hig+09; Liu+11; MK10; SSS09; Zic+02], which have the goal of understanding the space relations of a scene taking into consideration several images.

The Mars Rover [NCP97] vision is one of the most known examples of SLAM (Simultaneous Localization And Mapping) using stereo vision to build a depth map. Many SLAM robotic applications use several cameras to improve location detection. One of the main challenges is to build the depth 3D map in real time. Gauglitz et al. [Gau+12] present an example of a working real time SLAM system.

As previously stated in subsection 2.1.4, 3D reconstruction can be achieved with moving video or a handheld camera [AKR11; PAR11]. An example of that is Roussos's et al. [Rou+12] work. Another technique is based on fixed cameras

and moving illumination. The 3D structure is calculated using the change in the reflectance of the textures [HS05; Hig+09]

Sometimes the video and the several images are used to detect only certain features and aspects. The PTAM project by Klein and Murray [KM07] detects planar surfaces using only a short camera translation. The PTAM system has been widely used in several other systems, such as Lovi et al. [LBC10] which integrates PTAM algorithms to search for free space in a scenario.

2.2.3 Extra Sensor Detection

In this thesis, the main goal is to explore a solution for generic cameras without additional hardware. This limits the scope of the addressed problem but still it is important to state that additional technologies can improve the results of the scene detection.

Single and multiple image analysis can be improved by using extra sensor data such as depth sensors (e.g., Kinect camera) [Iza+11], accelerometers [KB11], compass [CY99] or GPS [Pol+07]. This usually improves the result but makes the applications dependent on specific hardware.

2.2.4 Cluttering in Images

Beyond the detection of the scene there is also a large body of research about the study of cluttering in images [WG10; YM08]. The visual detection of 3D spaces and 3D elements is sometimes only possible if the scene has only a certain amount of clutter or if the detection algorithm has some form of ignoring the clutter. Hedau and Forsyth present several techniques to understand rooms [For13], recover the room layout [HH09] and recover the room free space [HHF12] in cluttered scenes.

2.3 Mixed and Augmented Reality

In the previous section several methods were presented to understand a scene captured from an image or video. This detection is important for new visualization systems, to present the scene from different angles [SSN09] but also to introduce virtual content combined with the real scene [KHF11] in mixed and augmented reality applications.

Mixed and Augmented Reality systems have been around for many years [Azu97; Mil+94; Val98]. The first experimental prototypes date back from the

1970s but they lacked the graphical processing power to effectively implement the concept. In the last decade, with the evolution of hardware, several commercial applications emerged in game consoles (e.g., EyePet[3]), design (e.g., Atelier Pfister[4]), and GPS navigation (e.g., Layar[5], Junaio[6]). Augmented Reality (AR) has imposed itself mainly in non-critical environments but there is still a large discussion about its industrial applications [FG11]. An increased attention to AR has been given due to a boost in users with access to a smartphone or tablet with back and frontal camera. Several applications take advantage of this setup [MSS12a; Wag+10a]. There are currently several frameworks (e.g., ARToolKit [ART03], Studiertube Tracker [Stu11], Vuforia [Vuf13], Metaio [Met13], OpenCV [Ope13]), which enable the development of mixed and augmented reality applications using different techniques.

Every augmented reality system has to essentially solve two problems: registration and superimposition. In the registration phase it is necessary to acquire the location where the virtual information will be placed. Typically GPS/compass or markers are used. After the registration, the virtual information has to be rendered correctly in 2D or 3D (using pose estimation in 3D) in the correct spot.

2.3.1 Sensor and GPS Based

The simplest AR systems use the GPS position and the compass present in most mobile devices to track the location where the information will be rendered (e.g., ANTS [Ra+04]). The virtual information is usually displayed using the "bubble metaphor" [TP12], where the size of the bubble depends on relevance and the distance of the information to the user. Examples of this type of interaction are the AR platforms Layar and Junaio.

2.3.2 Marker Based

Another approach widely implemented is using fiducial markers (e.g., ARToolKit [ART03], Studiertube Tracker [Stu11]) [MSS12a] or QR-codes [TSS11] to locate where the virtual information should be displayed in the image.

Although these systems require more processing power than the previous systems it is usually very simple to detect the markers. They usually have very

[3]Eye Pet, Sony Playstation camera game,`http://www.eyepet.com/`

[4]Atelier Pfister, smartphone application for furniture design, `http://www.atelierpfister.ch/app`.

[5]Layar, GPS augmented reality application, `http://www.layar.com/`.

[6]Junaio, GPS augmented reality application, `http://www.junaio.com/`.

distinctive characteristics such as: binary color system, simple geometric form (e.g., quadrilaterals, circles) and known physical size. The main disadvantage of marker systems is the need to introduce a marker in the captured scene, meaning that the user has to have/print a marker and necessarily has to have access to the scene area. Alternatively, the user sees the marker in the world and has to get the specific application to see it augmented. This is one of the main points that will be discussed in this thesis with a solution to avoid these specific problems (presented later in Chapter 4).

Markers are ideal for augmented reality because they are simple to track, they can be used to setup a pose estimation algorithm to insert 3D content and the physical size of the marker can be linked to the scale of the virtual object. There are inumerous examples of projects using ARToolKit [ART03] markers (e.g., [MSS12b; SFE12]).

Most recently feature based systems have replaced ARToolKit markers by image markers [TM11; Uch11]. Mulloni et al. [MSS12a] present a system which use several images as ubiquitous markers which help in an indoor navigation system. Metaio [Met13] and Vuforia [Vuf13] are two software libraries for smartphone development, which have successfully commercially explored the concept of image markers for augmented reality. An interesting example is the Ikea 2014 catalogue[7] smartphone application using AR furniture models which can be introduced in the scene by using the cover of the catalogue as an image marker.

2.3.3 Feature Based

The most recent developments in AR are systems that do not dependent so much on markers and use instead scene analysis and visual element detection. The image markers presented in the previous subsection are one of the forms in which feature based detection can be used to introduce mixed and augmented reality.

Using several techniques described in the previous sections, namely SIFT, SURF or FAST image features or 3D point clouds, several applications [WSB09; Wag+10a] have been developed to track features in real-time and in smartphones. These applications track different elements instead of visible markers such as planar spaces [Sim06; SFZ00] or pre-programmed images [Ngu+12; TM11].

The PTAM [KM07] project automatically detects a plane in the scenario. This is achieved by translating the camera sideways. Using the structure from motion, the 3D scenario is acquired. Using the detected plane, virtual applications

[7] Ikea 2014 catalogue, `http://www.ikea.com/ms/en_AA/customer_service/catalogue/catalogue_2014.html`.

can take place on that plane, in the real world. The PointCloud[8] is a very recent framework, which improves the PTAM concept by providing a library to create augmented reality applications in a user real space. Advances in 3D reconstruction using handheld cameras are beginning to allow the introduction of virtual objects that interact with the 3D detected scene [Vuf13]. Point cloud detection and tracking has been used for architectural presentations such as the preview of Barcelona's cathedral Sagrada Familia [GPM10].

Other alternatives, based on feature detection are presented by Gupta et al. [Gup+11], where the scene layout is detected from a single image and free space is studied for the introduction of human models. Uchiyama et al. [UTM12] present an AR example based on countour detection while Bunnun et al. [Bun+12] present an AR system based on automatic 3D object detection and tracking on mobile phones. Liu et al. [Liu+08] propose an AR system for videos which is based on the analysis of scene transitions. Several studies and applications are being conducted to augment paper publications such as magazines [Ngu+12] or catalogues (e.g., the Ikea catalogue 2014).

2.3.4 Other AR Applications

There are many different types of AR applications. Some [TP12] augment the reality by deforming it and giving more screen space to important objects such as important sites and monuments. Others use special hardware such as head-mounted displays [NBK12; OF12] (e.g., Google Glass[9]) or accelerometers to detect the floor [KB11]. Some are specific for games [CKD11] or newspapers AR applications [Ngu+12].

2.4 Interfaces and Interaction

The main focus of this section is study the interface required to acquire the features, objects, 3D model or any element of the scene with the goal of introducing virtual content later. This research is essential to obtain answers for the research question 3 (**RQ3**, in section 1.1) about how can mixed reality systems be initialized by the user.

In this section, several applications that inspired the proposed solution (later explained in Chapter 4) are presented. At this point, most of this related work

[8]Pointcloud, 3D SLAM library for smartphones from 13thLab,`http://pointcloud.io`.

[9]Google Glass, Interactive glasses that can be used for AR,`http://www.google.com/glass/`.

has already been presented in the previous sections as examples.

Taking into consideration the different types of mixed and augmented reality applications presented in the previous section the type of registration influences the interfaces.

In GPS based interfaces (e.g., ANTS [Ra+04]) the main problem of the interface is the need to manage the real estate of the screen. The "bubble metaphor", described by Takeuchi et al. [TP12], means that several labels located at a certain geo-referenced place will be fighting for attention in the screen thus creating a confusing interface for the user. Grasset et al. [Gra+12] propose a solution, which simplifies the visual clutter of the labels by positioning them in more visible places according to the content of the image.

Marker based AR application need to find strategies to introduce markers in the environment. CAD applications [ML11; OS05] for furniture have used AR markers to simulate virtual objects inside real rooms [Del+11]. An example of that is DesignMyRoom[10]. In this application a marker is placed in the center of the room to extract the 3D scale and orientation of the room's floor. The user is instructed to take a picture of the room and that picture with a marker initializes the 3D environment inside the application. An interesting alternative for less expert users is the already mentioned Ikea Catalogue 2014, which uses the cover of the paper catalogue as an image marker. The main advantage of this solution is that it automatically provides a marker to the user, which is already available.

In this thesis, one of the main goals (**RQ3**) is to simplify the initialization process by removing the need for a specific marker in the scene. To do this, feature based systems need to be used (subsection 2.3.3) for detection. In order to detect the correct features many application rely on simple interface instructions to acquire the scenario.

In automatic panoramic construction applications [Wag+10b] the user is usually required to move the camera using only rotation or only translation in order for the distortion to be minimal. It is important also that the user moves the camera slowly to avoid motion blur and for the tracking to work.

The previously mentioned PTAM [KM07] project presents a very interesting interaction concept regarding the acquisition of the scene using a camera. Using it, virtual applications can interact with the surrounding environment taking into consideration obstacles and open spaces. The interaction consists in waving the camera sideways, with this, the application has a rudimentary notion of the real

[10]DesignMyRoom, Interior design application, `http://designmyroom.com`

world space and detects the largest available plane in the scene. The Ball Invasion[11] game is based on a similar system. It has a short tutorial with instructions and animations to illustrate how should the main plane of the game be acquired.

The previously mentioned furniture design application Atelier Pfister (section 2.3) is an accelerometer based mixed reality application. The user is instructed to take a picture of a room where he/she wants to lay virtual furniture. The orientation of the floor is detected using the accelerometer of the mobile device. To detect the scale of the scene, a virtual person is introduced in the scene, and the user has to scale it up or down until the scale of the virtual human feels natural.

Additional analysis of the scene main lines is important to add other functionalities such as snap to line and automatic alignment of objects [Del+11; FSB09]. Karsh et al. [KHF11] detect the scene structure from a single image by using a human assisted method where the user is constantly asked to refine the detection by annotating geometry and lights.

Exploring the properties and image features in videos can lead to interesting interface results in video editing taking into consideration the motion flow of objects [DR08], games using real footage [Lai+11], or video summarization [RAPP06]. You et al. [YHC12] also present several interface concepts for a better selection of interactive objects in augmented reality.

2.5 Summary

This chapter categorizes and summarizes the state-of-the-art of several topics that will be presented throughout this document. Additional related work will be presented and contextually revisited on each chapter whenever appropriate. The next chapter will present some of the techniques and algorithms already mentioned in section 2.1, but with additional detail and implemented prototypes.

In this chapter, the related work was approached from four perspectives giving emphasis to computer vision (section 2.1), different forms of scene acquisition with cameras (section 2.2), mixed and augmented reality paradigms (section 2.3) and interactive interfaces to support the previous systems (section 2.4).

[11] Ball Invasion, Smartphone AR game, `http://13thlab.com/ballinvasion/`.

3

Algorithms and Researched Techniques

Building applications using images and videos requires a thorough interpretation of such media elements. At a very basic level, raster images or video frames are composed of a two-dimensional grid of pixels with different color channels. These pixels compose altogether different forms, shapes, lines, corners and areas, which define in turn, different objects that constitute a scene. Applications that use computer vision or image processing often rely on the automatic detection of visual elements to build an interaction apparatus. For this it is vital to detect the aforementioned elements automatically with high reliability without requiring much information besides the image, such as meta-data or manual annotations. The goal in machine vision applications is to minimize the introduction of data by using only images and video.

As mentioned in section 2.1, images can be described by a collection of special interest points that make the image almost unique. To describe these points several features are available such as SIFT, SURF or FAST (section 3.1). Using these features several small portions of an image can be quickly identified in another image. With this it is possible to build applications that find a certain image marker in another image or video (section 3.4), combine several consecutive frames in a larger panoramic picture or find the spatial homographic relation between two images. In the beginning of this chapter, in sections 3.1 and 3.2, several of these techniques are explained along with examples from implemented

prototypes.

Analyzing the points of interest of the image is very important to detect common regions in two pictures. By using several images from a certain scene it is possible to triangulate these points and generate 3D information. Section 3.5 presents a stereo vision prototype, which takes advantage of this technique.

For a better understanding of the structure that composes a single image, it is common to analyze the main lines and main pixel regions that constitute the scene. By detecting the main straight lines it is possible to converge the projection of parallel lines, find vanishing points, understand the scene orientation or find squares or different specific polygons. By combining these detected elements it is possible to calibrate cameras to superimpose virtual objects in three dimensions in accordance with the scene.

In this chapter, several techniques are presented to retrieve and classify information from images. These techniques are very important in augmented and mixed reality applications. Bearing this in mind, there is a large quantity of other methods and elements that could be explored [Sze10] but the main focus in this work is on technologies that are useful to combine real world images with virtual content. Some of these techniques were already briefly approached in Chapter 2, but are here described with additional detail because of their importance for the presented work and to properly document all the essential algorithms.

3.1 Image Keypoints and Features

One of the most common problems in computer vision is matching keypoints between images, as seen in Figure 3.1. The detection of keypoints that can be matched between images is essential. Simple corner detectors (next section), can be used for the detection of these keypoints. To be comparable across images, the keypoints have to be described according to certain features, such as SIFT, FAST or SURF. (sections 3.1.2 and 3.1.3).

3.1.1 Harris Corner Detection

The study of images has improved greatly with the introduction of descriptors that can summarize several features of an image. Most of these descriptors describe special interest points that represent local maximum peaks in the image such as corners. One of the most significant corner detection algorithms is the Harris corner detector [HS88]. The main goal of the detector is to find local maximum values of a score function that evaluates each pixel. The result should be

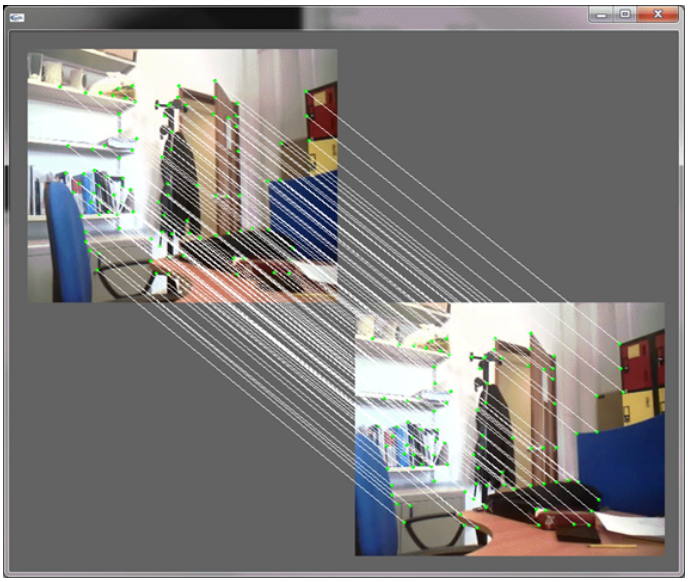


Figure 3.1: Image matching is one of the most common problems in computer vision. Prototype screenshot where the keypoints of the image in the left were matched with the keypoints of a slightly different image, on the right.

a matrix representing all pixels with very high values in the corners, as seen in Figure 3.2, where corners are represented in white.

To detect if a certain pixel (assumed to be at the origin) is a corner, the Harris corner detector analyzes the neighbour pixels around a certain squared window. A corner is a point on the image where the derivative changes in almost all directions. For every neighbour pixel an E function is calculated,

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2 \tag{3.1}$$

and every (x,y) pixel in the neighbour window, is compared against each pixel in the original window, where:

- $E(u,v)$ is the difference between the current window and the window moved by a small amount and centered at (u,v).
- (u,v) are the displacements of the window in the x and y direction. Common used values for u and v are $(-1,0,1)$.
- $w(x,y)$ is a weighting factor defined by a Gaussian function with parameter σ and origin on the center of the window, which means that closer pixels will have a larger impact factor than farther away pixels.

$$w(x,y) = exp(-\frac{(x^2+y^2)}{2\sigma^2}) \tag{3.2}$$

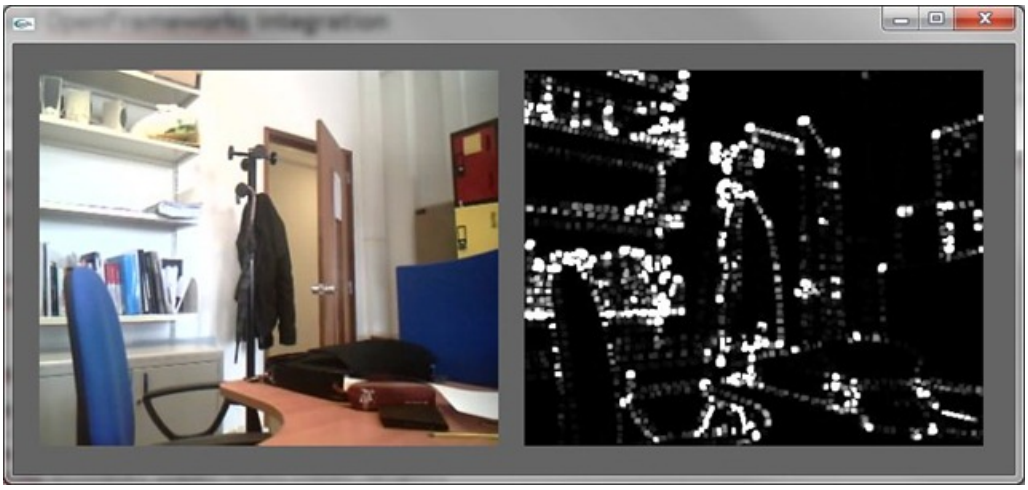


Figure 3.2: Corner detection using the Shi-Tomasi [ST94] corner detector, an evolution of the Harris corner detector.

- I is the image intensity at a certain pixel.

If $min(E(u,v))$ (for all u and v) is above a certain threshold then there is a high probability of having a corner and we should look for a local maxima in the original window ($max(I'(x,y))$). To show how the algorithm works, Figure 3.3 presents the difference between a uniform area and a corner area.

The Harris detector is actually an evolution of the Moravec detector [Mor80], with optimizations, which take advantage of several properties of the eigenvalues of a matrix. As explained in the Harris algorithm [HS88] using a Taylor series expansion and derivatives functions I_x and I_y, we have:

$$E(u,v) \approx \sum_{x,y} w(x,y) \left[I(x,y) + uI_x + uI_y - I(x,y)\right]^2 \tag{3.3}$$

by expanding the square:

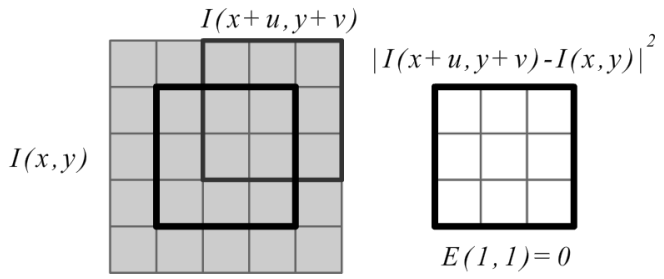
$$E(u,v) \approx \sum_{x,y} w(x,y) \left[u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2\right] \tag{3.4}$$

and putting in matrix form:

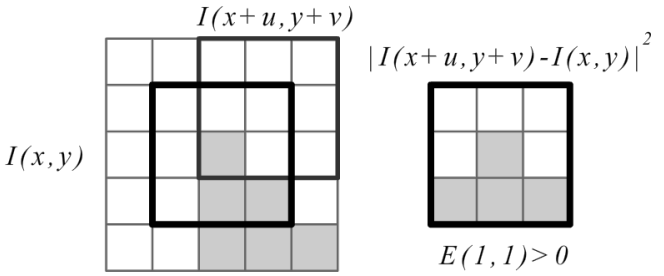
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{3.5}$$

using M, we have:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{3.6}$$



(a) Uniform area.



(b) Corner area.

Figure 3.3: These two figures represent the analysis of two areas using a 3×3 window around the center pixel. Figure 3.3(a) represents a uniform area and Figure 3.3(b) represents an area with a corner. In this example one of the neighbours $E(1,1)$ is tested. If all tested windows in all directions produce a large E above a certain threshold then we can say that there is a corner.

Using the properties of eigenvalues λ_1 and λ_2 of M, Harris and Stephens propose a shorter representation using an R score:

$$R = detM - k(trace(M))^2 \tag{3.7}$$

where k is a parametrization constant with recommended values in the interval $[0.1, 0.15]$:

$$detM = \lambda_1\lambda_2 = I_x^2 I_y^2 - (I_x I_y)^2 \tag{3.8}$$

$$traceM = \lambda_1 + \lambda_2 = I_x^2 + I_y^2 \tag{3.9}$$

The higher is R the higher is the probability of a corner. This optimization means that to find all the corners of an image the computation can be simplified by finding for every pixel I_x, I_y and $I_x I_y$, computing R and applying a threshold. Several improvements have been made to this implementation by Shi and Tomasi [ST94] to improve stability under tracking algorithms. Figure 3.2 presents a corner detection implementation in OpenCV using the Shi-Tomasi detector.

3.1.2 SIFT: Scale Invariant Feature Transform

Matching the detected points (e.g., corners) across images, requires the use of comparable features. One of the most notorious image features are SIFT [Low04]. Using this technique it is possible to describe a single image saving only the most significant points of the image and their SIFT descriptors. One of the main advantages of the SIFT features is the fact that they are invariant to scale, rotation, illumination and viewpoint. This means matches can be found between images representing the same scene from different points of view. According to Lowe [Low04] there are four main stages in SIFT algorithm extraction algorithm: (1) Scale-space construction and extrema detection; (2) Keypoint localization; (3) Orientation detection on each keypoint; and (4) Generating keypoint descriptors.

Scale-space construction and extrema detection: The first step is to construct a scale-space. The goal is to obtain the same image in different scales, each scale is half of the previous. For each scale the image is progressively blurred using the Gaussian blur operator, creating several versions of the image with different blur levels. Each set of blurred images is called an octave. The number of octaves (scales) and blur levels depends on the implementation. In the end there will be a multi-level set of images with different scales and blur levels. This is done because certain details only make sense at certain image resolutions. The different levels of blur clear the noise in the scene leaving only the main forms of an object.

After building the scale-spaces, the results of each Gaussian image are subtracted with the next image in the octave, using the difference of Gaussian (DoG) seen in Figure 3.4. This locates the main edges of these images. According to Lowe the DoG images provide a close approximation to the scale-normalized Laplacian of Gaussian, meaning that they are directly comparable between images of different scale.

Keypoint localization: From the DoG images several keypoints are generated from local maxima/minima. This is done by analysing each DoG image using a similar approach as the Harris corner detection explained in the previous subsection (subsection 3.1.1). The main difference is that each pixel is compared in the neighbours from the current image but also against the neighbours in the image below and above. Each detected keypoint is associated with a certain scale and Gaussian level.

Orientation detection on each keypoint: The orientation of each keypoint is required to make the keypoints invariant to rotations. To do this the idea is to obtain the gradient of every direction around the point and create an histogram of the magnitudes of those gradients. The histogram divides the 360° in several

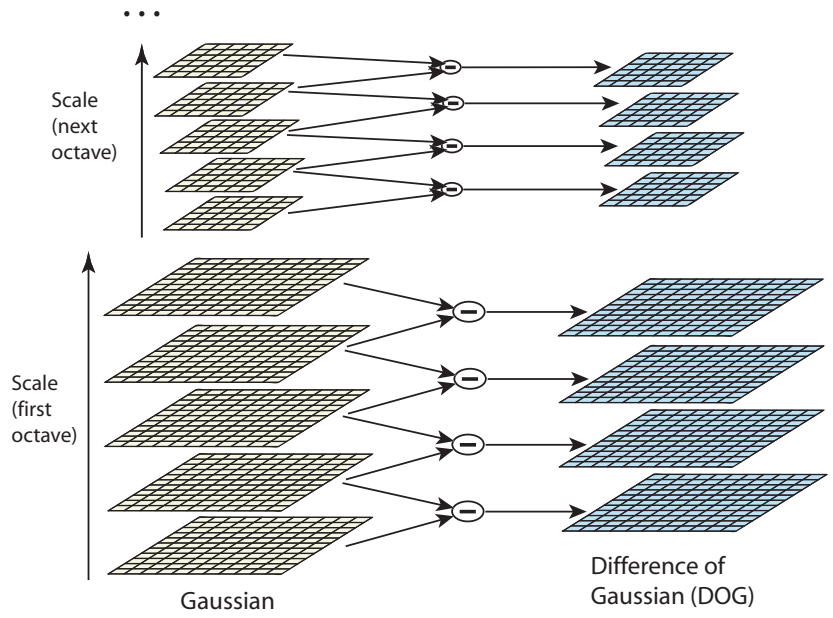


Figure 3.4: The scale-space where each image is repeatedly convolved with Gaussians to produce the next image. The difference of Gaussians is presented on the right.

(Source: David Lowe 2004 [Low04])

bins. As an example, if there are 8 bins, each bin covers $45°$. For each bin the magnitude of the gradient is calculated for those angles. The idea is to measure how large is the variance of the image in a certain direction. Using the histogram the angles where there is a larger magnitude are identified and assigned to the keypoint. If there are several peaks in the histogram, new keypoints are created in the same position but with different orientation.

Generating keypoint descriptors: After obtaining the oriented keypoint, the goal is to obtain a unique descriptor which can be comparable with another keypoint from another image. Since obtaining keypoints that match exactly is extremely difficult, the algorithm studies the area near the keypoint.

The descriptor consists of a window around the keypoint, as seen in Figure 3.5. This window has the same scale and rotation as the keypoint. The window is broken down in smaller 4×4 windows. For each small window the gradient is calculated in 8 different bins, representing 8 magnitude direction values. The values in the larger window are weighted using a Gaussian function to make the farther gradients count less then the closest ones.

In Figure 3.5, the keypoint descriptor has 4 small windows, each window has 8 values thus the final descriptor will have $4 \times 8 = 32$ values. Most common implementations of the SIFT descriptor, such as OpenCV [Ope13], use a 16×16

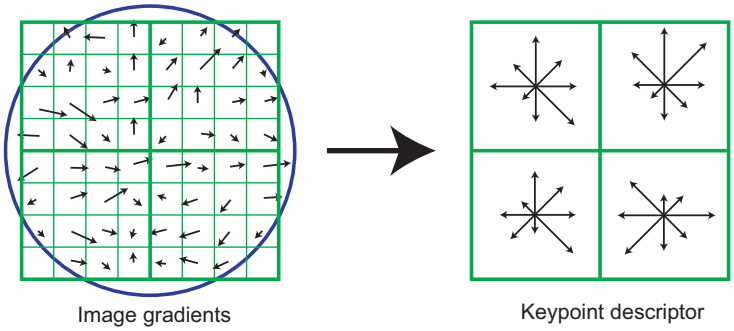


Figure 3.5: SIFT descriptor. Each 4×4 window is described by 8 gradients.
(Source: David Lowe 2004 [Low04])

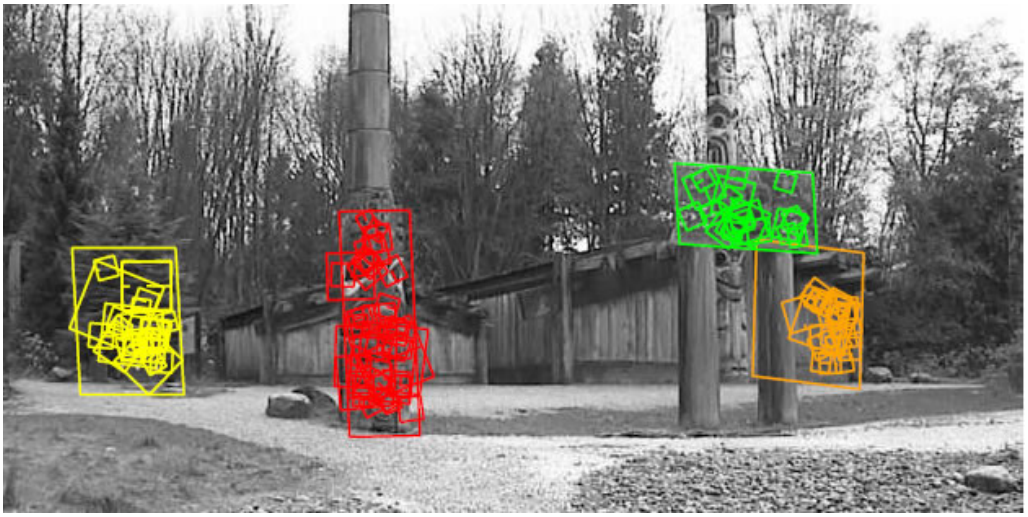


Figure 3.6: Finding 4 images inside the larger image. Each large rectangle contains a representation of the squared descriptors that were matched.
(Source: David Lowe 2004 [Low04])

window resulting in a descriptor with 128 values for each keypoint. This is called the feature vector and it is completely invariant to scale and rotation.

The final result (seen in Figure 3.6) is a set of oriented keypoints with a descriptor associated to each. The values of the descriptors can be directly compared between each other. A certain keypoint exists on another image, if there is a descriptor from that image that is sufficiently similar from the descriptor of that keypoint.

3.1.3 FAST and SURF features

The SIFT feature, described in the previous subsection, is a very robust feature, largely invariant to scale and rotation and used in many applications. Other features have been proposed later [CLY09; CM09; TRD09] with different objectives

but usually the goal is to extract and match the features faster. One of the most used features is SURF, a scale invariant feature [Bay+08], which is able to find one image in a video in almost real-time.

Another important feature is FAST [RD05] a feature that uses BRIEF (Binary Robust Independent Elementary Features) descriptors and was developed to work in real-time, although with small guaranties of invariance of scale or rotation.

There are several alternatives with improvements on the previously mentioned features. ORB (Oriented Brief) [Rub+11] presents a rotation invariant version of the FAST+BRIEF implementation.

Although faster, both SURF and FAST are less accurate than SIFT. Both SIFT, SURF, ORB and FAST have an implementation in the popular computer vision framework OpenCV [Ope13].

SURF: Speeded Up Robust Features: The main concept of SURF is to apply different Laplacian Gaussian filters to the same image and find the local maxima that is common across filters. To detect the features, the SURF algorithm detects the Hessian matrix H for each pixel.

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \tag{3.10}$$

The determinant of this matrix gives the strength of the local variation of the pixels in one direction. To calculate the second order derivatives, this matrix is computed using Laplacian Gaussian kernel of different scale σ (equation 3.2 from the Harris corner detection, subsection 3.1.1). Every local maxima will then have three variables $H(x, y, \sigma)$. To optimize the computational cost, the Laplacian are approximated by Gaussian kernels. Using convolution the kernel matrices can calculate the derivatives with small cost. The typical SURF implementation uses 12 diferent kernel sizes, going from 9×9 up to 99×99. In their work, Bay et al. [Bay+08] present several kernel alternatives.

Since there are different σ Gaussian kernel levels being calculated the SURF features are invariant to scale. They are computationally much faster to extract than SIFT but are less strong against rotation.The final result is a set of keypoints with a scale σ associated.

FAST: Features from Accelerated Segment Test: The FAST features [RD05] revolve around the definition of what constitutes a "corner". A keypoint is declared if $3/4$ of the pixels in a virtual circle around the center are different from the center itself, as seen in Figure 3.7.

This is a very quick test to do, and in their work Rosten et al. [RD05] present

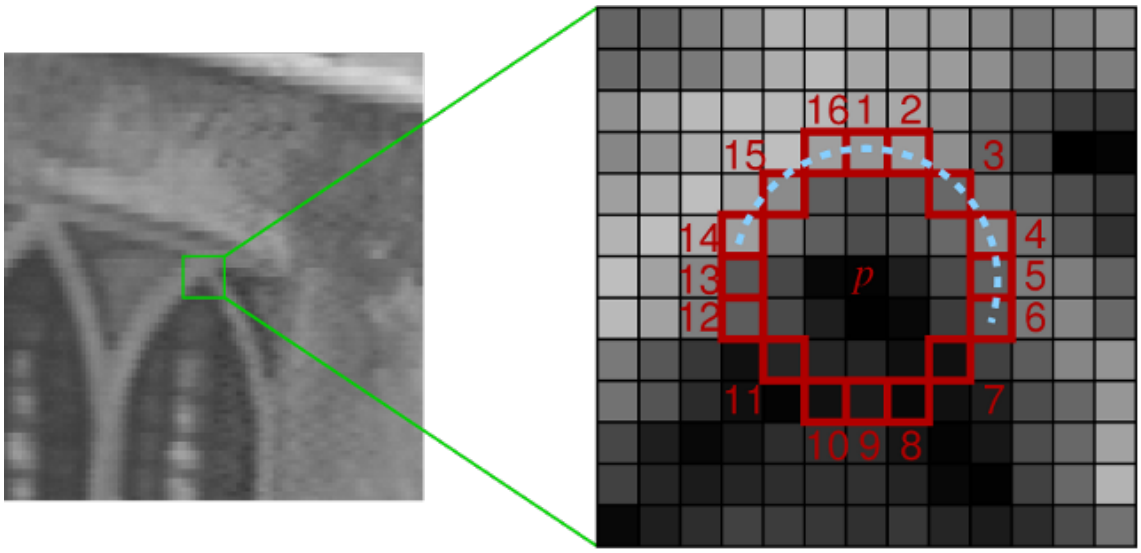


Figure 3.7: The FAST keypoints are extracted by analysing 16 points around the center. $3/4$ of the points must be sufficiently different.
(Source: `http://www.edwardrosten.com/work/fast.html`, [RD05])

several additional minor optimizations which speed up even more the process. Typical FAST implementations use a radius of 3 pixels around the point and use BRIEF descriptors [Cal+10].

3.2 Homography Calculus

As described in the previous section, images can be defined by a certain number of points. Each relevant point contains a descriptor (e.g., SIFT [Low04], subsection 3.1.2) describing the area around the point. To search for an image inside another, the descriptors from one image are matched against the descriptors from the other. If the matching degree is above a certain threshold and the position of the points is geometrically correct a match is signaled. The similarity between two images is described by the homography matrix (equation 3.11). This matrix defines an affine transformation, which distorts one image, so that the points in one figure match the points in the original figure with the smallest distortion error possible.

The detected match can be visualized by superimposing the rectangle of the first image with the distorted quadrilateral figure of the second image (Figure 3.8).

Before matching two images, the keypoints and their descriptors of each image must be acquired using one of the features described in the previous section.

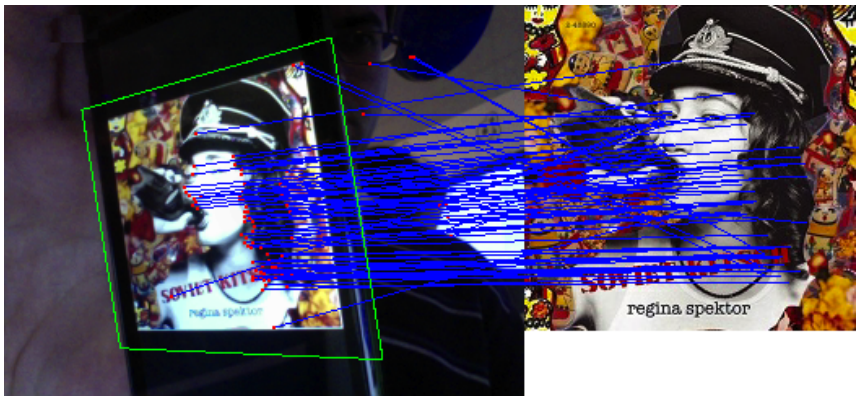


Figure 3.8: Image detection with superimposition of a green square using the homography matrix.

After this, the descriptors of one image are tested against the descriptors of the other image to find matching points. To avoid testing every point against every point from the other image, a nearest neighbour approach is often used for matching. A common algorithm is FLANN (Fast Library for Approximate Nearest Neighbors) [ML09].

At this stage, there is a list of points identified in two images as seen in the blue lines in Figure 3.8. The next step is to find if the matched points are spatially coherent and not just randomly matching in space. Estimating the homography matrix defined by these points verifies the spatial coherence. If it is not possible to estimate the homography, it means that the points are too scattered around and it is not a valid match.

The 3×3 homography matrix H, defines the affine transformation between the two images and is defined by the following equations:

$$H = \begin{bmatrix} u_x & v_x & t_x \\ u_y & v_y & t_y \\ u_w & v_w & 1 \end{bmatrix} \tag{3.11}$$

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \approx \lambda H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3.12}$$

The vectors u and v define the new X and Y axis providing skew and rotation while the vector t is the translation of the image. The distorted point (x'_i, y'_i) is obtained by multiplying the original point (x_i, y_i) by the H matrix.

The H matrix is estimated using RANSAC (Random Sample Consensus) [FB81] as seen in Figure 3.9. RANSAC is a fast method that avoids testing all points

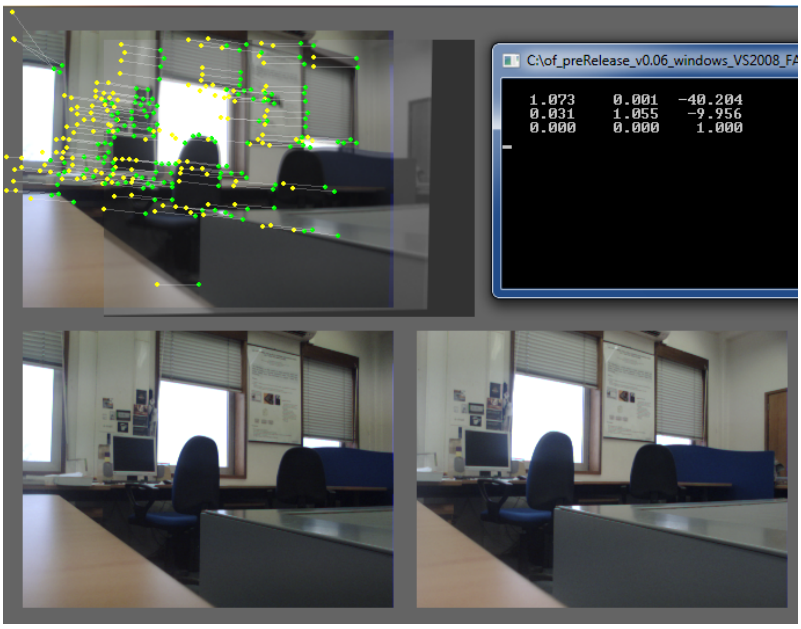


Figure 3.9: Homography example. Two almost equal images where on the top left, the second image is distorted and superimposed on the first. The console presents the calculated homography matrix.

against each other, instead, it only tests randomly selected pair samples of points from the two images.

RANSAC tries several subsets of the available points in order to discover the transformation matrix, which minimizes the error of the Homography matrix. This error is the sum of all the differences between the original image points and the distorted second image points. Several geometric constrains, such as checking for minimum area or significant determinant value, can be used to verify if the matrix was correctly detected.

With this matrix the second image is transformed and blended into the first as seen in Figures 3.9 and 3.10. In Figure 3.10, the main focus of the implemented prototype is the positioning of the photos above each other. This was a technique that was used in one of the applications with the goal of finding all the homography relations in a set of related images. This application is presented in detail in section 5.3.

Besides superimposing another image on top of the original, applications can superimpose virtual content in 2D with the correct distortion. This works similarly as the markers in ARToolKit [ART03], but this time the marker is an image (Vuforia [Vuf13] and Metaio [Met13]). A prototype using this technique will be presented in section 3.4. Pose estimation needs to be calculated to overlay 3D objects on top of the image [ART03; Vuf13].

Figure 3.10: Proof-of-concept prototype: with the homography between images it is possible to graphically arrange the pictures spatially; in this example two pictures are spatially aligned.

3.3 Structure from Motion

Using a composition of several images it is possible to build a panoramic image of a certain scene, as seen in Figures 3.11 and 3.12. This type of quadrilateral distorted images is also known as mosaics.

Using image mosaics it is possible to visually document a larger area otherwise impossible due to limited field-of-view. But what if the system needs to infer the depth of the scene? Szeliski [Sze96] formalizes the problem of constructing panoramic images by presenting the mosaic solution and extending it to a three-dimensional space. This project evolved to become the Photo Tourism application [SSS08] where a large set of images from a certain place is collected from the Internet. Additionally, it also calculates the camera source of all images by comparing

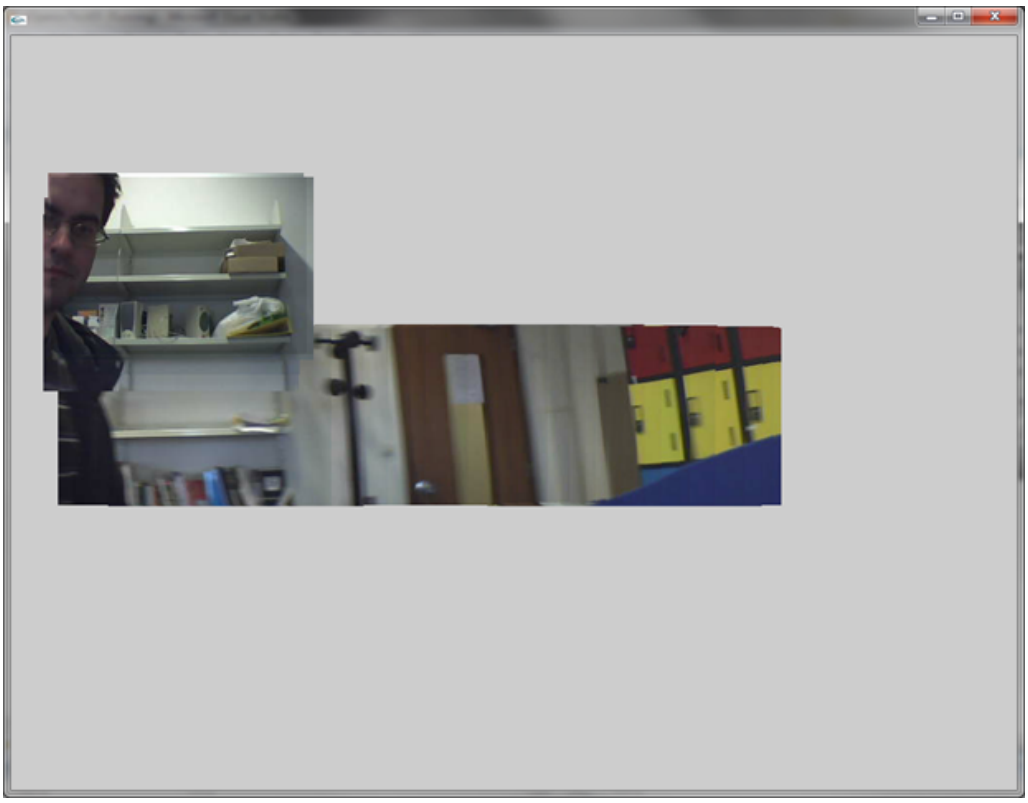


Figure 3.11: Optical Flow, real-time panorama construction by using translation of a camera.

Figure 3.12: Prototype for fast creation (∼5 fps) of panoramic scene from a continuous video sequence using flow detection to calculate the homography and the respective rotation of the images.

all pictures against each other. This way it displays all images in 3D space identifying important points and creating a 3D point cloud of the scene. From these point clouds it is possible to reconstruct the 3D scene that was photographed [PH09].

Considering the case of a free moving camera taking photos and videos sequentially of a given scene, continuous video sequences have the advantage that each frame is spatially related with the previous frame [LHG08]. Depending on the moving speed of the camera and the frame rate it is highly probable that neighbour frames share a large portion of the scene. At this stage, the problem is to find the portions of the images that overlap or, in other words, for each image pair, given a set of points of the first image, to find them in the second image. There are essentially two common methods to do this, the first is to use optical flow operations [CLY09; LK81] and the second is to match features, such as SIFT or SURF (section 3.1).

The optical flow method is best suited for video sequences and depends on the relative motion of the camera. The method starts by identifying important points in one frame such as corners, edge points or contrasting characteristics. This is done using methods such as Harris corner detection (subsection 3.1.1) or others [CLY09]. The next step is to find in the next frame the same points by analyzing the flow of the motion. The prototype in Figure Figure 3.11 uses the Lucas-Kanade algorithm [LK81] to detect the flow. This algorithm scans a portion of the second image, trying to find the pixel that minimizes a given color/intensity distance function.

The other method to detect matching points is using image features (e.g., SIFT [BL06]). This method has the advantage of working with non-sequential images. The input can have any orientation, since the SIFT features are invariant under rotation. SIFT is usually more accurate, although optical flow methods can be very fast in videos. In the prototype of Figure 3.12, it is possible to run a mosaic algorithm with a camera in real-time at around 5 fps in an average computer and still have around 90% matches.

In the end, for each sequential image there will be a rotation matrix R, relative to the previous image. Panoramic images usually involve a later process of correcting lenses distortion, cleaning edges and blending in order to obtain a better photo quality.

3.4 Finding Scene Elements in Video

Using the previously image matching system (section 3.2), another problem that needs to be addressed is finding a certain image in a large set of images. Finding a known visual object or image in a large video is a classical problem with applications in video editing, summarization and surveillance [RAPP06]. When a video is being manipulated, it is sometimes useful to find all the visual appearances of an object, building or scene element. Additionally, besides the detection of these occurrences, the goal is to add graphical content to the detected objects. This content can include virtual text labels, blurred squares to hide unwanted content, and augmented reality objects for warning detection in surveillance.

In the prototype [NC12c], presented in Figures 3.13 and 3.14, a method for summarization of the video content was tested to discover the frames that contain a certain scene element. The scene element is an input of the application and is provided by the user by selecting an area of a certain frame. The detected frames should be processed in order to render a virtual object on the scene element following its position and orientation throughout the video.

The final goal was to enable two basic video editing operations: (1) finding all frames with a certain scene element and (2) re-render the video adding a given virtual object to the scene element. With small changes the second operation can even be implemented using live cameras.

The purpose of the proposed technique was to summarize a movie and find all the frames containing a given scene element. The scene element is chosen by the user and it can be any area in any specific frame of the video (Figure 3.13(a)). The area should be larger than 10 by 10 pixels with sufficient detail to be detected. The video can be of any size and feature all sorts of camera movements. In the performed experiments the camera has been intentionally rolled and tilted in several directions. Blurred frames should be avoided as they interfere with the detection. When loaded, video frames are processed to find the main points that best describe the image. Currently, SURF features are being used but the model is still valid for FAST and SIFT features. The calculus of the descriptor can be done initially when the video is loaded or, on the fly, frame-by-frame, while the matching is being done. The first option can be interesting if several matches are being made and if the movie is relatively small. The second is better for long movies.

In the current implementation, the SURF descriptors are calculated in the beginning and saved in a file to speedup later searches. Taking in consideration that matching the entire video for an element takes about a third of the time that

(a)

(b)

Figure 3.13: Prototype snapshots [NC12c]. Panoramic video of a campus taken from the roof of a building. The camera was tilted and rolled to make detection more difficult. In (a) a building is selected. In (b) the SURF descriptors of the selection are extracted.

it takes to analyze and find the descriptors, doing it initially saves time if several queries are made. The problem with this solution is the memory space it needs. The solution to this problem will be calculating the descriptor as needed using parallel processing.

The matching is a process of comparing the descriptors and accepting them as equal as described in section 3.2. Initially the scene element selected by the user from the video is processed to find its descriptors. With the descriptors found, the next step is to compare them with each frame of the video to see if the element is present on that frame. Sequentially comparing all frames can be a very expensive operation. Since the element may not be present in large portions of the video, it

(a) (b)

Figure 3.14: Prototype snapshots [NC12c]. Bottom video summarizes matching frames. (a) and (b) are two different frames with superimposed dragon. The dragon follows the rotation of the camera.

is best to do a first search where only certain frames separated with an interval n are scanned. A second search is performed in the intervals near frames where a positive match was signaled. On each frame to have achieved a match, the element descriptors need to be compared against the frame descriptors. This n to m comparison is implemented through the FLANN [ML09].

Figures 3.13(a) and 3.13(b) are screenshots of the prototype application. In the first image the user can view the movie, pause it at any frame and select a rectangular area containing a scene object (Figure 3.13(b)). Hitting a certain key starts the search process. The final result is shown in Figure 3.14, the video below (shown in two different frames) contains only the frames where the building appears. If it is possible to detect an object across the video, then it is also possible to add virtual content that uses that same object as an anchor. More interestingly, the virtual content should blend with the world following its anchor.

Extending the prototypes of Figures 3.10 and 3.12, it is possible to add a virtual object in one frame above the selected sample and replicate it to the other frames maintaining the relative position to the original sample using the Homography matrix. This is an example of applications where the user can introduce virtual content on non pre-defined real world scenarios, thus presenting a possible solution to **RQ1** (research questions in section 1.1).

In Figure 3.14(a), the virtual figure is placed above the selected sample area. Figure 3.14(b) shows the result of the replication to the other frames, where the virtual figure maintains its relative position to the original sample. This is a form of marker-less augmented reality, as stated in the end of section 3.2, that can be used to quickly add moving content to an entire video, blur unwanted movie parts or automatically create animated guides to improve post-production.

3.5 Stereo Vision

In the previous sections, the main focus has been the understanding of the image and how can it be described and searched. In order to create an internal model of the perceived scene, several aspects have to be inferred from the current descriptors. Using several images in the same context, the system can create a geometric three-dimensional internal model of the scene.

Stereo vision algorithms use two raster sources to triangulate between the points of two images and the cameras that captured them as explained by Forsyth and Ponce [FP02]. The main idea in stereo vision is that closer objects will have a higher displacement in the images than farther objects. If the two images are in overlapping planes with only a small translation, then the depth of a point is an inverse proportion of the displacement. This is the simplest case, but knowing the angles that the cameras have between photos, as explained in the previous section, it is possible to triangulate the depth. A prototype that explores the measurement of the depth of a point can be seen in Figure 3.15. The main problem with stereo vision is occlusion. To eliminate that problem multiple views are used instead of two. An example of this can be found in the work of Furukawa et al. [Fur+09b]. After obtaining the depth of points it is possible to reconstruct a scene in 3D [PH09; SSS08].

To create virtual applications that use as input a space from user-supplied photos, a spatial model of that space has to be extracted from the images (**RQ1**, from section 1.1). Figures 3.16 and 3.17 [NC12c] shows an example of how depth and surfaces can be extracted quickly from image keypoints. This means that a user can capture two pictures of a scene and use the detected model to insert 3D geometry in context.

In Figure 3.17, stereo-vision is used to detect an invisible 3D model of the scene. The top two images have a short side translation difference. The lines in green show the matching points on each image. The figures in the middle represent de Delaunay triangulation with the shading on the right representing the

Figure 3.15: Interactive prototype to roughly measure the distance of a point chosen by the user to the camera on the left image. The camera parameters must be known for this stereo view method.

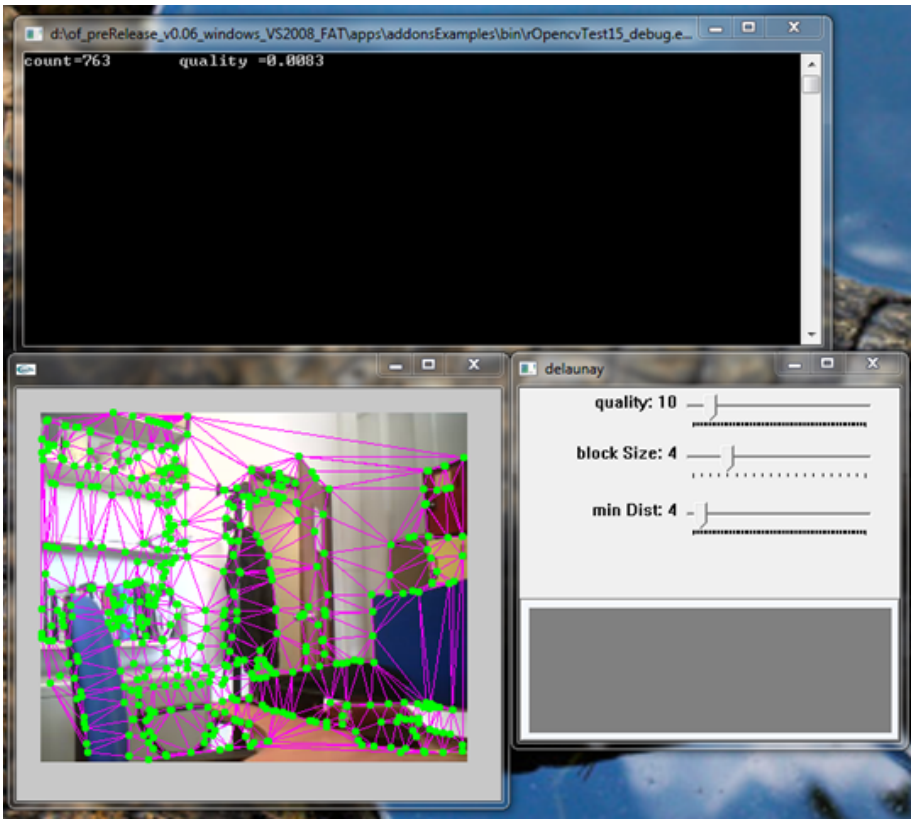


Figure 3.16: Delaunay triangulation prototype based on OpenCV.

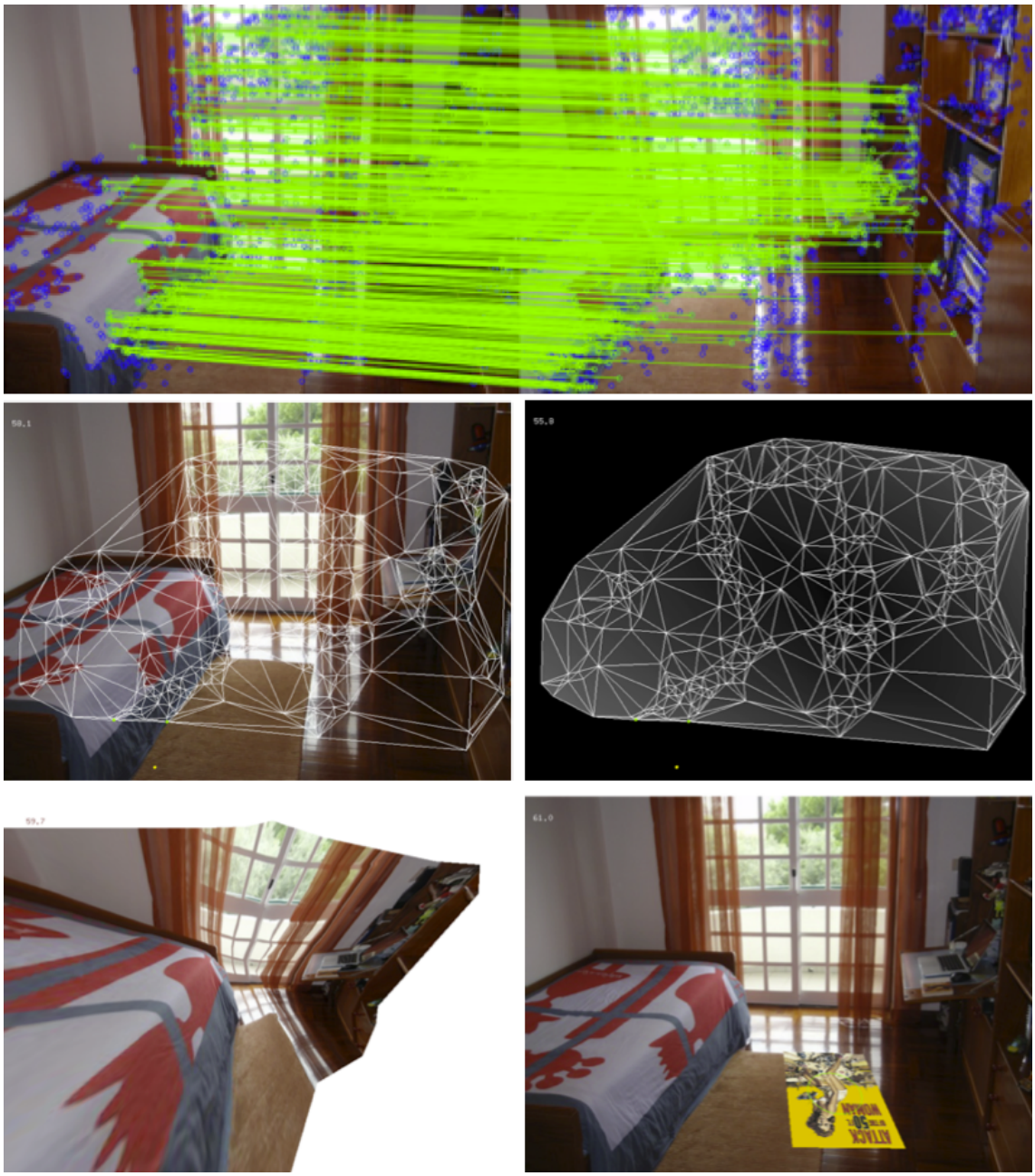


Figure 3.17: Interactive application for introduction of virtual objects attached to surfaces using the stereo-vision concept.

depth of the scene. The bottom left image presents a distorted view of the scene with a direct mapping of the pixels in the depth. On the right, a virtual object is included in the photograph with the correct distortion based on the invisible 3D model.

By analyzing the features of one image it is possible to extract the main keypoints of that image (section 3.1). These points are used to create a triangulated 3D mesh representing the image. This 3D mesh will not be rendered but will be used to support virtual content, which will use it as "floor".

To obtain the triangles for the 3D mesh a Delaunay triangulation is used. A prototype using this triangulation can be seen in Figure 3.16. The Delaunay triangulation (OpenCV [Ope13] implementation) calculates the best set of triangles, which fit the points, while trying to avoid extremely acute angles.

Using the Delaunay triangulation, the depth of each point is inferred based on the camera parameters and the displacement of each point between images. Using the normal vector of each polygon created with the triangulation, the virtual object (yellow poster) can be attached to surfaces.

Later in this dissertation, in subsection 5.1.4.2, a technique for introducing virtual objects glued to the invisible 3D detected scene will be presented.

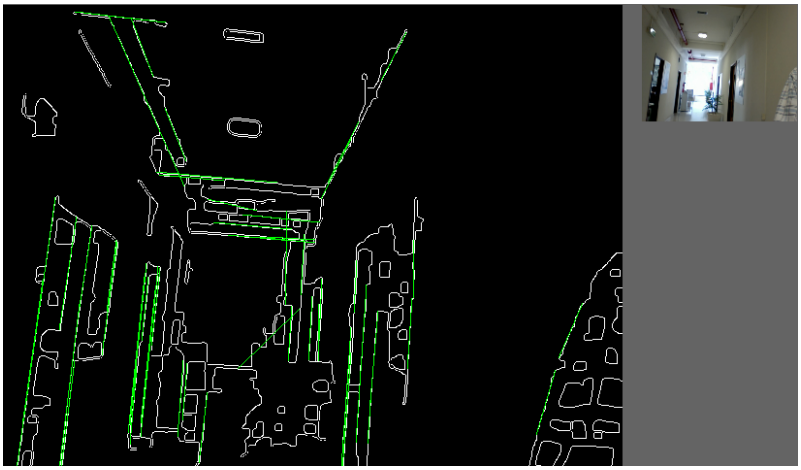
3.6 Edges and Lines

In the previous sections, images have been explored using keypoint based features. Another way of analyzing an image is by interpreting the main edges and lines that compose the picture. One of the most common solutions to detect edges is by applying the Canny filter [Can86] seen in Figure 3.18(a). Lines can be extracted using the Hough Transform [Hou62] with a variant to extract line segments (OpenCV implementation [Ope13]). The combination of the two algorithms can be seen in Figure 3.18.

Canny Edge Detector: The goal of the Canny edge detector [Can86] is to find the main edges of an image with a minimal error. It works by applying several convolution filters to every pixel in the image with the goal of finding pixels where the intensity variation is high. The algorithm starts by reducing the noise in the image by applying a Gaussian filter or an approximated Gaussian kernel. This will blur the image a little but will remove undesired noise points from the detection.

(a) Canny edges

(b) Hough-based line segments



(c) Sobreposition of Canny and Hough

Figure 3.18: Line segments and edges.

After de-noising, two Sobel kernels are applied to each pixel to find the gradients in the horizontal and vertical direction:

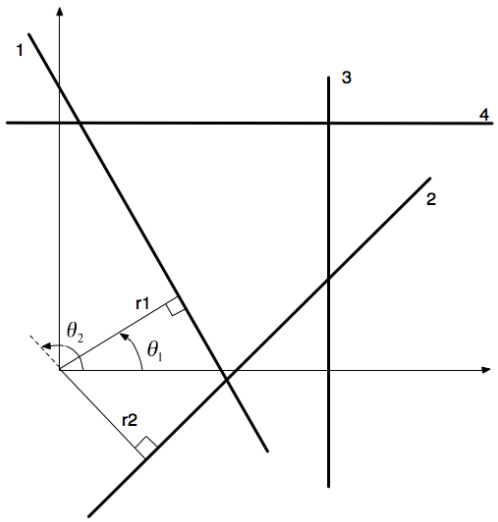
$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.13}$$

The gradient strength is calculated with:

$$G_p = \sqrt{G_x^2 + G_y^2} \tag{3.14}$$

A pixel will be accepted as an edge if it is above an upper threshold, below the lower threshold or connected to a pixel that is above the upper threshold. In the prototypes used throughout this dissertation the OpenCV implementation is used with the parameters: kernel size (e.g., 3, 5, 7,...), upper threshold and lower threshold ([0,255]).

Hough Line Transform The Hough Transform (HT) [Hou62] is a method for



Line 1: $r_1 = 10$, $\theta_1 = 30°$
Line 2: $r_2 = -8$, $\theta_2 = 135°$
Line 3: $r_3 = 25$, $\theta_3 = 0°$
Line 4: $r_4 = 23$, $\theta_4 = 90°$

Figure 3.19: Lines in Hough Space.

detecting linear structures in images. It can be used to isolate features of a particular shape within an image. It is commonly used in image processing to detect lines. Lines can be represented in Cartesian space using the equation

$$y = mx + c \tag{3.15}$$

For computational purposes it is sometimes better to use Polar Coordinates in the equation

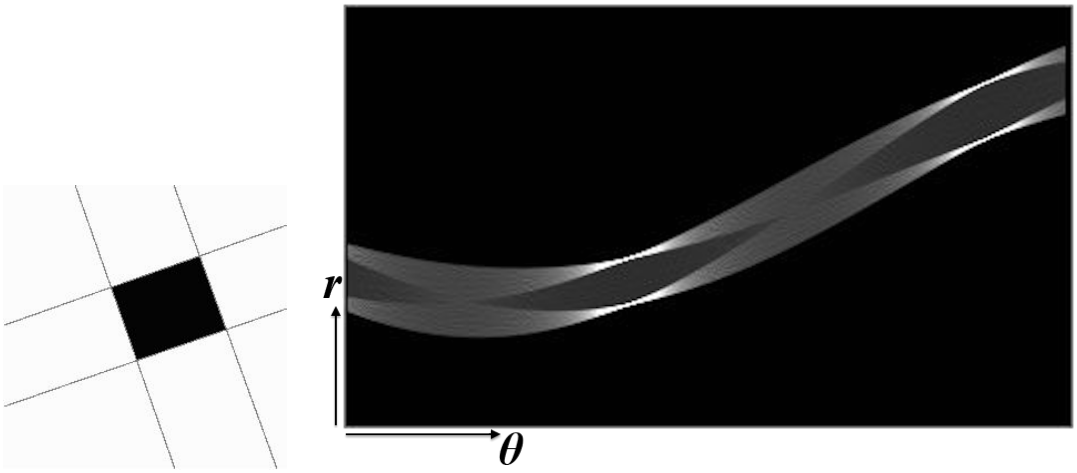
$$y = \left(-\frac{cos\theta}{sin\theta}\right) x + \left(\frac{r}{sin\theta}\right) \tag{3.16}$$

This can be rearranged in the form of a *Hough Space (r, θ)*:

$$r = x_0\ cos\theta + y_0\ sin\theta \tag{3.17}$$

Using this equation every line can be described as a combination of *(r, θ)*, where *r* is the length of the normal vector that connects the line to the origin and θ is the angle of vector r. In order to be able to represent all possible lines, the angle varies in the interval [0°,180°[and *r* is negative whenever it is below the abscissa, as exemplified in Figure 3.19.

To find the main lines of the image the Hough Line Transform scans a large quantity of points in a image after applying a Canny filter (or equivalent). These



(a) Rectangle (b) Hough space

Figure 3.20: Several types of lines.
(Source: `https://courseware.ee.calpoly.edu/~fdepiero/STL/STL-Image-HoughTransform.htm`)

points can be random, the main corners or all pixels. For each point several lines with different (r, θ) values are tested. Each generated line is scanned to detect how many edge pixels it overlaps. The count for each line is saved in an accumulator. Figure 3.20 represents an image of a rectangle with the detected lines (Figure 3.20(a)) and a chart of the Hough Space (Figure 3.20(b)) for that image in (r, θ) coordinates. The chart in Figure 3.20(a) represents the value of the accumulator for all the tested lines (white means high value). Finding the maximum values (above a threshold) of the Hough Space will provide the main lines of the image. In the example of Figure 3.20(a), there are four main lines, represented by the four whitest spots in the chart.

The traditional Hough Line Transform finds all the lines in an image. This algorithm can be extended to provide additional functionalities such as line segment detection by introducing parameters such as minimum gap between lines or minimum length of the lines. In the prototypes developed in the scope of this dissertation and especially in Chapter 4 the OpenCV implementation is used with the parameters (Figure 3.21): Canny edge detector parameters, minimum angle difference tested, threshold to be considered line, minimum line length and minimum gap between lines (OpenCV documentation [Ope13]).

Polygons: Using the extracted line segments it is possible to detect polygons. Polygons can also be extracted by contour fitting algorithms or by testing detected line segments. Using the extracted line segments it is possible to detect

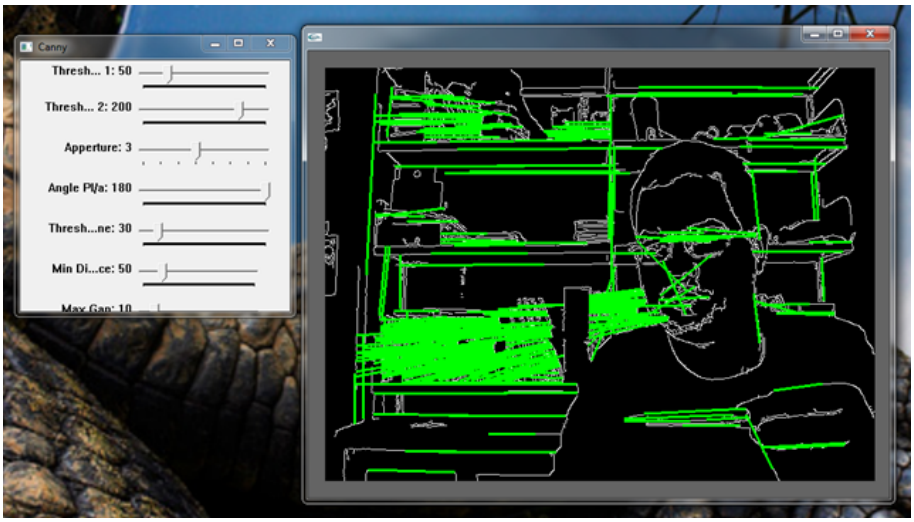


Figure 3.21: Hough line segments with OpenCV implementation.

(a) All polygons (b) Quadrilaterals

Figure 3.22: Several types of lines.

complex polygons as seen in Figure 3.22. To detect specific shapes such as quadrilaterals (Figure 3.22(b)), the number of sides of each polygon can be limited to four and the inner angles made by the connected lines must be concave. Concavity can be tested by using the cross product between lines.

This section summarized how edges and lines can be obtained. Several applications of these features will be presented in Chapter 4 and Chapter 5.

3.7 Implementation and Prototypes

All the prototypes presented in this chapter were implemented in C++, using OpenCV [Ope13] structures and algorithm implementations (namely SURF, SIFT, Canny edge detection, Hough Line detection and Delaunay triangulation). They run on standard dual core computers with 4GB of RAM and the proof-of-concept examples were built in OpenGL with the support of the openFrameworks library [ope13].

3.8 Summary

This chapter explores and details several low-level features, which can be extracted from images. It also summarizes several relevant algorithms in order to make this dissertation a self-contained document. Most of these algorithms are extensively used in Chapter 4 and applied in the prototypes in Chapter 5.

Several proofs-of-concept are presented to illustrate the studied techniques with special relevance to two prototypes. (1) An application to find scene elements in video and superimpose it with virtual elements (section 3.4) and (2) an application to find depth in a scenario based on two images using stereo-vision (section 3.5).

The main goal of this chapter was to provide the basic elements and features as building blocks for the most complex systems proposed in the next chapter.

4

Inserting Virtual Objects in Images

This chapter presents a systematic approach to acquire a real world scenario and superimpose information according to detected image features. Inspired by the state-of-art presented in Chapter 2 and using the techniques described in Chapter 3 here, several concepts, ideas and techniques are proposed to address the main research questions defined in Chapter 1.

Combining points of interest and detected lines, it is possible to analyze images with low computational effort and detect enough elements that can be used to anchor and define surfaces for mixed and augmented reality. The starting point for this research is the motivation to build interactive applications allowing the user to recreate and change virtually a physical space. Possible implementations of this system would be important in scenarios such as room decoration, house interior rearranging, furniture manipulation, games, 3D modeling and animation systems.

The main research questions addressed in this chapter can be divided in two parts. First, is it feasible to create interactive mixed reality applications where virtual objects interact in a user chosen real world scenario? (**RQ1**, section 1.1) Secondly, are untrained users capable and motivated enough to perform the necessary steps to initialize the AR environment (**RQ2**, section 1.1)? Altogether, what different possibilities exist for markerless AR applications? To answer these questions a computer vision system, which analyses pictures captured by the users, was developed. More than building a system for perfect photo-realistic 3D reconstruction, the intention is to construct a robust model to be used by a non-expert

user. The presented system was used to create prototypes, which were tested with several users in different environments to evaluate the reliability of the interaction. The prototypes and user studies will be presented in the next chapter.

The proposed mixed and augmented reality system aims to create applications based on the user's physical space. The main objective is to capture the physical environment without using markers in the real world and using only widely available sensors such as webcams or smartphone cameras. In order to accomplish this goal, an internal model of the real world must be created so that the mixed reality application can use the captured images. The cameras should have the same features and specifications of any available camera of standard smartphones, tablets or laptops. This excludes cameras with special sensors, such as depth cameras.

Taking into account the aforementioned objectives, the main requirements of the proposed system were analyzed. A user that does not have much experience in handling 3D manipulation or cameras should be able to make the acquisition of the augmented reality application scenario. The user should be able to start the system with basic instructions and little or no training at all. The system should automatically create the spatial model and track the high-level features required for the introduction of virtual elements. The interaction with the application that uses the model should be natural and virtual elements should respond to the world's physical properties (e.g., gravity and collisions with the ground).

Most of the work described in this chapter was proposed [Nó09; Nó11], analysed [NC11], described [NC12d; NC13a; NC13b] and demonstrated [NC12a] in several publications.

4.1 Concept Overview

With the advent of tablets and smartphones, with frontal and back cameras, new opportunities arise for multimedia applications that mix captured images and virtual content. Images and videos can be recorded instantly and immediately used in the multimedia system or saved for later interaction. Another obvious possibility is the real-time interaction with the real world using the camera.

Using real video or captured images as input for interactive mobile applications (Figure 4.1) presents several challenges that need to be addressed.

First of all, there is a factor of uncertainty related with the user's skills to capture good quality images. Good quality images are not only defined by their beauty or good illumination properties but by the subject of the photography.

Figure 4.1: Handheld application using camera. The current prototypes run in tablets and regular computers with webcams. Currently most tablets already incorporate frontal and back cameras.

Depending on the application the user may be asked to target certain objects, empty spaces or spaces with certain qualities (e.g., with many lines). The user may shoot the photograph too close or too far, shake the camera or not fully frame the subject of relevance.

There can be limitations related with the camera device itself, as the image can be blurred and the scene poorly illuminated. Many webcams and smartphone cameras try to balance the illumination of the scene. This can lead to a large variation in the colors of the image. These are common problems in all computer vision projects.

At the interaction level, the main challenges are related with the heterogeneity of the possible photographed scenes. Including special tags or markers in the scene may improve the reliability of mixed reality scenes but introduces additional interaction hazards for the user. The user has to print/build/move physical markers and insert them in the environment. This makes certain applications difficult to replicate and less scalable because they are not self-contained in one digital device.

At the mixed reality application level, users may have difficulties in positioning or moving objects in a 3D space. They may expect that the properties of the

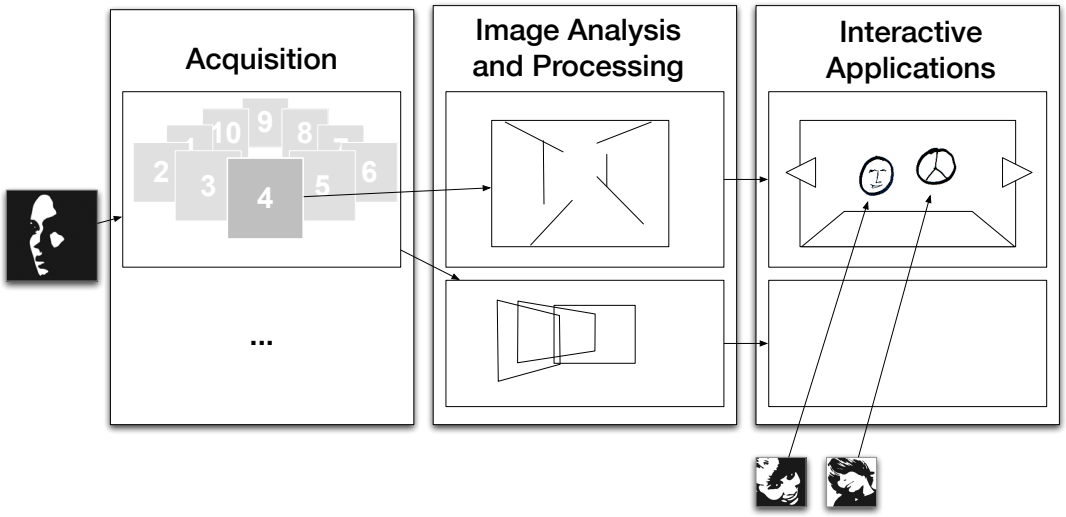


Figure 4.2: Requirements diagram (1) Acquisition, (2) Information Processing and (3) Interactive Applications.

real world apply to the virtual objects in the AR application. As an example, virtual objects should collide and be occluded by objects in the real world. There are several properties that can be explored in mixed reality in order to create rich multimedia applications, such as:

- **3D structure:** detect the 3D mesh model corresponding to one or a set of images.
- **Segmentation:** detect areas or regions with common colors or patterns.
- **Object detection:** detect known objects, faces, façades and markers.
- **Flow detection:** detect movement, directions and flow.
- **Tracking:** track objects, corners, points of interest, colors or people.

Combining several techniques is sometimes difficult due to increasing processing cost (image processing is usually an heavy task). In this dissertation, the main focus is on 3D structure and orientation detection from a single image and segmentation of user assigned areas.

The proposed interaction concept can be divided into three stages [Nó11]: (1) Acquisition, (2) Information Processing and (3) Interactive Applications, as detailed in Figure 4.2. In the first stage (1) images and information about the scenario where the AR application will be deployed have to be acquired. Here the system interface plays a key role because it should instruct the user on how to take pictures with some meaning and that are useful for the system. Additionally,

the user may be required to locate certain visual elements such as floor or ceiling in order to assist detection algorithms. These image processing and computer vision algorithms come into play in a second stage (2). Taking the obtained information, the visual detection targets elements such as floor, horizon, vanishing points, scene orientation and presence of three-dimensional elements. Using all the detected features several interactive applications can be created (3) enabling the possibility of interacting with virtual elements in a real space chosen by the user. These three stages will be revisited in greater detail in the following sections where this system is presented.

4.2 Problem Formalization

The fundamental concept for the proposed system is the possibility of creating mixed and augmented reality applications that take advantage of detected visual features inside images of real world spaces. These images are obtained with regular cameras with no additional hardware.

The research problem being addressed here is how to embed virtual objects in an image, in such a way, that the virtual objects react to physical concepts (i.e., gravity, obstacles) inside the virtual world. As presented in Chapter 2, several previous studies [KHF11] have found different answers to this problem.

The main problem can be divided into the following different sub problems:

$$\delta = f(I), \tag{4.1}$$

$$\varphi = g(I, \delta, C), \ and \tag{4.2}$$

$$arApp(I, O, M) \ where \ M = \{\varphi, \delta, C\} \tag{4.3}$$

These three equations summarize the main steps required to create a mixed reality application based on an image I. The main objective is to build a logical model M, which is divided into two groups of features, δ and φ. The δ features are considered low-level features because they are obtained directly from the raw image I. The camera parameters C are not directly obtained from I but will be also considered a low-level feature to simplify the terms. The φ features are considered high-level features because they are obtained through a processing algorithm from the low-level feature δ.

In the first equation (Equation 4.1) the input image I must be analyzed using

f to search for low-level features δ such as edges, straight lines or relevant points. In the implemented solution this constitutes the low-level section of model M as explained in Figure 4.3. The function f represents the initial preprocessing steps to transform the raw image into low-level features using algorithms such as the ones described in Chapter 3. The f function also encapsulates possible questions and tasks given to the user (e.g., selecting the floor) to assist in the process of low-level features detection.

Figure 4.3 represent the low-level δ section of the logical model of the application. It contains the image itself, e iterations of detected edges and for each edge image, h iterations of line detection. In other words there are e edges obtained using the Canny edge detector and for each edge h sets of lines obtained using the Hough line transform (section 3.6). The goal is to have redundancy so that every line in every image light condition is extracted (explained in section 4.3.3). The user will be asked for a simple selection of the floor (explained in section 4.3.2) and the camera parameters should be known for the camera device that will be used.

Using δ, the image I and the camera parameters C, the function g (Equation 4.2) attempts to find high-level features φ such as vanishing points, horizon, room orientation or floor, as seen in Figure 4.4. The camera parameters C and all the detected features and constraints, compose the virtual model M of the scene.

Figure 4.4 explains the three main layers that compose the high-level features φ. These are obtained from the δ features represented in a darker color. The (φ_1) vanishing points rely on the aggregation of the main line segments after a decluttering filter explained in section 4.3.3. The detection of the vanishing points itself is thoroughly discussed in section 4.3.4. The scene orientation (φ_2) is highly dependent on the Camera parameters and is explained in section 4.3.5. Finally the floor (φ_3) is detected using a selection mask introduced by the user. This is explained in section 4.3.2 and 4.3.6.

Revisiting the three equations from the beginning of the section, the last equation (Equation 4.3) represents the final mixed and augmented reality application that receives the model, the virtual objects O and the input image I. There are many possible applications that can take advantage of the detected model M. Several applications are suggested and explored in Chapter 5.

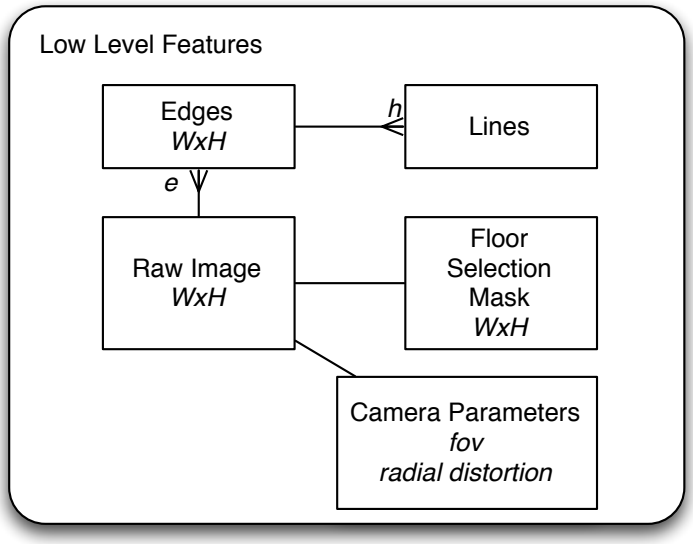


Figure 4.3: Low-level features δ of model M including *a priori* known camera parameters C. These are features that are directly extracted from the raw image I, which has a dimension of $W \times H$.

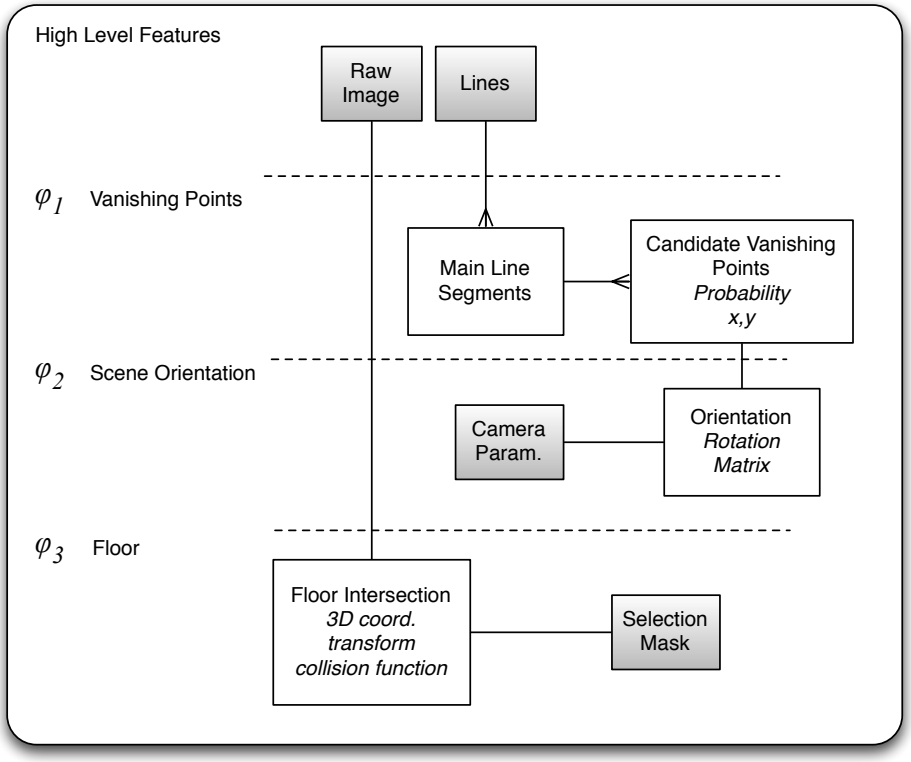


Figure 4.4: High-level features φ of model M including their relation with the low-level features δ (seen in darker shade). These are composite features that are extracted from the image I and from δ.

4.3 Implemented Solution

Using the previously discussed formalization, the current implemented solution follows the structure presented in Figure 4.5. Our solution is based on the interpretation of the main straight lines of the scene, much in the same way as Lee et al. [LHK09]. The goal is to detect as much lines as possible, with different illumination thresholds, and delay, as much as possible, the decision of which of the parameters is the best for a correct line analysis of the current image.

By analysing the lines and their intersections it is possible to identify a large set of candidate vanishing points that can be used to observe the main orientation of the scene. Using a clustering methodology the main vanishing points of the scene are extracted, thus acquiring the vanishing points feature (φ_1). This is done using a similar technique as the one proposed by Rother [Rot02].

The image I is assumed to be an image of a Manhattan World scenario [CY99; Del+11], where the floor is in the lower part of the image and there is a large uncluttered space in the room, where the virtual objects can be laid on. The camera parameters C are assumed to be known, and when they are not, they can sometimes be extrapolated from the vanishing points, or several prior models can be assumed (e.g., webcams). This means that each scene will have one, two or three highly identifiable vanishing points because most of the lines will be parallel in the real world. If we exclude the vertical axis, two of these vanishing points will define the horizon and have a high probability of being vertically situated inside the image (even if horizontally they are outside). These are common heuristics and assumptions that were considered in this work.

Having found the vanishing points (φ_1), the scene orientation (φ_2) is obtained by aligning the virtual objects O with the detected vanishing points by rotating them in the adequate proportion.

The floor (φ_3) is extracted using segmentation techniques [RK04] and 2D to 3D transformations.

The current model, depicted in Figures 4.3 and 4.4, focuses mainly on these three high-level features but can be easily extended to incorporate others [Sim06] such as walls or plain surfaces. These features are primarily obtained trough automatic analysis of the image (f and g) but may need to be refined through a human in the loop process [KCG11] where the user can give simple hints to help the algorithm.

The implementation of our solution is summarized in Figure 4.5 and the next

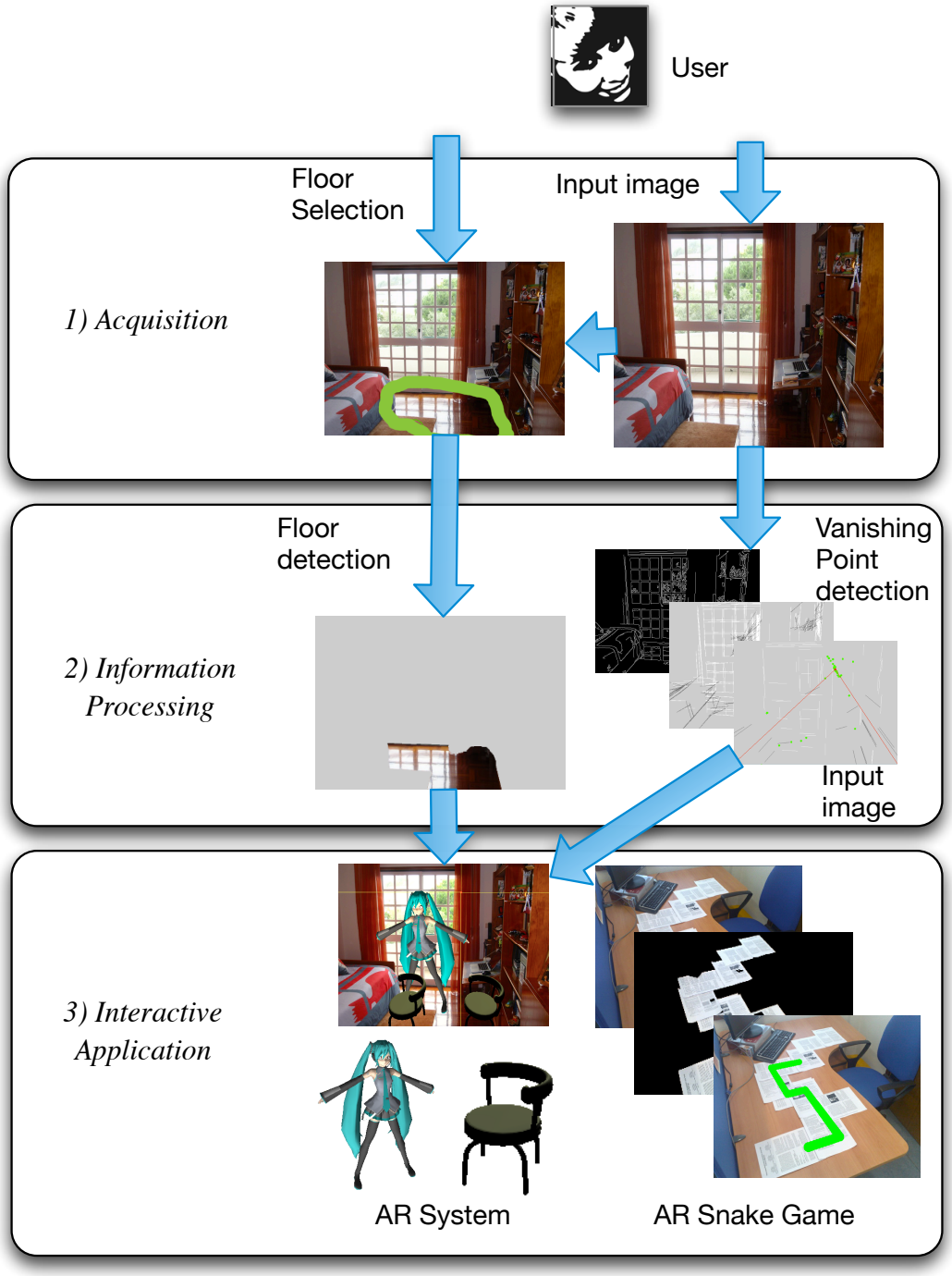


Figure 4.5: Diagram explaining the several steps involved in semi-automatic construction of two Augmented Reality applications. The contribution of the user consists in giving an input image and roughly selecting the floor. The AR model is then automatically inferred.

sections will detail the construction of the augmented reality model M. The figure also details the three main stages of the algorithm proposed in the Concept Overview section: (1) Acquisition, (2) Information Processing and (3) Interactive Applications.

The current solution requires an initial input provided by the user consisting of an image and a floor selection; this is further discussed in the next section. The floor detection is based on a segmentation algorithm described in subsection 4.3.2. The scene structure detection is based on line segment analysis and is described in section 4.3.4. The scene orientation, which allows the insertion of virtual objects in the same perspective of the world, is described in subsection 4.3.5.

4.3.1 Input Image

The initial input of the system is a raster image I that meets the assumptions described in section 4.1 (Concept Overview). One of the main obvious premises is that the image should not be blurred and it should contain enough detail for feature detection. In the current prototype, images can be obtained through a webcam or by loading a file from the disk. When using a camera, there is a system to help the user selecting a good picture. The system does a real-time analysis of the image to understand if it has enough potential. To assess if the image is complex enough for further analysis, FAST [RD05] or SURF [Bay+08] feature points and line segments are extracted. For the image to be accepted the amount of detected features must be above a certain threshold value (dependent on the resolution).

One of the core steps for the mixed reality system is the image acquisition process from the space where the application will run. As already stated, the correct detection of the three-dimensional model of the scene depends on image analysis algorithms. For this reason it is required that the picture has enough content to be analyzed. This excludes photographs of blank walls and very close or blurry shots. Ideally, the photos must contain a large quantity of straight lines and smooth surfaces with Lambertian diffuse textures [Fol+95]. Typically this type of photographs can be found in urban buildings and house interiors, where the walls, floor, ceiling and furniture create rectangular angles (the already mentioned Manhattan World scenarios [CY99]).

Another problem is that cameras have different parameters such as angle of view, depth of field and radial distortion. The current system does not consider in detail the camera calibration methods [BDS10; PKG98], instead, it assumes that

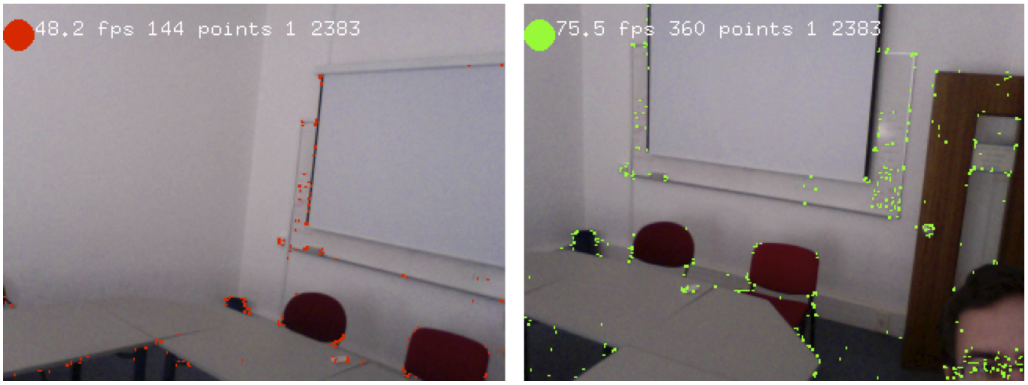


Figure 4.6: Snapshot of the image capturing system. The application gives an hint to the user about the amount of detail in the image. The right image (with a green circle) has enough detail to detect the necessary visual cues.

the application knows the correct parameters of the device that it is using. Additionally the scale and size of the scene needs to be assessed to understand the size of the introduced virtual objects.

Taking into consideration that the current prototype supports the introduction of images through webcams or directly through an image file, capturing a meaningful picture of the scene is vital for the system to work. The user is instructed to take a picture holding the camera straight. It should have reference points such as edges, doors or windows. Using the webcam mode, the system advises the user in real-time of whether an image is detailed enough as depicted in Figure 4.6.

The system detects SURF [Bay+08] feature points on each frame and only allows a picture to be captured when a given threshold of points has been surpassed. This threshold should be dependent on the amount of scene luminance saturation. This improves the chances of detecting properties in the images for future interaction.

For more information about the system, several instructions or questions may be suggested in order to get simple information about the captured space. Some of these instructions were later implemented in the prototype. The suggested questions or tasks are:

"Draw a line around the floor." The user draws an outline roughly around the area that considers the floor of the scene. The location of the ground can then be extracted by using segmentation algorithms.

"Draw a line around objects that are on or in front of the ground." Similar to the previous task, here the goal is to infer what scene objects should be

presented in front of the virtual objects occluding them.

"How high from the floor was the photograph captured? (1.70 m by default)" This question will help to infer what should be the position of the virtual plane that represents the ground.

"The horizon was correctly detected?" This should be presented in the interface with two or three horizon line hypotheses. The analysis system (explained below) typically presents several results with different degrees of probability. Sometimes, the horizon with a higher degree of probability is not the most correct, so the user can choose another option.

"Draw a straight line along a vertical edge of a door or window." Knowing the average height of a door or a window or a table, some information can be inferred about the scale of the objects that will be inserted. This also allows a more accurate detection of the vertical direction of the scene.

All information gathered can be incorporated into the model M of the AR application scenario. The next sections will explain briefly how the input image is processed.

4.3.2 Region Segmentation

Focusing on the first suggested task in the previous section, *"Draw a line around the floor"*, there are many hints that the user can give. Any user can immediately identity the visual area of the floor in a picture. In the input example presented in Figure 4.7 the user is asked to *"Sketch a line around the floor"*. The word *Sketch* is used instead of *Draw* to emphasize the fact that what is expected is a rough or unfinished drawing around the area defined as floor. Using this rough estimation the system can detect the scene floor (φ_3).

The detection of the scene floor (φ_3) is required for the AR application to understand where to place the virtual objects. The rough sketch is used as input for an implementation of the GrabCut segmentation algorithm [RK04] to define a mask $m_{\varphi 3}$, where for each pixel it states if it belongs to the class floor or not. This mask will later be important to verify collisions with the floor.

The GrabCut algorithm [RK04] is an interactive approach to foreground extraction using graph cuts. The proposed algorithm, by Rother et al.[RK04], uses a human in the loop process where the user selects, in several iterations what part of the image is more likely to be a foreground element that is distinct from the

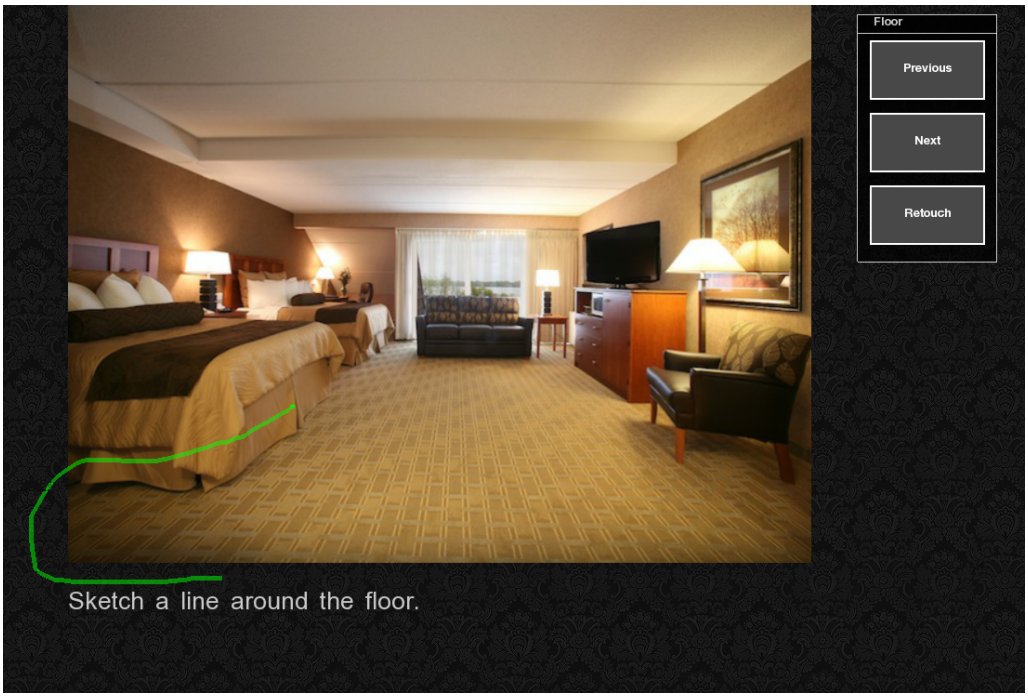


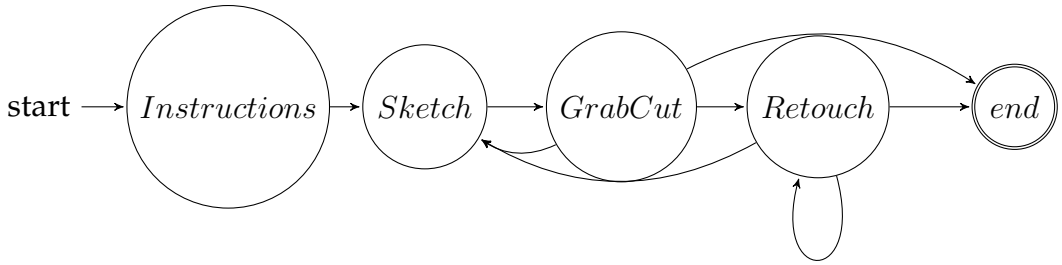
Figure 4.7: Prototype user interface asking the user to sketch a line around the picture area that is considered the floor.

rest of the image. The goal is to isolate all the elements considered foreground without a precise selection.

The iterative algorithm works by alternating rounds of automatic segmentation with user interaction to correct possible mistakes. The user initially selects an area as being of the foreground class and everything else is considered background. Using an energy function based on the colors of each pixel, the algorithm attempts to minimize this function to detect a visually coherent image inside the foreground area. After each round of minimization a masked image is presented to the user with the results of the segmentation, as presented in Figure 4.8. If there is a part of the image that was not correctly included the user can select further areas that should be considered foreground as well as areas that are not. Using the extra feedback and the previous results, the algorithm runs another round of minimization and presents another possible foreground mask. This can iteratively go on until the user is satisfied.

The current implemented system follows a similar workflow as the GrabCut interactive solution as seen in Figures 4.7 and 4.8. To acquire the floor mask $m_{\varphi 3}$ the following interaction flow takes place:

Figure 4.8: Using the GrabCut algorithm to extract the floor mask (on the right). The user roughly sketches the green line (best seen in color) around the floor and the algorithm extracts the dominant floor pixels.



Initially the simple instructions are presented as depicted in Figure 4.7. The user sketches a line around the floor and immediately after the line is drawn the algorithm based on an OpenCV [Ope13] implementation of GrabCut is executed. The execution time is negligible in modern computers (around one second or less). The main difference between the proposed system and the one presented by Rother et al. [RK04] is that instead of several incremental iterations of sketches here only one is used. If the user is not satisfied with the segmentation result presented in Figure 4.8, he/she can always do the selection process again. After that, comes a *Retouch* step where the user can paint or erase the mask using the mouse (Figure 4.9).

This new approach has some drawbacks and some advantages. The main drawback is that an experienced user can use the iterative process proposed by

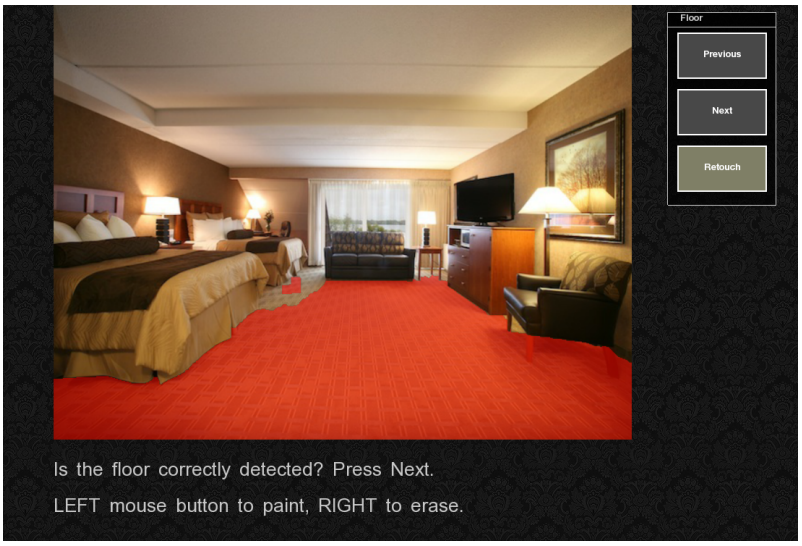


Figure 4.9: Prototype user interface for retouching. After selecting the floor, the detected floor mask is shown in transparent red. At this point the user has three choices: be satisfied with the selection and continue to next stage, select again or retouch using the mouse to paint the appropriate area.

Rother et al.[RK04] to obtain a better and more accurate segmentation of the floor. The main advantage is that the interaction requires less instructions and it is easier to learn. In the current implementation the only instructions are to sketch a line and to paint the rest of the mask or start all over again. Explaining what foreground and background selection areas are and how to add them iteratively presents several challenges for inexperienced users.

After this process is over, the acquisition of the floor selection mask from the low-level features δ (Figure 4.3) is calculated and the Acquisition phase is over (Figure 4.5). The current prototype is only used to detect the floor area, but could effectively be transformed to acquire knowledge of other elements (e.g.,vertical wall, ceiling, tables and other specific furniture, trees or nature elements).

Currently this uses a human in the loop approach (e.g., Kowdle et al. [KCG11]) but more complex systems (e.g., statistical, prior knowledge base inference) could be introduced to do this step automatically. Some of these systems are detailed in the related work in section 2.2.1.

4.3.3 Detecting Main Lines

In this section the detection of edges and lines is explained. From now on all the features are calculated, thus starting the Information Processing step (2) presented before in the algorithm in Figure 4.5. The main goal is to calculate the main lines of the scene φ_1. For that, the low-level features Edges and Lines (δ

from Figure 4.3) are first detected.

One of the main problems with edge and line detection is that they are very prone to illumination problems. There is the risk of not detecting enough lines in poorly illuminated scenes and on the other side, detecting too many lines in luminance saturated pictures, thus having to deal with clutter and excess of information. The goal is to have a solution that handles both problems by having enough lines and avoiding clutter.

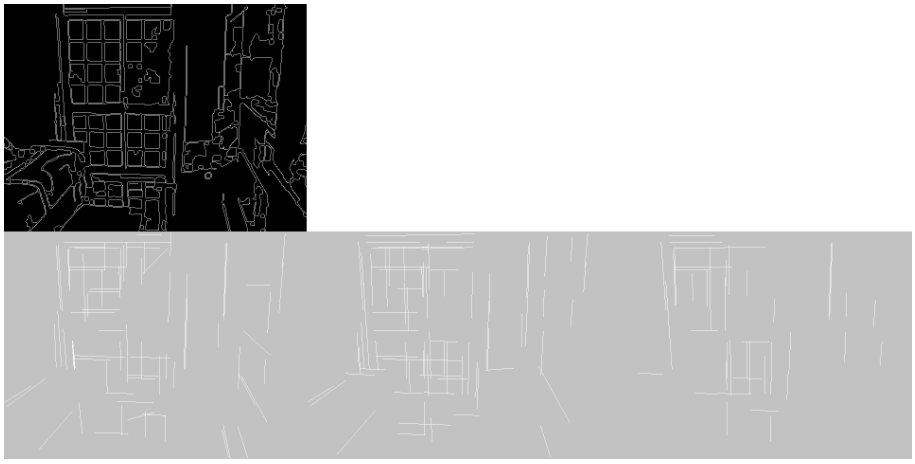
The current implemented approach makes use of several filters to be prepared for all illumination scenarios. The main difference in this algorithm is that the scene is analysed several times $(e \times h)$ (Figure 4.10) with different parameters. First there are e different edge detectors that use different thresholds in a Canny filter (section 3.6). Additionally, the edges can be refined with Erode and Dilate filters to clean the outliers. Secondly, there are h iterations of the Hough transform line detector (section 3.6) with different gap tolerances and different minimum and maximum lengths. To be accurate the current used implementation of the Hough line detector returns line segments.

In the end, the system has $e \times h$ line sets from which to choose. The choice of the best set of parameters is delayed until the last moment to evaluate through a score function, which one gives better results. This is important to find the main lines, which is a preparation step to find the main vanishing points φ_1.

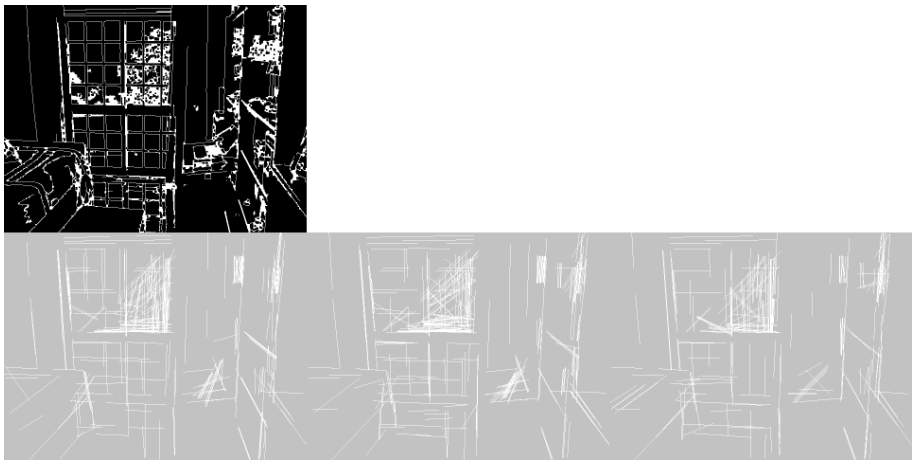
The score function helps to decide which parameters give a better, uncluttered view of the scene. To calculate s, the first score, all lines are classified using its slope as Horizontal $(0° \pm 5°)$, Vertical $(90° \pm 5°)$ and Oblique (everything else). Considering only the Oblique line segments, for each line the number of intersections $int(i)$ with other line segments is counted. Line segments with a large number of intersections are considered clutter and have a lower weight in s, longer line segments give an extra weigh $w(i)$ to

$$s = \sum_{i}^{n} \frac{w(i)}{int(i) + 1} \tag{4.4}$$

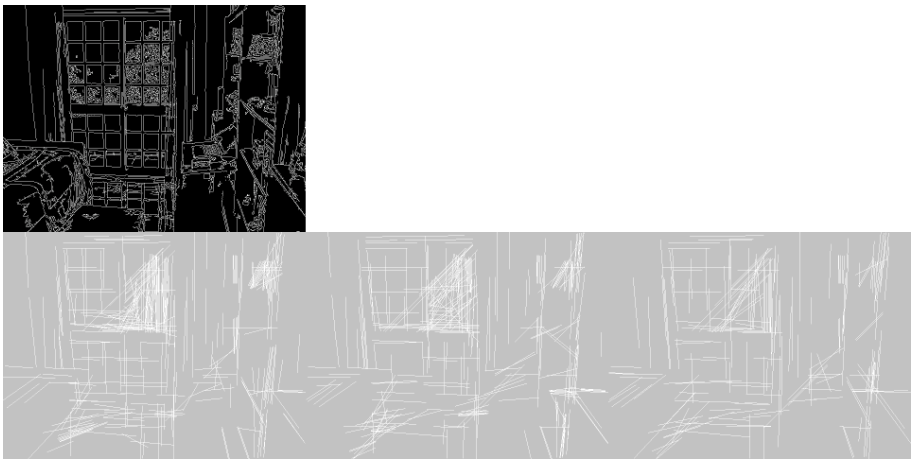
Having chosen the line set with better s score, the system is ready to step into the next phase, which is, to calculate the main vanishing points. Figure 4.11 summarizes the entire process while additional parameter tests for optimal detection were performed and are presented later in subsection 4.4.1.



(a)



(b)

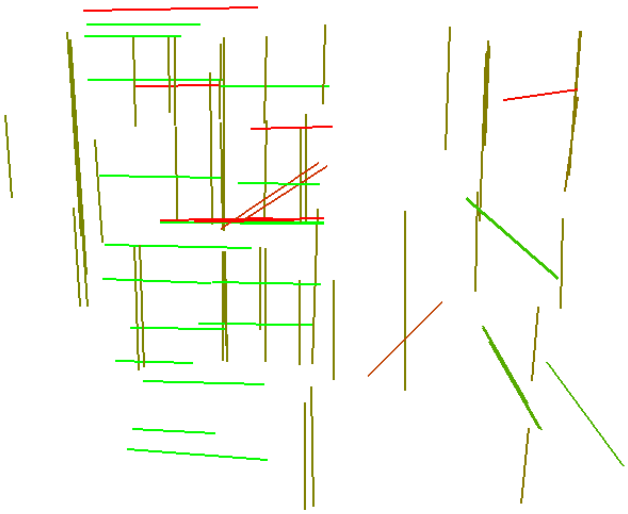


(c)

Figure 4.10: Line detection with several parameters. A single image was analysed and three different levels of edges were detected using Canny filters with different thresholds. For each set of edges, three sets of lines were calculated through the Hough line segment detection algorithm. See detail in Figure 4.11.

(a)

(b)



(c)

Figure 4.11: Main line detection (a) Input Image, (b) Canny edge detector, (c) main lines after de-cluttering filter through score s (colored according to slope).

Algorithm 1 Generating Candidate Vanishing Points.

```
Initializations

Finding the Main Lines from a set of e × h line sets.

Finding the line intersections.

Using RANSAC choose random intersections.

Finding the most relevant intersections using clustering.

return 1 to 3 points from the clustering.
```

4.3.4 Vanishing Point Detection

After acquiring the image and an estimate of where the floor is, the scene must be analyzed in search for the remaining features (φ_1 and φ_2, as stated in g).

The vanishing points (φ_1), are crucial to the augmented reality system. Virtual objects must fade into the same horizon as the scene. If at least two major vanishing points are found, the field-of-view can be extrapolated. The world orientation (φ_2) of the scene (more details on the next section) can be identified and emulated.

Our algorithm for finding candidate vanishing points [LHK09; Rot02] is briefly described in the Algorithm 1 code block and an extended version is presented in Algorithm 2. It analyzes the main lines containing the scene and tries to extrapolate the three most common line directions of the scene, bearing in mind that parallel lines have common vanishing points. Figure 4.5 illustrates the current algorithm. Code block 2 also illustrates how the main lines are calculated and stored as seen in the previous section.

Having chosen the line set with better s score (section 4.3.3), the same line set is analyzed for intersections between the lines. For efficiency reasons only some pairs of lines are tested using Random Sample Consensus (RANSAC)[FB81]. Each intersection t_{ij} is tested against several constraints, namely the intersection should be outside the two line segments and the line segments should have a slope difference above 10 degrees.

Figure 4.12 represents a simplified diagram of the class system, which was implemented following the Algorithm 2.

In the end, there should be a large number of two-dimensional points populating the scene. The three vanishing points are extracted using a nearest neighbors approach to cluster the most common groups of points as seen in Figure 4.13.

Algorithm 2 Generating Candidate Vanishing Points (extended).

```
1  Let I        // the input image
2
3  Let E[e]     // the vector of e images containing Canny edges of
4               // the scene with different thresholds.
5
6  Let L[e][h]  // the vector of e × h line sets containing
7               // different detected lines for each E[i].
8
9  Let S ← ∅    // the score of the line set.
10
11 Let S' ← ∅   // the reference to the line set.
12
13
14 for i = 1 until e
15   E[i]← Canny(I, cannyParam_i)
16   for j = 1 until h
17     L[i][j] ←HoughLines(I, lineParam_j)
18     s←score(L[i][j])  // tests if has enough information
19                       //or excessive clutter.
20     if s > S then
21         S ← s
22         S' ← L[i][j]
23     endif
24   endfor
25 endfor
26
27
28 Let (l_i, l_j) ∈ S'
29
30 Let V ← be the set of line intersections.
31
32
33 for i × j random pairs of line segments (l_i, l_j)
34    // do (RANSAC)
35   t_ij ←intersection of (l_i, l_j).
36   if constrains(t_ij) = 1 then
37     add intersection to V
38   endif
39 endfor
40
41 v ←clustering(V)  // verify where are the
42                   // most predominant intersections
43                   // according to x, y and k
44
45 return v, vector with 1 to 3 points
```

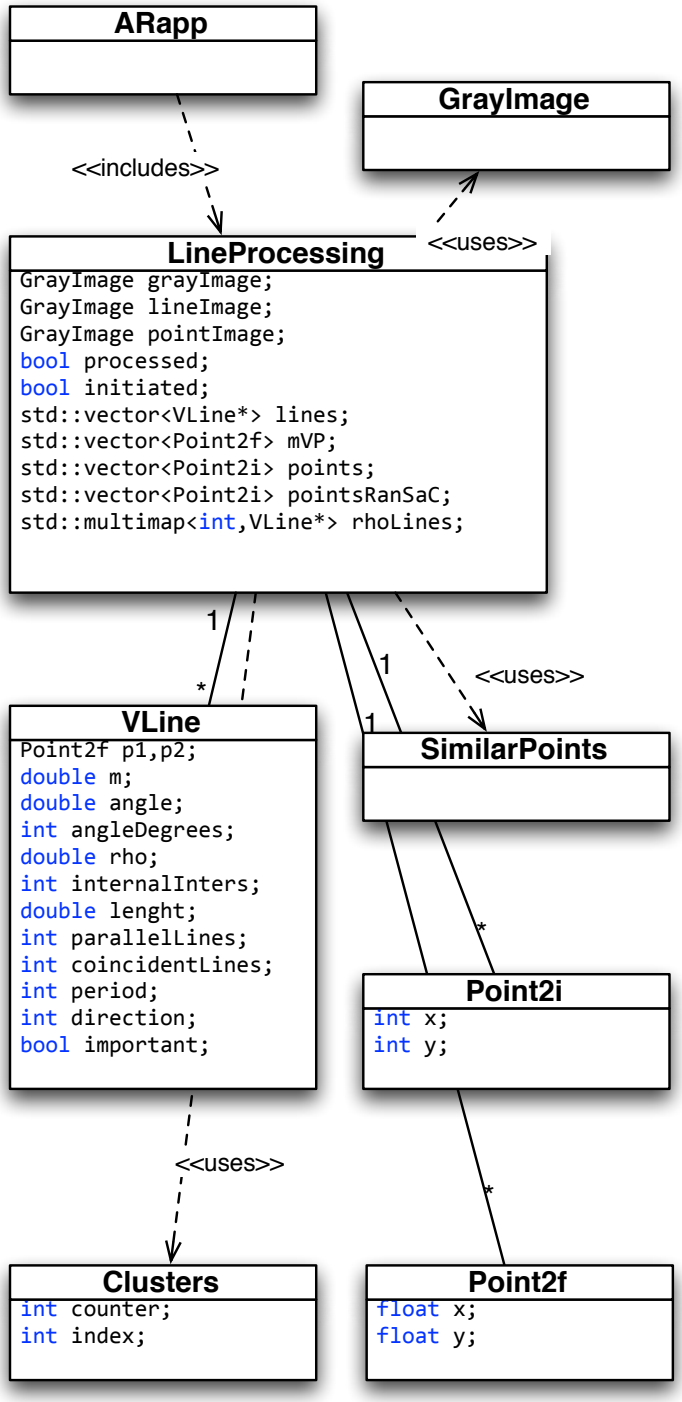
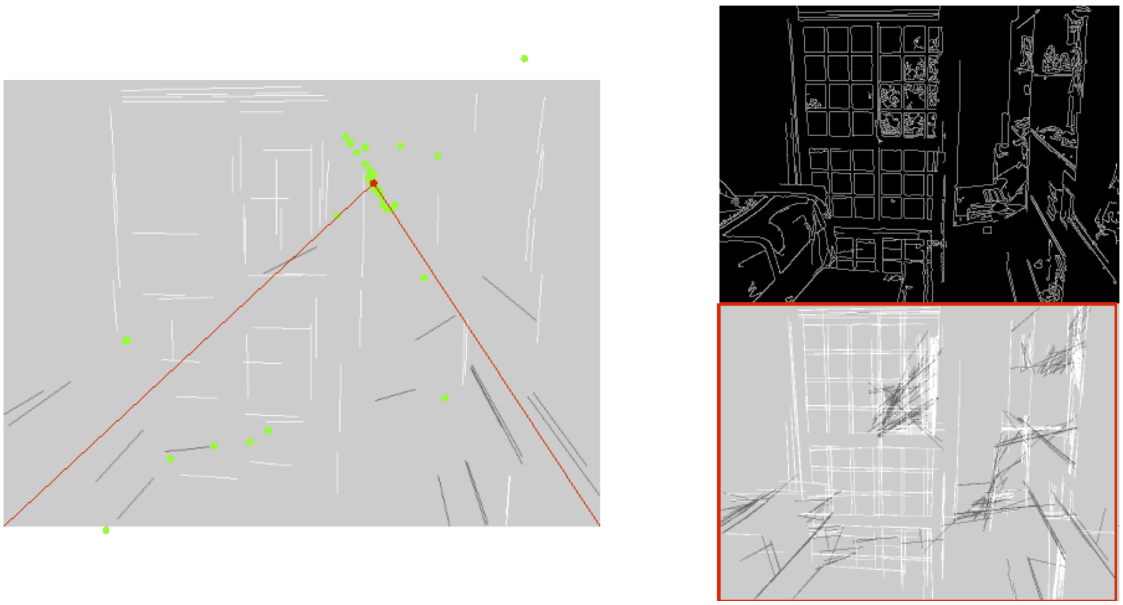


Figure 4.12: Simplified class diagram used to implement the vanishing point detection. The LineProcessing class centralizes the information, acting as a repository for all the data. Class VLine represents the detected lines.



(a)

(b)

Figure 4.13: Vanishing Point analysis for Figure 4.11(a). On the top right: Canny filter analysis. Bottom right: detected line segments, oblique lines are in black. Left: main line intersection points in green and main vanishing point detected using clustering nearest neighbors.

Each group must have a minimum number of points for the vanishing point to be considered. The vanishing point is the average position of the cluster points. Sometimes it may not be possible to find the three dominant vanishing points. If no vanishing point is found the algorithm tries the next line set from L with better score s.

4.3.5 3D World Orientation

The real world orientation (φ_2) depends on the direction in which the picture was captured. Currently we are considering that the Y axis (vertical) is fixed and the X and Z are free. The goal is to find the X and Z orientation of the scene in the photo. Most Manhattan world indoor scenes have parallel walls, the ceiling is usually parallel to the ground and corners are usually intersections of perpendicular planes.

Knowing the main vanishing points allows rotating the 3D virtual camera model in order to be aligned with the real world point of view. This means that when the depth of an object increases it will fade not to the center but to the direction of the vanishing point that corresponds to the Z axis. Furthermore, objects should look like they are parallel to the elements in the scene (i.e., walls). Using the main vanishing point (the one with a larger point cloud) the virtual world is rotated in two axis to align the objects with the scene. The rotations depend on the fiel-of-view (fov) and on the main vanishing point vp. $roty$ is the rotation around the Y axis and $rotx$ is the same around the X axis.

$$\alpha = tan(\pi \cdot fov/360) \tag{4.5}$$

$$diff_x = vp_x - middle_x, diff_y = vp_y - middle_y \tag{4.6}$$

$$rot_y = arctan(diff_x \cdot \alpha / middle_x) \tag{4.7}$$

$$rot_x = arctan(diff_y \cdot \alpha / middle_y) \tag{4.8}$$

The above are only valid for small rotations ($<\sim 45$ degrees) because for larger values the distortion is too large. The detection of the scene orientation (φ_2) is dependent on the success of the algorithm to detect the vanishing points (φ_1). Additionally, the virtual floor (φ_3) is assigned to be a plane in the world with a predefined negative y coordinate. This value is defined considering the average

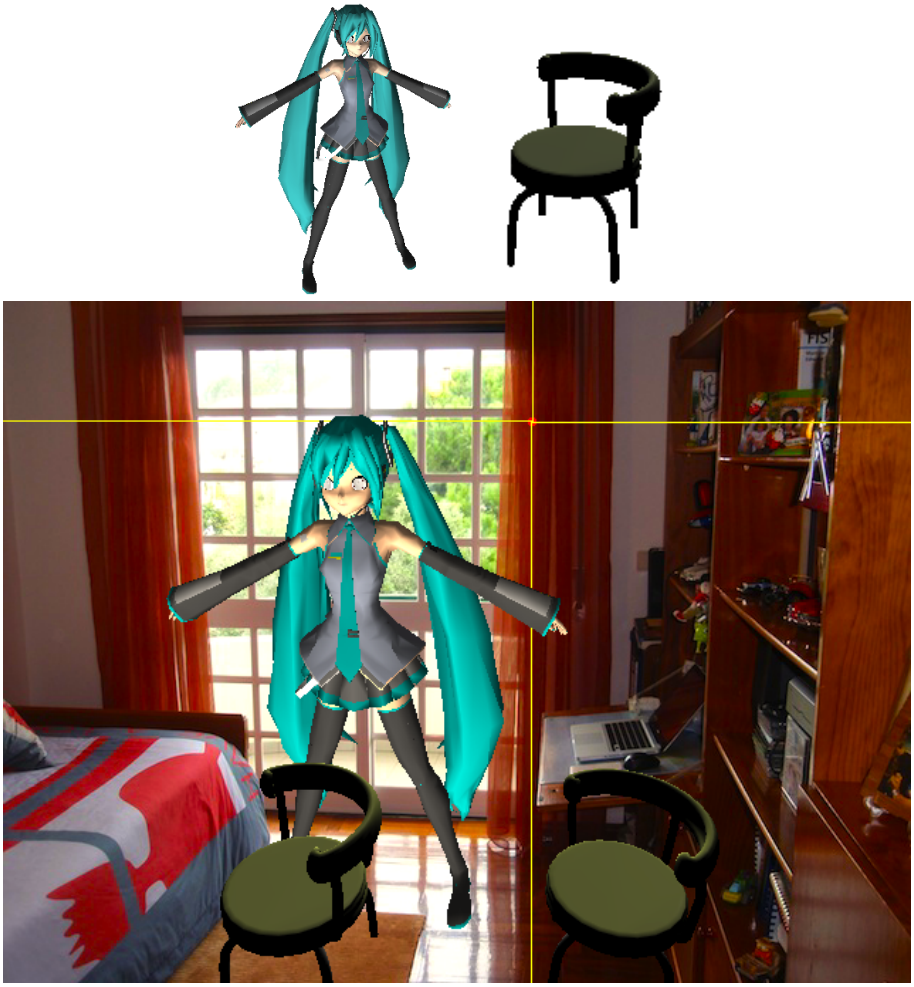


Figure 4.14: Based in Figures 4.11 and 4.13 the main vanishing point is discovered, the scene orientation calculated and 3D objects can be superimposed in the scene. The virtual objects are oriented in the direction of the vanishing point.

height at which people take photos, but can later be changed by the user in the AR tool (this will be further detailed in the next section). Using the scene orientation it is possible to introduce virtual 3D objects in the picture with the same orientation as the scenario (Figure 4.15).

The augmented reality (AR) framework takes into consideration the detected features vanishing points (φ_1), scene orientation (φ_2) and floor (φ_3). The current prototypes explore several possibilities that can be achieved in AR applications using the φ high-level features as seen in Figure 4.14.

Using the 3D oriented model M (introduced in section 4.2) it is possible to define a virtual floor where users can lay 3D meshes. The users just have to push the objects back and forth, or left and right, in a 2D model that allows this freedom. This is an important feature since not all users have the skills to correctly

Figure 4.15: Objects oriented with the scene. The yellow and green lines are oriented according to the X and Z axis. These are rotated so that the scene orientation φ_2 corresponds to the orientation of the photographed objects in the scenario.

visualize and place an object in a 3D world. Relieving the user from having to match the object with the floor and from having to orient the object to match the direction of the walls helps in the process. Figures 4.14 and 4.15 are examples of the 3D oriented world model with superimposed 3D objects that represent virtual furniture.

The introduction of virtual content should be done seamlessly, and the user should not be aware of all the processing that is involved in the application. In the user perspective, the interface should be effective, requiring only some simple instruction hints. The objects and drawings should be placed with the correct perspective.

4.3.6 Floor Definition

In the previous sections, the high-level feature Vanishing Points φ_1 (subsections 4.3.3 and 4.3.4) and Scene Orientation φ_2 (subsection 4.3.5) have been extracted. To complete the model M presented in section 4.2, the last high-level feature, the Floor φ_3, must be calculated.

In the proposed system, part of the problem has been solved by the user in the Acquisition phase (section 4.3.2). The goal of the high-level feature φ_3 is to answer to the following question:

Is the 3D virtual object on the floor of the photographed scene?

Using the floor mask $m_{\varphi 3}$, calculated in section 4.3.2, the problem of finding if a 3D object is hitting the floor can be divided in two parts. First, the elevation of the 3D virtual object must be equal to a given virtual floor. Secondly, the reprojection of the 3D coordinates of the base of the object, in the 2D screen, must be inside the floor mask.

The virtual floor is an *a priori* defined constant value, $virtualFloor_y$. The definition occurs in the initialization of the system in the interactive application (examples in Chapter 5) and is essentially dependent on the scale of the virtual objects that are going to be inserted in the scene.

The floor mask $m_{\varphi 3}$ is a 2D mask extracted from the original photograph. For interactive applications to recognize if a certain object is on the $virtualFloor_y$ the 3D homogeneous coordinates v,

$$v = [p_x, p_y, p_z, 1]^T \tag{4.9}$$

must be transformed in 2D window coordinates.

To do this several transformation matrices must be considered. Considering that the camera parameters C from model M are known, and the application resolution is know, the perspective projection model is a 4×4 matrix P. The modelview M_v is a 4×4 matrix acquired after the 3D world rotation described in section 4.3.5. Finally, the viewport $view$ of the application must be considered,

$$view = [view_x, view_y, view_{width}, view_{height}]\,. \tag{4.10}$$

The window coordinates w are obtained using the following equations (v' is a 4×1 matrix):

$$v' = PM_v v \tag{4.11}$$

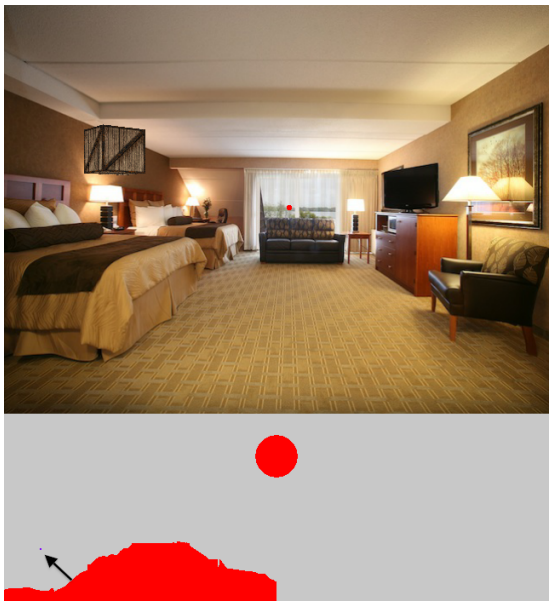
$$w_x = view_x + view_{width} * (v'(0) + 1)/2 \tag{4.12}$$

$$w_y = view_y + view_{height} * (v'(1) + 1)/2 \tag{4.13}$$

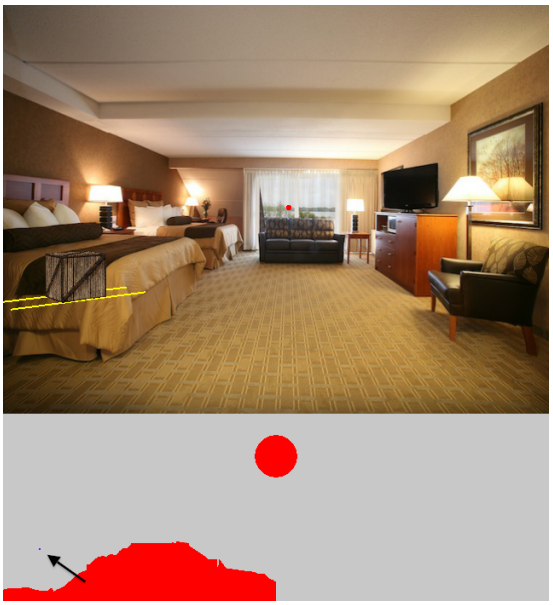
Answering the above goal question, the boolean variable $OnFloor$, which defines if an object is on the floor has the following behaviour:

$$OnFloor \leftarrow (p_y = virtualFloor_y) \wedge (w \in mask(m_{\varphi 3})) \tag{4.14}$$

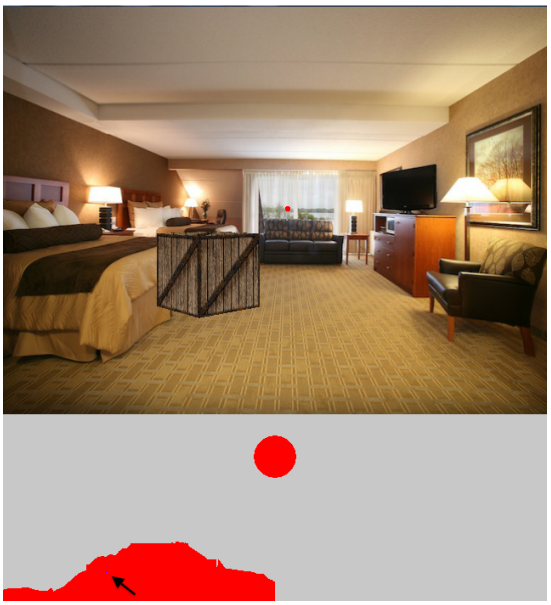
Considering Equation 4.14, Figure 4.16 presents all the possible logical states



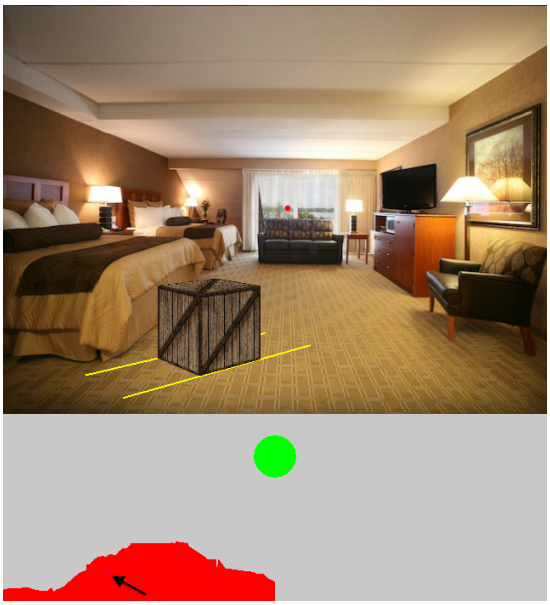
(a) $p_y \neq virtualFloor_y \wedge w \notin m_{\varphi 3}$



(b) $p_y = virtualFloor_y \wedge w \notin m_{\varphi 3}$



(c) $p_y \neq virtualFloor_y \wedge w \in m_{\varphi 3}$



(d) $p_y = virtualFloor_y \wedge w \in m_{\varphi 3} = true$

Figure 4.16: Defining if the virtual object is on the floor. The red area represents the mask of the floor, with the arrow pointing to the 2D window coordinates of the object relative to the mask. The yellow lines appear when the object hits the $virtualFloor$. The last image is the only example where the box is considered to be on the floor.

where a virtual object can be. Using this method presents several possibilities for mixed reality interaction. Virtual objects can be restricted to floor areas and places that are on or above the floor. This method can be extended for other planes, using other selection masks to define 3D restrictions. With the definition of the floor feature all high-level features φ previously presented in Figure 4.4 were presented and defined.

4.3.7 Implementation Details

The current framework, prototypes, examples and experiences were tested in a dual-core 2.53 GHz computer with 4 GB of RAM. Some usability tests presented in the Applications chapter (Chapter 5) were conducted in a tablet with a single 1.33 GHz processor with 4 GB of RAM. All webcams were used with a resolution of 640 by 480 pixels. Imported photos had different resolutions with a minimum of 640 by 480 but these larger images are resized to a maximum width of 640 pixels before the main line detection step.

The system is currently implemented in C++. The algorithms takes advantage of the OpenCV [Ope13] library, especially the data types, the Hough-transform implementation, several image filters and image descriptors. The interface and image/camera input is supported by the openFrameworks [ope13] library.

4.3.8 Extension to Video

The current mixed reality solution is implemented for single images. Since most augmented reality projects are based on real-time video, this question often arises.

Can this solution be extended to use video as input?

In this dissertation some examples of augmented reality were explored using video (section 3.4) and several images (sections 3.2 and 3.5 in Chapter 3). Other projects that work with video solutions (sections 2.1.4 and 2.2.2) are presented in the Related Work (Chapter 2).

A naïve direct application of the current approach for video is difficult because the vanishing point detection algorithm (section 4.3.4) is not optimized to run in sub-second time (as will be demonstrated in the next section). It would be difficult to continuously calculate in real-time (more than 15 frames per second) the lines and the vanishing points that support the scene orientation system. The best solution for a limited area is the one presented by the PTAM project [KM07]

where several points are tracked with the camera and a small plane is detected and tracked while the user is moving.

The nature of the interactive applications would have to be completely different. The current solution based on images has three well defined moments as explained before (section 4.2): Acquisition, Processing, Interaction. An offline video solution could be directly implemented by replicating these three moments for all video frames. A real-time video solution based on a camera is a continuous loop where these three stages have to be constantly executed.

Focusing on the real-time video solution, the user would have more freedom to move around the scene, but the amount of tracked features would have to be necessarily more limited. As an example, tracking a given area such as the floor, would mean tracking the features that define each point. The user would always have to be pointing to the scene, and the application should be prepared to loose focus of the scene and regain it very quickly. The augmented reality would probably be more engaging but less accurate because the user has to acquire and interact with the scene at the same time. This is exactly what happens in the Ball Invasion game[1], previously described in section 2.4.

Another approach for video solutions is to obtain a scene input that has more information than a single image, even if only a static scene is used in the end. This can be important to deal with occluding objects in the scene or choose which image is the best from a set of frames. The additional information from video [Sze96], can be 3D stereo vision [PH09], automatic panoramic reconstruction [Gau+12] or even rapid scene reconstruction [PAR11].

4.4 Tests and Results

In order to assess the reliability of the vanishing point detection φ_2 and scene orientation φ_3, two main studies were conducted. The first study presented in subsection 4.4.1 uses a previously classified dataset to measure the pixel error in the vanishing point detection. The second study, in subsection 4.4.2, presents a reliability test using several image datasets where on each image the vanishing points were detected and validated through a manual classification system. In the end, several considerations are presented in subsection 4.4.3 about the visual quality of the virtual object insertion in photographs.

[1] Ball Invasion, Smartphone game based on PointCloud and PTAM, `http://13thlab.com/ballinvasion/`.

4.4.1 Parameter Testing

The presented system is highly dependent on the correct detection of the image main vanishing point, φ_2, as explained in subsection 4.3.4. Before obtaining the main vanishing points, the image main lines have to be obtained. This is explained in subsection 4.3.3, where several candidate line sets are generated so that the algorithm can choose, which one is the best for vanishing point detection.

To obtain the main lines, e images are generated containing the edges of the image, and from each e image, h line sets are extracted. These different edges and line sets increase the level of redundancy by making the algorithm less prone to errors from illumination factors. The probability of finding a set of lines, which have a clear defined vanishing point, increases when there are more line sets from which to choose. On the other side, each line set calculated increases the processing time required. In the end there are $e \times h$ line sets from which to choose the best for vanishing point detection according to different parameters. The edges e are obtained using Canny filters with different parameters (e_1) and Erode and Dilate morphological operations (e_2). The line sets h are obtained using the Hough line segment detector again using different parameters (h_1, h_2 and h_3). The Canny filter and the Hough Transform were already described in section 3.6.

Variable Parameters

In this subsection, the goal is to find the optimal parameters and number of line sets which need to be generated to best detect the vanishing points. Table 4.1, briefly explains the multiple parameters which where tested to generate the edge images e (e_1 and e_2), and for each edge, to generate the h line sets (h_1, h_2 and h_3).

Table 4.1: Parameters to generate $e \times h$ line sets.

Parameters	**Description**	**Redundancy tests**	**configurations** i
e_1	Canny lower and upper thresholds	$\{(cl_1, cu_1), ..., (cl_i, cu_i)\}$	$[1, 5]$
e_2	Number of erodes and dilates	$i - 1$	$[1, 5]$
h_1	Line max. gap	$\{lg_1, ..., lg_i\}$	$[1, 5]$
h_2	Line max. length	$\{ll_1, ..., ll_i\}$	$[1, 3]$
h_3	Line threshold	$\{lt_1, ..., lt_i\}$	$[1, 2]$

The optimal detection parameters were obtained by testing several parameter configurations. The goal is to find the set of parameters that minimizes the $error$ of the mean Euclidean distance d between the detected vanishing points and the

annotated vanishing points from an image dataset.

$$d(\hat{V}_i, V_i) = \sum_{i}^{n} \sqrt{\left(\hat{V}_i - V_i\right)^2} \tag{4.15}$$

$$error = \frac{1}{n}\, d(\hat{V}, V) \tag{4.16}$$

In the former equations, V represents a vector with the real vanishing points and $\hat{V}$ represents the points detected by the algorithm presented in section 4.3.4.

The experience was performed using an external annotated database, which was used as ground truth for this experiment. The used image database was the York Urban DB[2] [CY99], which has 102 images with manually annotated vanishing points. These are urban images of buildings and lobby interiors, and most are Manhattan scenes. All the images have a resolution of 640×480. Since the implemented algorithm (section 4.3.7) reduces larger images to a maximum width of 640 the $error$ of Equation 4.16 was normalized by dividing the distance by 640. The results presented next use this normalized form.

The detector uses generic camera parameters C (section 4.2). This will result in some additional error but is a more realistic approach since the objective in this chapter is to create a generic detector suitable for many different cameras.

Experimental design

During development a long period of experimentation narrowed the possible values of the parameters to a small subset. Table 4.2 summarizes the experimented values for all the previously described parameters (Table 4.1 for description). As an example, when $h_2 = 3$, three different levels of maximum line length will be tested: 30, 60 and 90 pixels. The units in Table 4.2 are: grayscale values ([0,255]) for e_1 and h_3 and pixels for e_2, h_1 and h_2.

Using the configuration table (Table 4.2), three main detection tests were created using the configurations presented in Table 4.3. These are the most relevant combinations of configurations, each of them with a different purpose. Test A evaluates the number of edges that need to be created, Test B evaluates the line sets, and Test C takes the most relevant parameters from each block and tries to achieve a better combined result. Looking at Table 4.2, it is possible to observe that in the worst case scenario 750 different line sets will be calculated by the algorithm in order to detect the main vanishing point (considering the larger configurations, $5 \times 5 \times 5 \times 3 \times 2$). To keep the computational complexity low it is also important to consider in the results the number of operations required.

[2]York Urban Line Segment Database, `http://www.elderlab.yorku.ca/YorkUrbanDB/`.

Table 4.2: Different possible configuration for each parameter.

	Configuration i				
Parameters	**1**	**2**	**3**	**4**	**5**
e_1	(50,200)	(30,200) (70,220)	(30,220) (70,200) (100,200)	(30,220) (70,200) (100,200) (20,150)	(30,220) (70,200) (100,200) (20,150) (70,220)
e_2	0	1	2	3	4
h_1	10	7 12	5 10 15	2 5 10 15	2 7 10 15 20
h_2	60	60 90	30 60 90		
h_3	55	40 70			

Table 4.3: Configurations tested (see also Table 4.2).

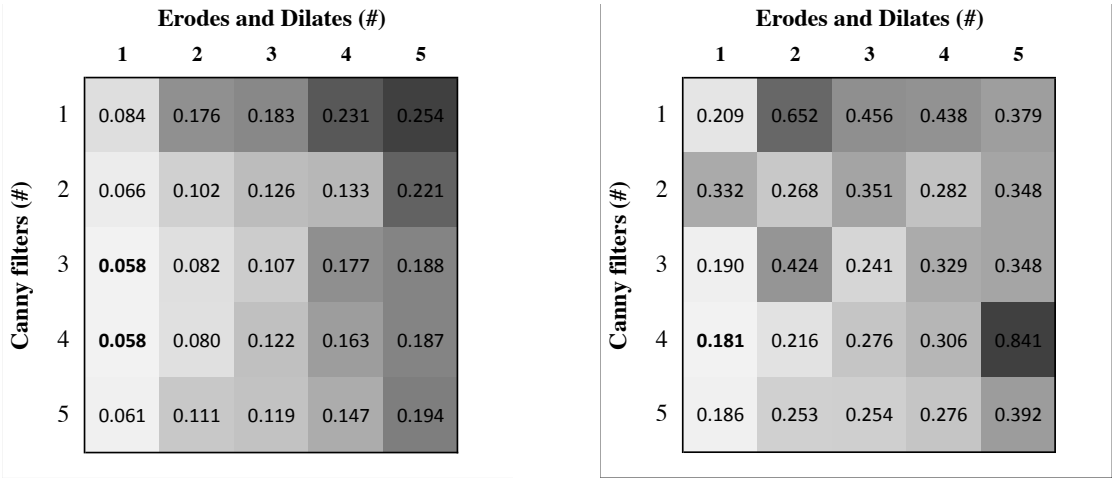
	Parameter				
Test	e_1	e_2	h_1	h_2	h_3
A	$\{1, ..., 5\}$	$\{1, ..., 5\}$	3	2	2
B	3	0	$\{1, ..., 5\}$	$\{1, 2, 3\}$	$\{1, 2\}$
C	3	70	$\{1, ..., 5\}$	2	2

Results

The results of the three tests are presented in Figures 4.17, 4.18 and 4.19. For each test the normalized $error$ is presented (where 1.0 means 640 pixels) in a grid combining two or three variables.

For each test, the results are presented with the mean normalized $error$ of the 75% images where the $error$ was lower (**error@75**) and the normalized $error$ of all the images in the dataset (**error@100**). The **error@75** measure gives a better view of the $error$ since it excludes the worst-case situations from the **error@100** where the $error$ can go up to high values. The measure **error@75** presents the error excluding the worst quartile of the sorted results for each image (visually perceivable in Figure 4.20).

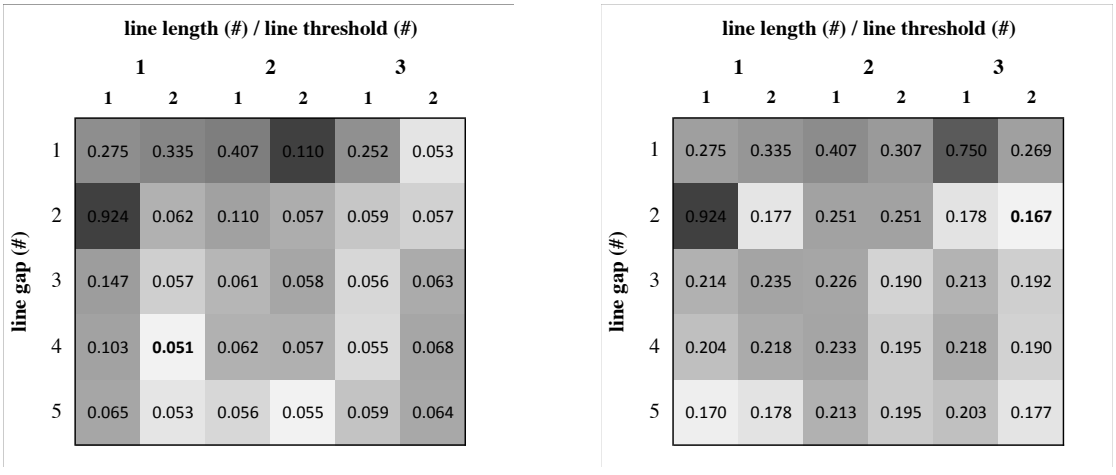
Test A results, seen in Figure 4.17, revealed that the detection improves when



(a) error@75 (b) error@100

Figure 4.17: **Test A**. Vanishing point detection error. Variables: Canny (e_1) *vs* Erode and Dilates (e_2).

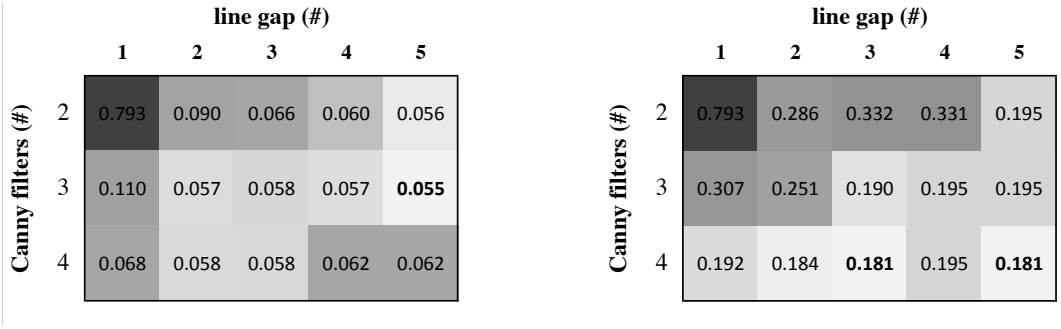
more Canny filters are used for redundancy. This detection improvement is especially significant in the configurations where $e_1 \leq 3$. After that the error does not decrease significantly while the processing time increases because of the extra number of filters applied. The main result from this test is that the Erode and Dilate morphological operations do not improve the detection. These operations are usually used to close gaps in the edges and reduce clutter, but probably with the line gap parameter used in the Hough Line detector, these Erode and Dilate operations do not contribute to the detection.



(a) error@75 (b) error@100

Figure 4.18: **Test B**. Vanishing point detection error. Variables: line gap (h_1) *vs* line length (h_2) *vs* line threshold (h_2).

Figure 4.18, presents Test B where the parameters of the Hough Line segment detector are tested. Increasing the number of line gaps tested improves the detection. Although the best results occur when $h_1 = 4$, these are not significantly higher than using $h_1 = 3$ while the later is less computationally demanding. In the h_2 and h_3 variables, no clear tendency can be observed with the best result in **error@75** going for $(h_2 = 1, h_3 = 2)$, and for **error@100** going for $(h_2 = 3, h_3 = 2)$.



(a) error@75 (b) error@100

Figure 4.19: **Test C**. Vanishing point detection error. Variables: Canny (e_1) *vs* line gap (h_1).

The last test, Test C in Figure 4.19, considered the most significant variables to calculate the edges (e_1) and the lines from the edges (h_1). These were the variables that more significantly changed in Test A and Test B. In this test, configurations 1 and 5 from (e_1) were excluded, the first for being largely prone to error and the second for being computationally heavy. For the Canny filter parameter (e_1) this test confirms that the best configuration is 3 or 4 and for the line gap parameter (h_1) the best results appear in configurations equal or larger than 3.

Discussion

Analyzing the three tests, considering the above trends and the computational cost of each configuration there are some aspects which can be concluded. For the Canny filter parameter (e_1) configuration 3 is the better one since it performs equally to configuration 4 but costs less computationally. One of the main conclusions from Test A is that Erode and Dilate operations do not add precision to the algorithm, in fact the best result indicates that they should not be used (configuration 1, $e_2 = 0$). This means that the vanishing detector should use the three redundancy Canny filter combinations defined in Table 4.2.

For each edge image, several line sets are calculated using the h_1,h_2 and h_3 parameters. In these parameters the results were not so expressively clear. The line gap parameter h_1, presented a consistent variation in Tests B and C, which

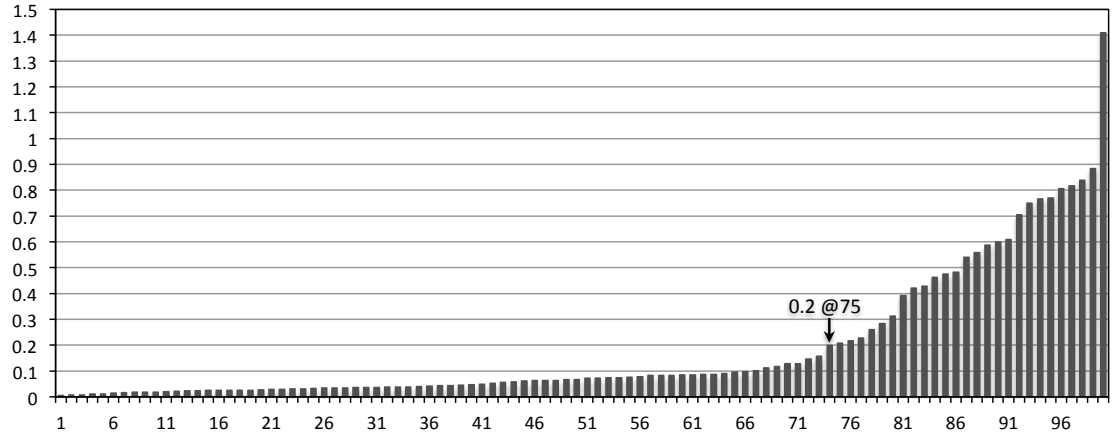


Figure 4.20: Sorted $error$ for each image in the dataset with the configuration: $e_1 = 3$, $e_2 = 1$, $h_1 = 3$, $h_2 = 2$, $h_3 = 2$. The **error@75** is 0.058 with the 75th image having an $error$ of 0.207 (Figure 4.19).

indicates that configuration 3 or higher should be used. The line length h_2 and line threshold h_3 parameters show mixed results with no clear pattern, although $h_3 = 2$ seems to prevail.

One of the most important conclusions from these tests, is that generating several edge images and several line sets, and only deciding which one is the best combination in the end, is the best strategy. Single combination configurations obtained most of the worst scores, while configurations with more combinations were more resistant to error.

Figure 4.20, presents the results from a configuration using the conclusions from these tests and experimental analysis using other datasets (subsection 4.4.2). The used configuration is $e_1 = 3$, $e_2 = 1$, $h_1 = 3$, $h_2 = 2$ and $h_3 = 2$. The chart represents the distance of the detected vanishing points and the annotated vanishing points for each image in the York Urban database. Figure 4.21 compares this configuration with the absolute best **@75** and **@100** in the tests from these section. These can be observed in Figure 4.18 were the minimum **error@75** is 0.051 and the minimum **error@100** is 0.167. The results show a slightly better result in the first images for **best@75**, but all configuration fail in the last quartile. Since there was no clear pattern between the h_2 and h_3 parameters, it is not clear if the combination used in **best@75** and **best@100** is better than the one used in Figure 4.20. For this reason and because it is more generic, and tests more combinations, the configuration presented in Figure 4.20 was used throughout the examples presented in the next subsection and in Chapter 5.

In the end, it can be observed that for Manhattan scenes such as the ones in

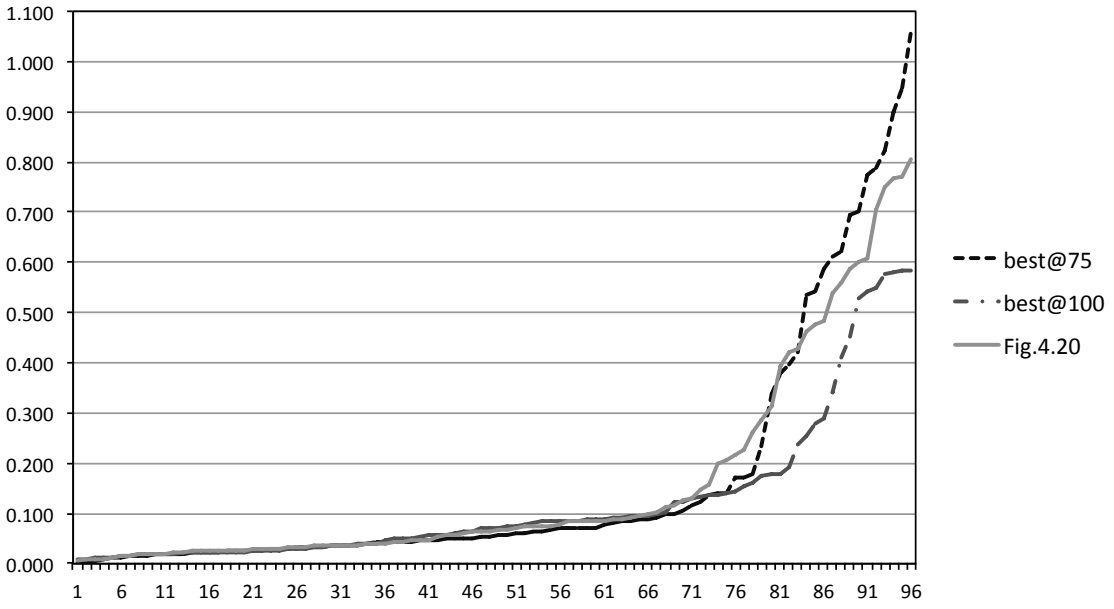


Figure 4.21: Comparing the best configurations **@75** and **@100** with the configuration from Figure 4.20.

the York Urban database the expected pixel error should be around between 0.05 and 0.08. For each chart, in Figures 4.20 and Figure 4.21, there is a residual error in each image which can be associated with the fact that this is an uncalibrated system with image distortions and different camera principal points. Taking that in consideration it can be observed that the detection is very stable in the first 75% images of the dataset.

4.4.2 Detection Reliability

In order to assess the reliability of the vanishing point detection [NC13a] presented in section 4.3.4, the algorithm was tested with several images from different datasets (Figure 4.22). In this study the main goal is to visually evaluate the reliability of the detected vanishing points. In the previous study the main focus was on the pixel error from the annotated and the detected vanishing points. In this study what will be evaluated is the visual coherence of the vanishing points. As an example, a vanishing point that is detected with a very acute angle (extremely to the left or to the right), can have a very large pixel error but the low angle error makes it visually coherent.

As stated before, the system relies on the visual information available in each image, especially orthogonal lines. Other factors must be taken in consideration,

such as image clutter, misleading lines and blurred images. Note that since the goal is to use this system in an interactive multimedia system, processing times should be kept as low as possible.

Experimental Design

To evaluate the detection system, many images were collected and tested. For each image the two major vanishing points were annotated and the processing time registered. After that, an implemented visual application was used to evaluate the results and its correctness. The application seen in Figure 4.23, sequentially presents all the images from the dataset with three lines. The red line represents the most probable vanishing point detected. This will be the main vanishing point used to signal the scene and virtual objects orientation. The green line represents the second most probable vanishing point. The blue (horizontal) line represents the horizon line of the scene. Depending on the confidence level of the detection, the system may only present some lines and in worst-case scenarios, none. It allows a user to manually analyse each image and classifies each image as correctly detected or incorrectly detected. The operation is quickly carried out using only two keyboard keys.

The algorithm was tested in four different datasets, each with its own characteristics (Additional information in Appendix C):

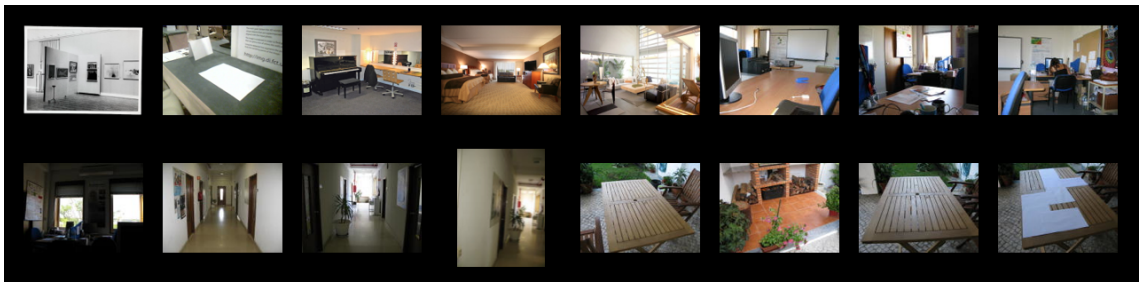
General: Mixed collection of photos used to test the algorithm, which includes indoor and outdoor photos, old, blurred and pixelated images.

Lab: Photos from the interior of a building and from the FCT/UNL faculty campus.

Flickr-Mix: Photos obtained from Flickr[3] with keywords such as "House", "House Interior" and "Buildings". It includes some landscape images with nature. Images which did not constitute a picture of a physical space were removed (e.g., animals or people).

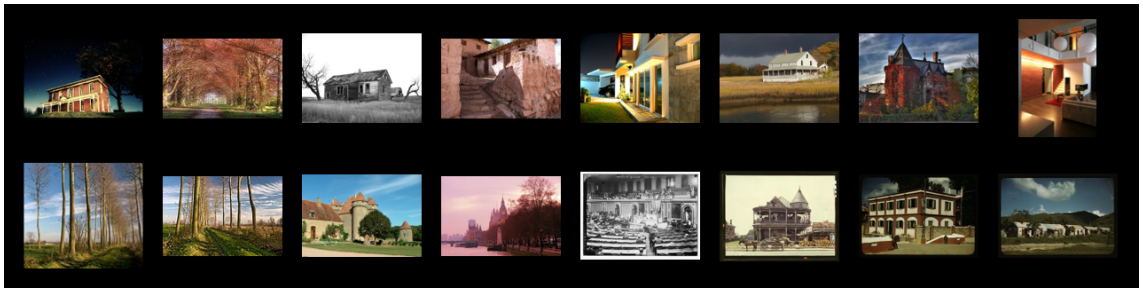
YorkUrbanDB: The York Urban Line Segment Database, presented in the previous subsection, is an online database used by Coughlan and Yuille [CY99], which contains images of urban environments consisting mostly of scenes from the campus of York University and downtown Toronto, Canada. Most images follow the rule of the Manhattan world. This is the same database already used in subsection 4.4.1.

[3]Flickr, Photo sharing website,`http://www.flickr.com`.



(a) General

(b) Lab



(c) Flickr

(d) York Urban

Figure 4.22: Sample from each dataset. Datasets a) and b) were locally built, dataset c) was built using the image storing website Flickr and dataset d) YorkUrbanDB is commonly used for algorithms exploring vanishing points. Additional samples can be seen in Appendix C.

(YorkUrbanDB: `http://www.elderlab.yorku.ca/YorkUrbanDB/`)

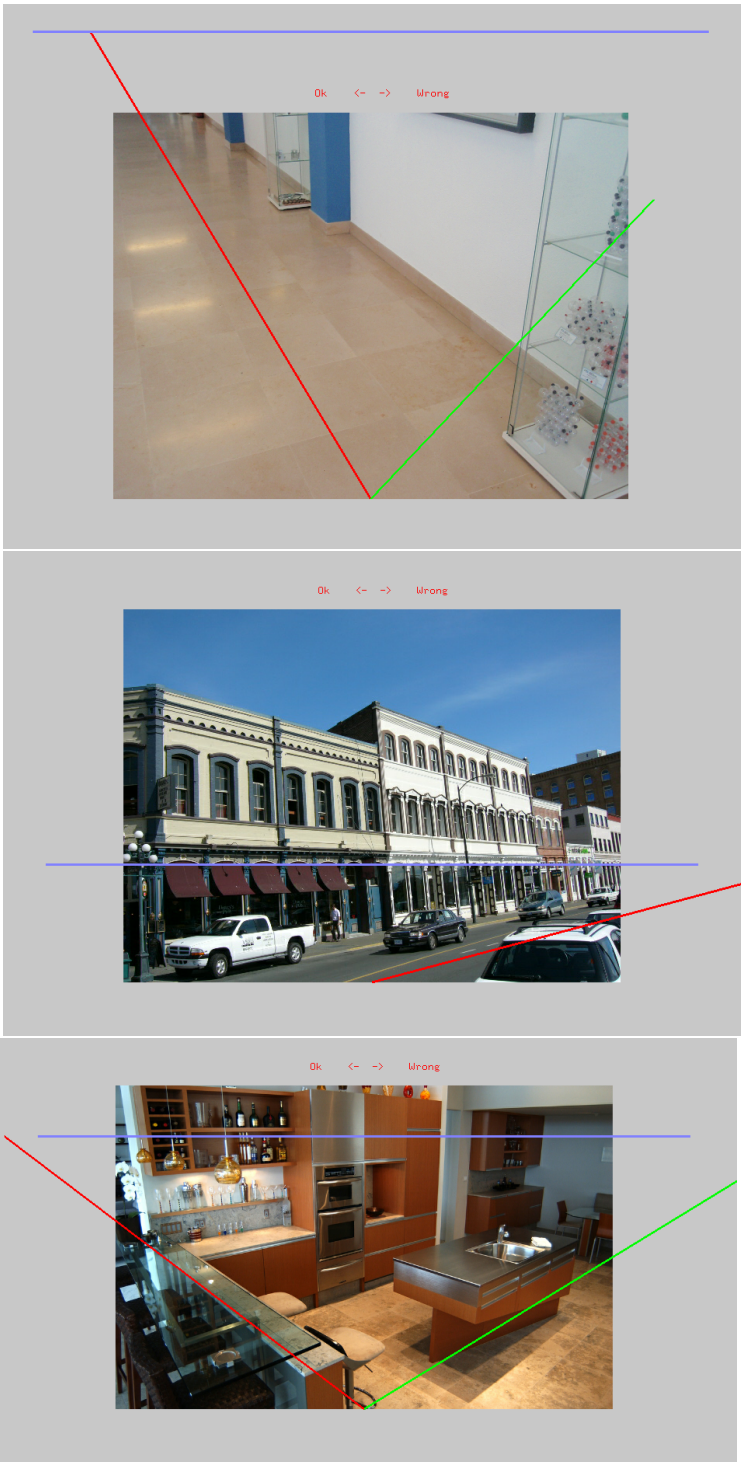


Figure 4.23: The system automatically detected the most probable main vanishing point (red line) and the second most probable (green line). The blue line represents the detected horizon line. The above images are screenshots of the manual application used to verify the correctness of the automatic detection algorithm.

Table 4.4: Scene orientation computational processing times and image resolutions for each dataset.

	Photos (#)	Time (s)		Resolution	
		Total	Average	Min	Max
General	53	59.61	1.12	640×480	4000×3000
Lab	35	54.95	1.57	2560×1920	
Flickr-Mix	317	909.6	2.87	450×450	7127×1729
YorkUrbanDB	102	193.62	1.90	640×480	
Total - Flickr	**190**	**304.92**	**1.62**		
Total	**507**	**1213.32**	**2.40**		

Results

The vanishing point detector was used in all images from all datasets. Table 4.4 presents the processing times for each dataset. The images were analysed in batch for each dataset. The implementation and computer used were detailed in the previous subsection 4.3.7. The results from Table 4.4 show that average time increases when the dataset has larger images. This happens because one of the first steps of the algorithm is to resize the images to a common factor (width of 640 as stated in subsection 4.3.3). Another factor, which seems to affect the average processing time is the size of the dataset indicating that there can be a performance degradation when there are more images.

Figure 4.23 presents some of the results from the vanishing point detection system. These results were manually classified two times using the previously described interface. In the first execution, Successful Detections were counted when the red line was pointing to a visible vanishing point, obtaining **S@1**. In the second run, Successful Detections were counted when the red line was pointing to a visible vanishing point **or** if the green line was pointing to a visible vanishing point thus obtaining **S@2**. The objective is to evaluate what is the visual success of the first guess and of the combination of first and second guesses of the system. This second guess will be used in the next chapter to suggest perspective alternative.

The results from the classification using the application presented in Figure 4.23 can be seen in Table 4.5 and Figure 4.24.

Discussion

The results show a high reliability of the algorithm with most datasets having a success rate between 72% in **S@1** and 83% in **S@2** (average combined success

Table 4.5: Main vanishing point detection rate for each dataset with the values for the first recommendation (S@1) and the accumulated values of the first and second recommendation (S@2).

	Photos (#)	**Successful Detections**			
		S@1		**S@2**	
		#	**%**	**#**	**%**
General	53	35	66.04%	44	83.02%
Lab	35	29	82.86%	34	97.14%
Flickr-Mix	317	152	47.95%	178	56.15%
YorkUrbanDB	102	73	71.57%	81	79.41%
Total - Flickr	**190**	**137**	**72.11%**	**159**	**83.68%**
Total	**507**	**289**	**57.00%**	**337**	**66.47%**

66.27%). As expected, the system worked better with Manhattan scenes, especially house interiors. Outdoor images of city buildings also perform well. This is the reason for the good results in the Lab and YorkUrbanDB datasets. The Flickr-Mix was not so successful mostly due to problems related with nature such as trees, landscapes, and modern and unconventional architecture. The General and Lab datasets have images of lobbies, floors, tables and spaces suitable for the presented AR prototypes. Since the main objective of this thesis is to create mixed reality applications in user spaces (**RQ**), most of the examples that perform well in this study are from places, which are interesting for mixed reality. Finally, one of the main conclusions of this study is that a second guess system increases the success rate by 10%.

These results show that, although not 100% reliable in all situations, the current algorithm works in enough situations to be useful in a mixed reality interactive system. To increase the success rate in a multimedia application, the user can be instructed to take pictures of places with enough line information or to use systems, as the one presented in section 4.3.1. Additionally, since the processing time is under 3 seconds, the user can always take or use another picture if the first one was not of good quality.

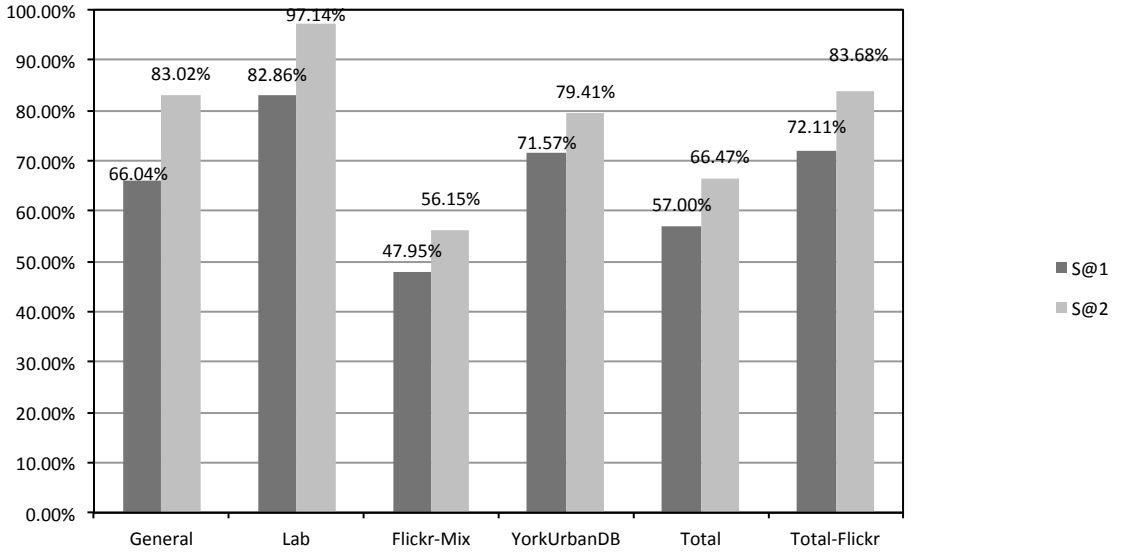


Figure 4.24: Main vanishing point detection rate for each dataset with the values for the first recommendation (S@1) and the accumulated values of the first and second recommendation (S@2).

4.4.3 Visual Quality Assessment

One of the main measures of success in any mixed or augmented reality interface is the quality of the immersion and the illusion that the virtual objects merge seamlessly with the real world. The proposed framework presented in this paper intends to build an internal model of the scene that can be used to overlay computer graphics on the real world image. For this reason, the goal is to create the perception of geometric correction and order in the virtual objects. The quality of the computer graphics and features such as shadows or lightning, although important to the overall user experience, are out of the scope of this work.

To summarize the results, Figure 4.25 and Figure 4.26 show examples of a real-time application where a wooden box is placed in each scene, aligned with the photographed world. Figure 4.25 presents results from the several datasets where the box was successfully introduced in the scene. Figure 4.26 presents some of the failures resulting from erroneous vanishing point detection or incorrect scene orientation. The box was chosen to better exemplify the three axis that compose the three-dimensional world, but any other object could be used.

Altogether most of the results were generally visually appealing with objects being correctly introduced in the scene with the correct perspective.

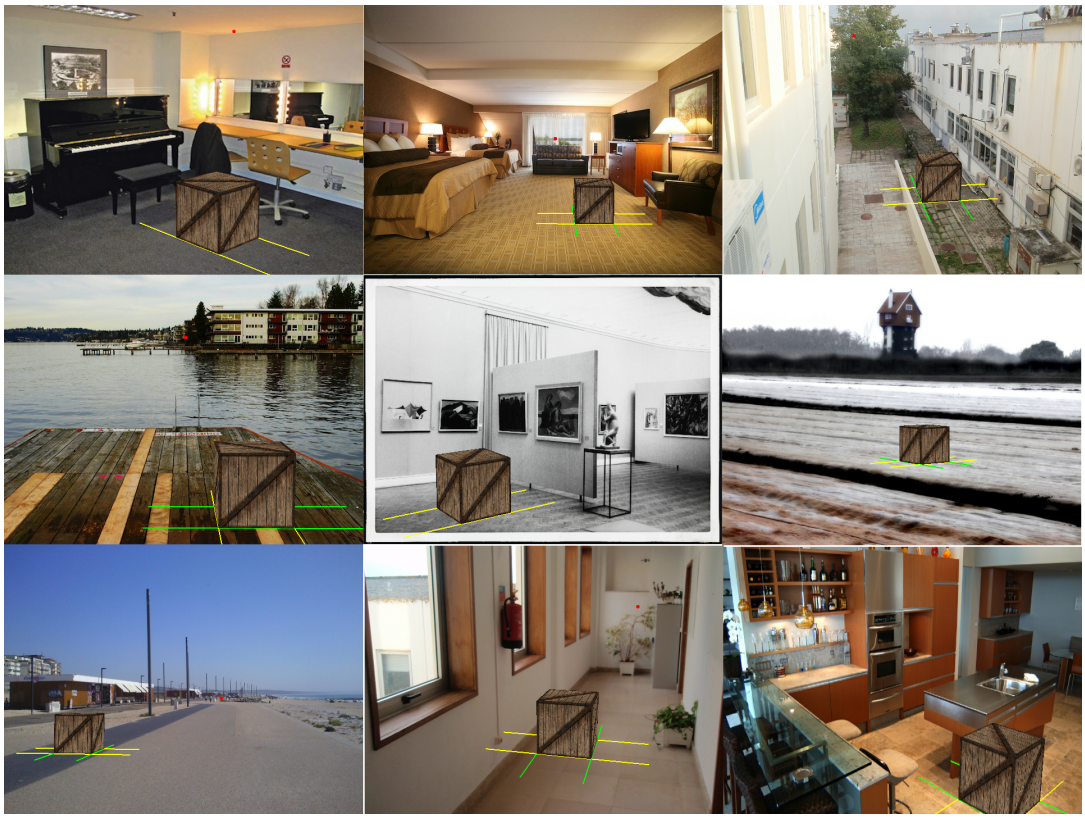


Figure 4.25: Examples of successful results. (Best seen in color).

Figure 4.26: Images with some detection problems. In some, the main floor plane was not correctly detected, in others the cube is not aligned (cube lines parallel to the image lines) with the world.

4.5 Discussion and Comparison

The previous subsections presented relevant results for the virtual insertion of objects in photographs, for the purpose of interactive applications. From the combined results of the parameter testing and detection reliability subsections (subsections 4.4.1 and 4.4.2), it can be expected that the scene orientation for Manhattan scenes will be correctly detected about 70 to 80% of the times (Table 4.5), using the algorithm proposed in this chapter.

Analysing other similar systems it is possible to perceive how the proposed system compares with other techniques. Probably the most similar projects, which use single image input, are Lee et al. [LHK09] and Karsch et al. [KHF11]. Although these projects have different objectives, goals and methodologies, there are common elements between them, which were summarized in Table 4.6. This table compares these two projects with the framework proposed in this chapter (first column [NC13a]).

Lee et. al [LHK09] try to recover the image structure by generating plausible interpretations of the scene based on the main detected lines. It creates several room hypotheses and then chooses the most probable model from the final set of hypothesis. The detection system is used to detect several planes and classify the orientation of each pixel.

Karsch et al. [KHF11] is a system to render synthetic objects in photos with correct illumination. It uses an automatic bounding box detection system proposed by Hedau et al. [HH09] to detect the main structure of the room. In addition, the user does some annotations to help the system detect surfaces and light sources. The result is a system where objects can be introduced in the scene with the correct illumination and reflectance.

The results from the algorithm proposed in this chapter were presented in subsection 4.4.1. For a 102 image external dataset (YorkUrban DB), the best results of the mean absolute error were **error@75** = 0.05 and **error@100** = 0.167.

The results from Lee et. al. [LHK09] and Karsch et al. [KHF11] are not directly comparable because different datasets were used, and the test goals were different, but give a different perspective of the success of other projects.

In Lee et. al. [LHK09] 54 indoor images were used to find the vanishing point and calculate the 3D model of the scene. The pixel orientation average error was 81% correct and qualitatively around 70% of the images return acceptable 3D models.

In Karsch et al. [KHF11] the geometric automatic bounding box detection

Table 4.6: Comparison table between the solution presented in this chapter and two others.

	Nóbrega and Correia [NC13a]	**Lee, Herbert and Kanade [LHK09]**	**Karsh, Hedau, Forsyth and Hoiem [KHF11]**
Single image input	✓	✓	✓
Additional input	floor annotation	–	surface and bounding geometry, and light annotation
Type of scene	Manhattan scene	Manhattan scene	indoor scene
What is detected	floor, vanishing points, orientation	vanishing points and 3D plane structures	image structure and illumination sources
Detection method	line analysis with delayed decision based on score (section 4.3)	line analysis with model fitting	estimating box layout [HH09]
Other publications	(section 1.5)	[Gup+11; LGH10; Xio+11]	[HH09; HHF12]
Superimposition of virtual objects	yes, with snapping and scene orientation	–	photo realistic

used the algorithm from Hedau et al. [HH09], which claims a pixel error of 0.183 using a 308 image dataset. Karsch et al. [KHF11] test the correct detection of texture reflectance achieving, for 80 images, a mean absolute error (MAE) of 0.141 and root mean square error (RMSE) of 0.272 (results @75 percentile). In a user study, 30 users were asked to choose which photo was more realistic, between a real photo and the same photo with a virtual element. The results show that 67% of the time users chose the image with a virtual object in it.

These numbers show that the obtained results are in line with other projects, even though the goal of each project is different. In this thesis the goal is to produce a fast algorithm with simple initialization, ready to be used interactively while in Lee et. al [LHK09] the goal is to infer the scene 3D model without annotations and in Karsch et al. [KHF11] the main focus is on photo-realism.

4.6 Summary

In this chapter a solution was presented to insert virtual objects in photographs captured by the users. The concept was introduced and motived in the beginning of the chapter and in the section Concept Overview (section 4.1).

The main problem considered was how to build mixed and augmented reality applications that take advantage of detected visual features in images of real world spaces as stated in section 4.2. The proposed solution is based on the detection of several high-level feature φ from low-level features δ. The process from which these features are obtained is detailed in the Implementation section (section 4.3).

The proposed system is based on a semi-automatic detection where the user takes a picture of a Manhattan scene and indicates the location of the floor. After that, the system finds the floor and scene orientation automatically, thus enabling interactive mixed reality applications where virtual objects are seamlessly introduced in the photographed scene.

The results presented in section 4.4 show a high degree of success in detecting the model in internal and external datasets. Additionally, throughout the chapter several extensions and comparisons are discussed. Subsection 4.3.1 suggests additional input workflows for other features and section 4.3.8 suggests a video extension to the proposed system. The next chapter (Chapter 5) is mostly dedicated to applications and user studies of this system and finally section 4.5 compares this framework with other known similar projects.

This framework presents a viable solution to the problems proposed in the

Reseach Questions section (section 1.1). Namely it answers to **RQ1** by providing a solution to the creation of mixed reality applications where virtual objects interact with a non pre-defined real world scenario. More importantly the algorithms presented in this chapter can be repeated and the results presented in section 4.4 can be replicated by using the same datasets.

5

Applications

In the previous chapters, several techniques and algorithms were studied, presented and tested. Chapter 3 summarizes important key technologies and how to use and take advantage of them. The following chapter, Chapter 4, takes additional steps and proposes an integrated solution for the insertion of virtual content in photos. This chapter presents several applications that instantiate the ideas proposed in the previous chapters with concrete examples.

Using the previously explained techniques and algorithms three applications were implemented. The goal is to evaluate if the proposed algorithms can be useful in interactive applications by conducting user studies. For each of the following applications the concept is proposed, followed by the interface description and the user studies results. The three main applications are:

Magnetic Augmented Objects in Photographs This was the first approach to implement the concepts proposed in Chapter 4. It is a system for computer-aided insertion of virtual objects in photographs with the help of virtual guidelines and an artificial snapping system. These tools help less expert users to virtually position 3D objects in images.

Mixed Reality Snake Game This is a game based on photos taken by the users. The goal is to play a mixed reality game in a real world space chosen by the user. In the game, two users control a virtual snake that moves along the floor, bumping the walls and obstacles. This system assesses the viability of creating the applications using the framework proposed in Chapter 4.

Past Museum Exhibition Navigation Through Overlapping Images The researched techniques presented in Chapter 3 have several practical applications in fields that use images. This application is a museum navigation system based on overlapping images. The relation between images is defined by several pre-processed homography matrices (section 3.2). As a case-study, the system was implemented in the context of the exploration of a modern art exhibition from the 1950's. The goal was to create a contextually dynamic interaction with the pictures from an exhibition that does not exists anymore.

Each of these applications has the main goal of adding interactivity and virtual content to photographies. The first two applications (sections 5.1 and 5.2) use a single user-supplied image, while the third (section 5.3) presents an example where an image dataset is provided.

5.1 Magnetic Augmented Objects in Photos

Using the developed framework explained in Chapter 4, an application was developed [NC11; NC12c; NC12d; NC12e] to take advantage of the proposed system for virtual insertion of objects in photos. The main contribution of this application is the introduction of the magnetic augmented objects concept [NC12d]. These objects are aware of the properties of the photographed scene, and react to them as if they were in the real world. These properties include the existence of floor or walls, the orientation of the scene, the vanishing points or the existence of obstacles. The tools discussed in this application could be useful in interior design applications, architecture or augmented reality games.

The current prototype proposes several design tools such as *snapping* and *glue*, which help the user, to introduce objects in photographs. These tools are based on the detected features. The mixed reality framework (presented in Chapter 4) takes into consideration the detected features vanishing points (φ_1), scene orientation (φ_2) and floor (φ_3). The current prototype explores several possibilities that can be achieved in mixed reality applications using φ high-level features.

Using the oriented model M (introduced in section 4.2), it is possible to define a virtual floor where users can lay 3D meshes. The users just have to push the objects back and forth, or left and right, in a 2D plane. This is an important feature since not all users have the skills to correctly visualize and place an object in a 3D world. Relieving the user from having to match the object with the floor and from having to align the object to match the direction of the walls helps in the process.

In the previous chapter, Figure 4.15 presented a preview of a 3D oriented world model with superimposed 3D objects that represent virtual objects. Figure 5.1 summarizes the concept of the magnetic augmented objects.

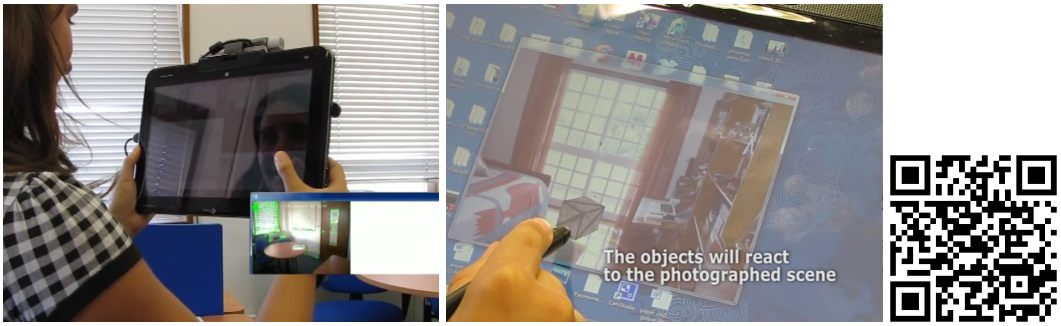


Figure 5.1: Magnetic augmented objects in photo applications.
(Video: `http://img.di.fct.unl.pt/rpn/phdthesis/magnetic.mp4`)
(instructions in Appendix B).

5.1.1 Interaction Concept

In order to take advantage of the framework for virtual insertion of objects in photos, different applications can be implemented, especially for areas such as computer aided design (CAD), furniture positioning and point and click editable spaces.

What all these applications have in common is the virtual transformation of a real space using only cameras. The main objective is that someone with no real experience with geometry and 3D manipulation can achieve this virtual transformation. The user is encouraged to take one or several pictures of an indoor scenario, with a standard camera or using a webcam, smartphone or tablet, as detailed in Figure 5.2. After that, the interaction takes place in the computer or tablet. Considering as an example the context of an interior design application, the user can experiment with several virtual objects (e.g., furniture), testing them in different positions.

The core application, the magnetic augmented objects, is presented in the next subsection. The concept and how it is implemented is explained along with the required interface. The following subsections explore in depth practical applications of the magnetic objects. Subsection 5.1.4.1 explores the possibility of creating a furniture testing application. Subsection 5.1.4.2 presents an alternative approach to positioning objects in 3D, which makes the attachment of objects instantaneous. In the end, a user study (subsection 5.1.5) compares the presented

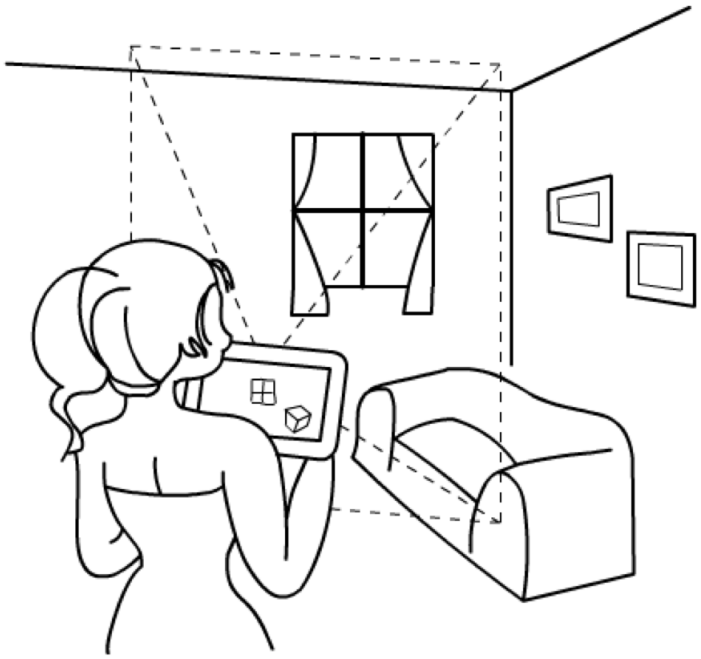


Figure 5.2: The user takes a photo of a space using a device such as a tablet with a camera. Using the photo it is possible to add virtual augmented objects that react to elements of that image, such as floor or room orientation.

techniques and evaluates the feasibility of user editing through the use of magnetic objects.

5.1.2 Magnetic Augmented Objects

Virtually reshaping a known space can be a troublesome task because it usually requires the creation of a 3D model of that same space in order to introduce and visualize new objects and 3D content. Using an image of the space, captured with any camera the user has available, may simplify the process. The problem is that most users do not have the required skills to correctly place objects in a manner which is visually and geometrically correct and appealing.

The concept of **Magnetic Augmented Objects** [NC12d] is used to describe virtual objects that react to the properties of the **Photo** where they are being introduced. These objects are attracted to the floor, react to the orientation of the scenario and can sense when they are sitting on existing volumes. These apparent properties of the magnetic objects are entirely based on the previously discussed high-level features, which are detected from the image.

Magnetic objects have several constraints that help the user to position them in three dimensions. They possess a *snapping* property which makes the objects

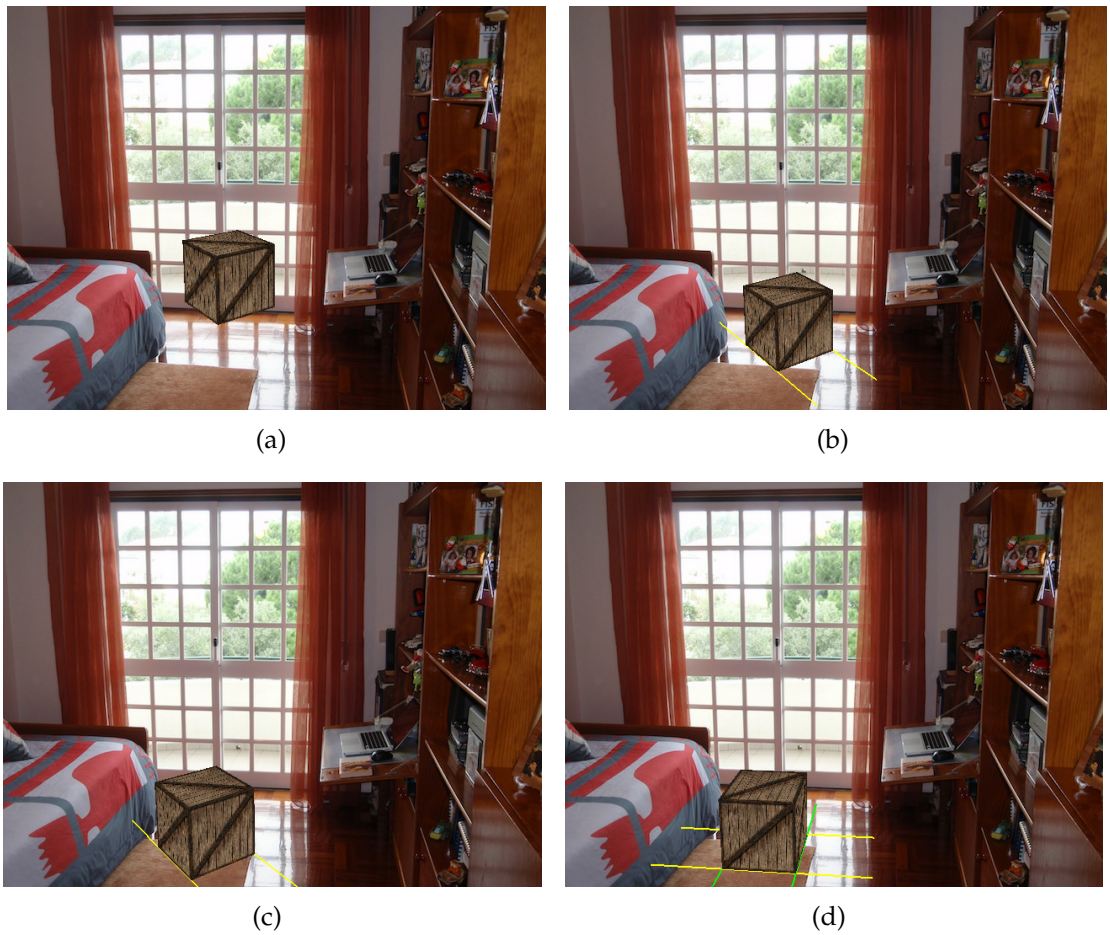
Figure 5.3: Example of the introduction of virtual objects that are aligned with the scene and snapped to the floor.

attach themselves automatically to the floor when they are close to it. Additionally, the objects also snap when the rotation of the magnetic object is close to the orientation of the scenario (assuming a Manhattan World [CY99]). To visually give feedback to the user that the object is attached to the floor, two yellow lines mark the intersection between the magnetic object and the floor (Figure 5.3). If the object is aligned with the scene then two green lines appear signalling the alignment. Using the scenario floor definition described in section 4.3.6 it is possible to say if the object is above an area where it should not be. Figures 5.3 and 5.4 illustrate a magnetic object being positioned in 3D with the help of the *snapping* tool.

Adding the floor and alignment *snapping* tools helps the user to position objects in three-dimensions because it creates a structured environment where the image can be used as reference to position the virtual mesh. As an example, Figure 5.4 presents a simple mouse drag and drop interface where the user has two dragging options and one rotate option. The dragging options are: moving along the XZ plane and moving along the ZY plane. The rotation was limited (only for simplicity reasons) to rotation in the Y axis. The user chooses the desired operation by pressing one of the mouse keys (left, middle or right) and dragging (vertically and horizontally).

Considering this minimal interface the user tackles the three translation dimensions and the three rotation dimension separately using as reference points, the floor (XZ plane) and the alignment of the scene. Without these references the object would appear to be floating in the air leaving the user with no idea about the actual depth of the object.

As an interaction example, the user can drag the object down until it hits the floor (using only the Y axis), then move the object freely in the floor plane (X and Z axis) to the desired location and afterwards complete the task again with



(a) (b)

(c) (d)

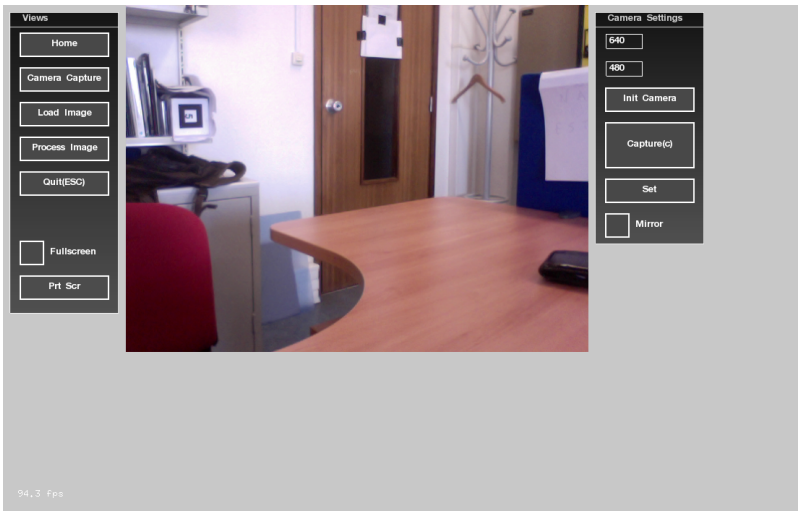
Figure 5.4: Positioning a magnetic object in 3D (best seen in color) (b) the yellow lines indicate that the object has touched the floor, (c) and can be moved freely in the 2D plane of the floor. (d) By rotating the object green lines will appear every $90°$ when its rotation is aligned with the scenario. Whenever the object is close to the floor it automatically attaches itself to the floor. The same way, whenever the rotation is close to the rotation of the scene the object aligns itself with the scene.

the desired height (again using the Y axis). With the *rotation snapping*, the object can be aligned with the main lines of the scenario. This is especially interesting to position objects in an organized fashion in the scenario, mainly on the floor, on the walls or other 2D planes. It is also important for virtual furniture applications such as the concept introduced in the next subsection.

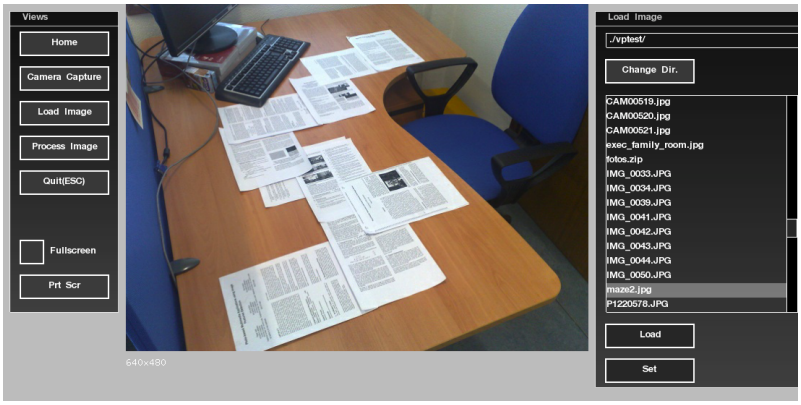
5.1.3 Interface and Implementation

To study the feasibility of the magnetic objects concept a multi-purpose prototype [NC12a] was built and tested. The application allows the capture, processing and interaction with an image. It has all the required elements to test the use of the

(a) Main menu



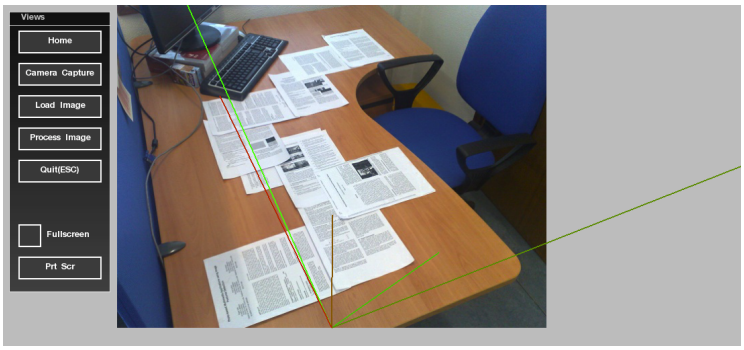
(b) Camera Capture



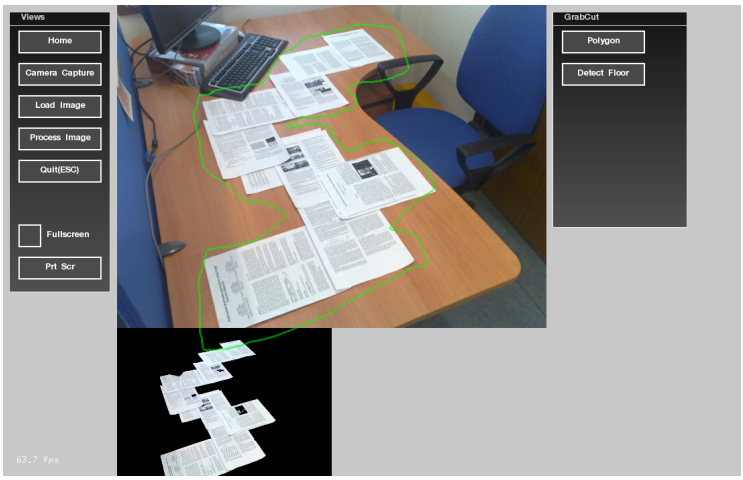
(c) Load Image (from disk)

(d) Process Image

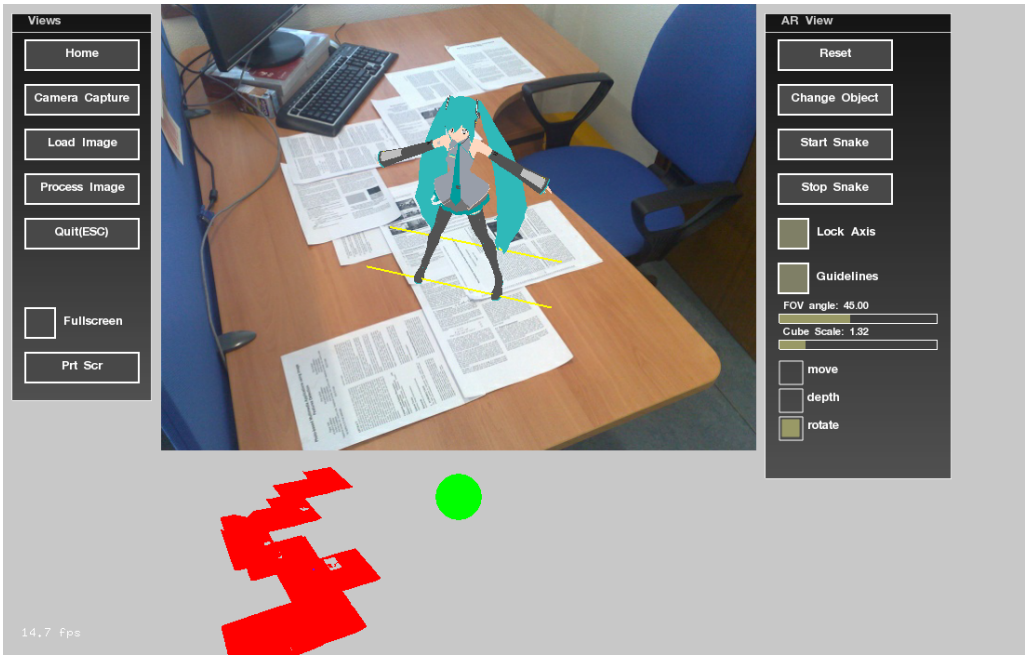
Figure 5.5: Magnetic augmented objects application screenshots. Menu structure and how to load the image.



(a) Detected Vanishing Points



(b) Floor Selection



(c) Mixed Reality

Figure 5.6: Magnetic objects application screenshots. The three main views for, (a) and (b) detect features, (c) is the mixed reality interactive part of the application.

magnetic objects in a laboratory environment. For a final user application, several features would have to be simplified, as will be demonstrated later in other prototypes (section 5.2). The implementation is entirely based on the framework described in Chapter 4, specifically in section 4.3. Since this is a case-study prototype all the detection steps described in the previous chapter do not occur automatically, they are manually controlled and allow some degree of fine tuning and experimentation.

The application, presented in Figure 5.5 and 5.6, is a desktop application using the mouse (or in a touchable version, the finger) as input and a webcam (with known parameters). It follows the same specifications and it is implemented with the libraries described in subsection 4.3.7. It was implemented using the openFrameworks library, with an EasyGUI user interface.

The interface has a three-level graphic layout that starts with an always visible options menu, presented in Figure 5.5(a). In this menu, the user has three main view options: capture, load and process images. The first two options, "Camera Capture" and "Load Image", are used to acquire an image. The camera capturing system (Figure 5.5(b)) shares the principles described in subsection 4.3.1. It has a submenu, which allows to change the camera, its parameters and toggle the mirror function. The loading image screen (Figure 5.5(c)) can be used to import any image from the disk. The image must have a minimum resolution of 640 by 480 pixels. To speed up processing, larger images will be resized to a more manageable size (e.g., maximum width 800 with height depending on the aspect ratio).

The third main option, "Process Image" presents the options menu shown on the right of Figure 5.5(d), which has several options to manipulate the image. The simplest option is "Set Vanishing Point" which allows the user to manually indicate where the main vanishing point is by clicking in the picture. This is just a testing feature for debug purposes. The other three buttons open up three different screens shown in Figure 5.6: "Detect Vanishing Points", "Detect Floor" and "Augmented Reality".

Pressing "Detect Vanishing Points" will lead the application to the state depicted in Figure 5.6(a). Here the algorithm presented in subsection 4.3.4 is applied to detect several candidate vanishing points, each with a different degree of confidence. In Figure 5.6(b), each vanishing point is represented by a line connecting the center bottom of the image to the respective vanishing point (the point may be outside the image). The degree of confidence is represented by the color with bright green representing the highest confidence and the bright red representing

the lowest. This representation allows a quick visual feedback of the correctness of the detection because the green line should have the same vanishing point as most lines in the picture.

Selecting "Detect Floor" on the "Process Image" view, will return the Figure 5.6(b) view where the user can select the floor by using a free selection lasso tool or a rectangle tool. This will classify a certain area of the image as "floor", as explained in subsections 4.3.2 and 4.3.6.

The last available button on the "Process Image" view, is "Augmented Reality", which opens the screen in Figure 5.6(c). This is the main screen of the application. It introduces the magnetic virtual objects in the photo. Preferably it should be called after the previous two screens. The right menu allows several experimentation options. The "Reset" button puts the virtual object on a random position. The "Change Object" cycles through several pre-loaded *.3ds* meshes, which can be introduced in the image. The "Start and Stop Snake" introduce a virtual mixed reality game on the scenario, which will be further discussed in the next section (Section 5.2). "Lock Axis" and the "move", "depth" and "rotate" are used to move the virtual object with the mouse in 3D. When the axis are locked a more restrictive moving approach is implemented. The "guidelines" button toggles the lines indicating the magnetic *snapping* explained in the previous subsection in Figure 5.4. Finally the "FOV angle" allows the user to change the default field-of-view of the camera and "Scale" changes the scale of the object. On the bottom of Figure 5.6(c) it is indicated if the object is on the floor or not, according to the definition in section 4.3.6. On this last screen, it is possible to observe the direct application of all detected high-level features from the image in an interactive virtual object positioning system.

The importance of this application is that it created a testing environment, which allowed the fine tuning of the algorithms presented in Chapter 4, and the experimentation with different concepts and users. This will be detailed in the following sections.

5.1.4 Application Concepts

In this subsection, two application concepts are presented: an interactive furniture testing application (with implementation, subsection 5.1.4.1) and a stereo vision glue method (alternative to the Magnetic Objects, subsection 5.1.4.2).

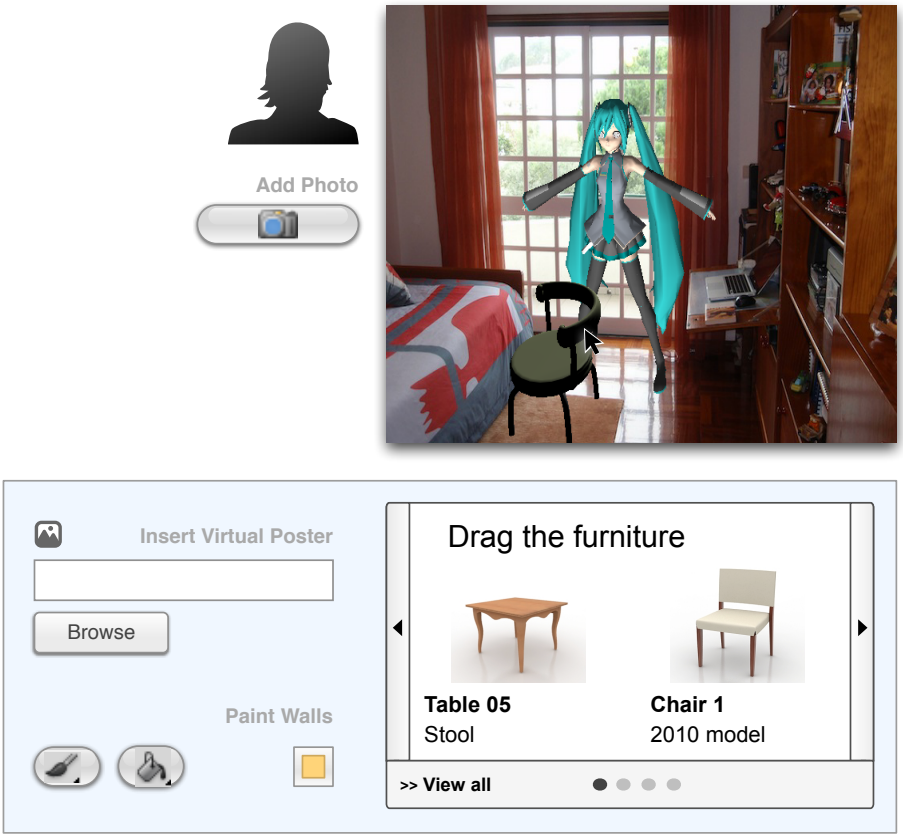


Figure 5.7: Wireframe concept for the interface. The user adds a photo of his room using the top button. The image is then displayed and can be edited with virtual painting and virtual objects such as furniture.

5.1.4.1 Interactive Furniture

One of the possible applications for the magnetic objects concept is the creation of an interactive furniture testing system. The goal is to create an application that allows the user to experiment with virtual furniture models in a picture of the user's home.

Designing and redecorating a room [NC11] is a difficult task for many people. Choosing colors, furniture, something that reflects what someone is or what a room is for. This system would provide a simple way to evaluate if something is appropriate for a physical space.

The virtual furniture should attach itself to the floor and be simple to align with the scenario. Moving the objects should take into consideration the properties provided by the photographed space.

The interface concept presented in Figure 5.7, uses real photos as background

for the creation of an indoor design. Many other systems, such as Atelier Pfister[1], allow the experimentation of furniture and different materials but they use predefined scenarios that sometimes have nothing to do with a real room. Others depend on special markers (e.g., ARtoolkit markers [ART03] or special images [Met13; Vuf13]) on the photograph to acquire a plane (e.g., DesignMyRoom[2]). Figure 5.7 shows an interface proposal for the room editing concept. The user takes a photo and use it as input. Then there are several options to alter the image. The idea is that the physical structure of the room affects the way it is painted or how virtual objects are placed. Painting a wall or placing an object should take into consideration the perspective of the scene. Objects are dragged from the bottom menu and moved around with the mouse or finger considering that there is a floor and gravity.

This kind of interface allows a person to see if a given piece of furniture from a store looks good in their house, thus helping in the choice decision process. In the user perspective, the interface should be effective, with only some simple instruction hints.

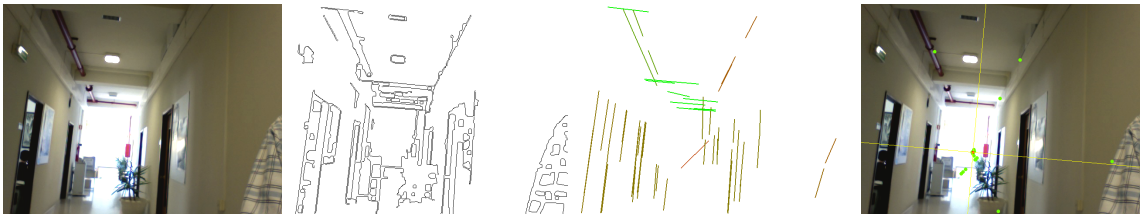
To implement the concept proposed in Figure 5.7, the application should have a simplified user interface based on the Magnetic Objects testing application (previous subsection, Figures 5.5 and 5.6). This version should detect the vanishing point automatically and have input strategies to obtain the selection of the floor, as discussed in subsection 4.3.1.

Using the framework discussed in the previous chapter (Chapter 1) to insert virtual objects in photos, a prototype was implemented to test furniture objects in photographs. The application presented in Figure 5.8, is a minimalist version of the Magnetic Objects application. It was conceived to be operated with the mouse or touch using a tablet. The input is captured with a camera or loaded from disk. The processing phase presented in Figure 5.8(a) automatically discovers the room orientation. The furniture presented in Figure 5.8(c) is glued to the floor and oriented with the room and can be easily dragged around with the mouse or finger. This prototype only deals with geometrical correctness, additional work is required to match lightning, shadows and textures.

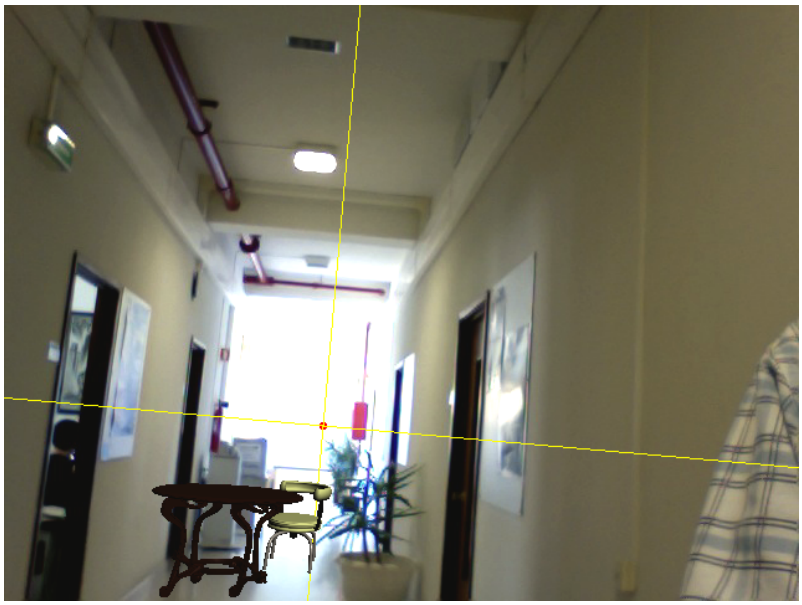
Furniture testing applications based solely on a photographed image using this technique are a possibility, especially in home environments that fit the Manhattan World model. These kind of applications are appropriate when the user:

[1]Atelier Pfister, smartphone application for furniture design, `http://www.atelierpfister.ch/app`.

[2]DesignMyRoom, Interior design application, `http://designmyroom.com`



(a) Loading and processing image.



(b) Main vanishing point.

(c) Interacting furniture.

Figure 5.8: Furniture testing application screenshots.

Figure 5.9: Automatic *glue* system for virtual objects. The user clicks on the image and the object is automatically laid on that spot. The right part of the image shows the detected 3D structure (based on two near images) that supports the interaction.

(1) has no direct access to a physical place, (2) has no experience or has no access to special markers (difficult to print a marker) or (3) does not have a specific hardware (no 3D camera or camera with accelerometer).

5.1.4.2 Stereo Vision Glue

Until now all the Magnetic Objects concept has been entirely based on the solution presented in Chapter 4. This concept can be extended to take advantage of other techniques, such as stereo vision presented in section 3.5. This means that the magnetic objects can be used in solutions with two or more photos. Many application, based on the PTAM system [KM07] already take advantage of this technique to extract the depth of the image by triangulation of features between images or video frames.

This type of solution, has the advantage of obtaining additional details in the 3D structure of the scenario when compared with the single photo solution discussed in Chapter 4. The drawback is that the input is more complicated for the user since several photos or a video are required. In many solutions [KM07; PH09], the user has to be trained to take photos/videos sideways, which increases the learning curve.

To compare with the single photo framework presented in this thesis, a stereo

(a) (b)

(c)

Figure 5.10: Interacting with the yellow poster. The user clicks on three different areas of the scenario.

vision prototype using two photos was implemented. The prototype, depicted in Figure 5.9, is a touchable/mouse application, seen previously in Figure 4.1. Much like the Magnetic Objects application (subsection 5.1.3) the goal is to improve the speed of how users introduce virtual objects in the photographed scene. Since the stereo vision solution can detect most of the 3D structure of the scene, objects can just be pinned directly on any surface (and not just the floor or plane surfaces). The prototype presented in Figures 5.9 and 5.10 introduce a new tool called *glue*, which can pin the virtual objects automatically in some point of the image. This is an alternative to the floor *snapping* system presented in the Magnetic Objects concept (subsection 5.1.2), although the rotation *snapping* can be used side by side with *glue*. Figure 5.10 presents an interaction example where the user introduces the object in several locations of the image, and the object is automatically positioned with the correct scale on the underlying surface with the direction of the

normal vector of that surface.

The implemented prototype requires the use of two images captured very near one from the other with horizontal displacement only. The goal was not to propose a new approach to 3D reconstruction systems based on multiple images, but to implement an alternative solution for the single image detection framework presented in Chapter 4. Having two solutions allows to compare these implementation with user studies. The next section presents the results of an user study comparing the stereo vision *glue* method with the magnetic augmented objects *snapping* system.

5.1.5 User Study

In the prototype described in subsection 5.1.2, the magnetic objects were defined with the floor and rotation *snapping* tools. In subsection 5.1.4.2, the concept of stereo vision *glue* tool was presented with a prototype implementation. This section presents a user study [NC12d], which measures the performance increase of using the *snapping* and *glue* tools to position virtual objects in photographs. This will show that it is possible for virtual objects to interact with properties of the scenario such as floor or orientation, as stated in **RQ2**, and will show that users are able to introduce objects in three dimensions in the scenario (**RQ3**, section 1.1).

Experimental design

The prototypes presented in subsections 5.1.2 and 5.1.4.2 were tested and evaluated by a group of users [Dix+03]. One of the main assumptions was that most non-experienced users have difficulties placing objects in three dimensions.

The study was conducted to assess the increase in the ability to position objects in the scene in 3D in the correct position and in the shortest time possible. The purpose was to understand the user difficulties in the manipulation of 3D objects and the gain achieved by introducing the proposed tools.

The current prototypes supports two set of tools that help users to place virtual objects in the scene: (T1) magnetic floor and rotation *snapping* (detailed previously in Figure 5.4) and (T2) direct object glue in stereo vision (detailed previously in Figures 5.9 and 5.10).

The study consisted of two tasks and a questionnaire. The first task (T1) was designed to test the floor and rotation *snapping* tools with guidelines. Users had to repeatedly place the box (Figure 5.4) in different points of an image with the correct position and scale factor (Example: “Consider a box, half the size of the

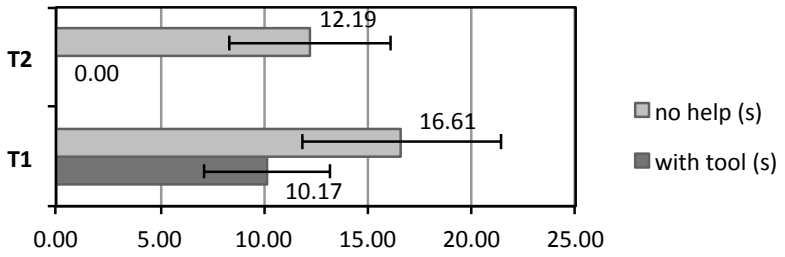


Figure 5.11: Average amount of time spent to position 3D objects using floor and rotation *snapping* (T1), object *glue* (T2) (instantaneous), and the same tasks with no help.

bed"). Each time they had to do it **with** and **without** the guidelines provided by the *snapping* tools, the execution time was recorded.

The second task (T2) was meant to test the *glue* tool, which glues objects (Figure 5.10). The users had to place the poster in a given place and then they had to do the same without help, using translations and rotations. This allowed to test the amount of time required to do the same operation without the help of the tool. The 3D control was done using a mouse. The questionnaire helped in the characterization of each user.

The study counted with 20 users, aged between 20 and 50 years old, and most of them had average technological skills (95%) but no 3D modeling skills (20%).

Results and Discussion

The results show that on the first task (T1) there was a 38% performance increase when using *snapping*, as detailed in Figure 5.11. Using Analysis of Variance (ANOVA) [She04] it is possible to infer that the mean performance time with the tool ($10.17 \pm 3.01s$) is significantly different from the mean performance without any tool ($16.61 \pm 4.79s$), with $F(1, 38) = 5.89, p < .00001$.

When positioning the objects without assistance in 3D, 43% of the times users placed the objects in geometric incorrect positions (wrong scale or rotation). With the help of the snapping system, that number decreased to 11%. Almost all users (95%) agreed that the snapping helped their task, 25% said the improvement was small, and 70% said the improvement was large.

In the second task (T2), users took an average time of $12.19 \pm 3.90s$ to achieve the same result without help of the *glue* (with the tool the placement is instantaneous, $0.00s$). When asked which techniques they would prefer to use in an applications to virtually reshape a room with objects such as furniture (e.g., furniture application in subsection 5.1.4.1), most users referred that they would use

glue (T2) to position the object initially, followed by *snapping* (T1) for some adjustments and then free movement (no tools) in the end. The *glue* tool (T3) raised concerns of constraining the positioning so it must be used only in an initial approach. It should also be possible to turn off the snapping for refined adjustments.

These results show that there is a significant performance increase when using this kind of magnetic properties. They also show that users have a real difficulty in correctly placing objects in a 3D scene. Most users are also open to use an augmented reality system that correctly interprets the scene and agree that the system improves the proposed tasks. Once they start using the system, they build expectations and may expect it to deal with certain situations such as occlusion, which are currently not supported.

5.2 Mixed Reality Snake

This section describes an application which takes advantage of all the detected features (presented in Chapter 4) and presents a user interface to enable a useful interaction. To demonstrate the full capabilities and possibilities of the system, a game was devised to demonstrate the framework (Figure 5.12). In the following subsections, the game concept, interface and user studies are described in detail.

The game [NC13a; NC13b], designed to showcase the proposed mixed reality system is a two-player snake game. The main novelty is that the game can be played in any real-world flat surface that the players can imagine. This can be a table, the living room floor, the back yard or the main street. Additionally, the users can build their own custom snake maze using sheets of paper or any other uniform material (Figure 5.12).

The game look and feel is designed to create the illusion that the snakes are immersed in the photo environment. All virtual elements are displayed as if they were correctly in the scene in three-dimensional space. Also, the gameplay takes into consideration possible obstacles in the playground. These obstacles can be the limits of the space chosen by the users or objects intentionally inserted by the players. The current prototype runs on a PC/Mac, but its interface and algorithms could be easily ported to work on a tablet or smartphone (iOS, Android, or Windows).

This application addresses all the research questions presented in the first chapter (section 1.1), especially the last two **RQ2** and **RQ3**. First (**RQ2**), is it feasible to create augmented reality applications where virtual objects, characters and elements, interact in a user chosen real world scenario according to the

Figure 5.12: Snake AR, a mixed reality game designed to be played in an arbitrary real world scenario.

properties of that scenario? Secondly (**RQ3**), are non trained users capable and motivated enough to perform the necessary steps to initialize the AR environment? Altogether, is there a future for interactive marker-less AR applications? To answer these questions, this game application, which analyses pictures taken by the users, was developed. Figure 5.13 presents a preview of the snake game.

5.2.1 Design Concept

The introduction of augmented and mixed reality in applications and specifically in games has been increasing in recent years with several commercial examples in game consoles (e.g., EyePet[3]) and tablet/smartphone applications (e.g, LayAR, Atelier Pfister). There are several toolkits (e.g., ARtoolkit [ART03; Met13; Vuf13]) and frameworks to develop Augmented Reality (AR) applications. Most Augmented Reality systems obey to certain principles, as they all involve some form of computer vision and either have a fixed (e.g., EyePet) or a free-moving camera [TP12]. In free-moving camera systems the augmented reality elements are usually associated with pre-defined marks (visual or GPS point as in the snake

[3]Eye Pet, Sony Playstation camera game, `http://www.eyepet.com/`

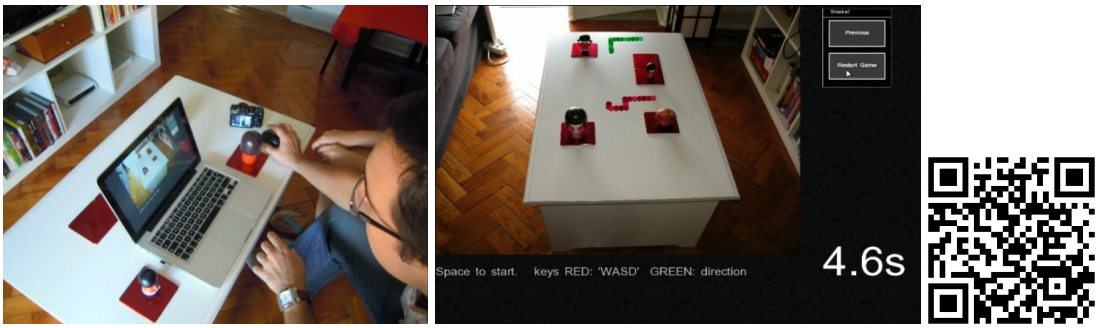


Figure 5.13: Preview of the mixed reality snake application.
(Video: `http://img.di.fct.unl.pt/rpn/phdthesis/snake.mp4`)
(instructions in Appendix B).

GPS based game proposed by Shirose et al. [Shi+12]). Fixed-camera applications commonly use a pre-built structure of the scene.

Currently augmented reality does not necessarily requires using predefined markers or objects. Instead, it is based in image recognition and automatic scene reconstruction. Virtual objects recognize real world elements and react accordingly to scene elements. It is exactly the combination of image analysis and recognition with the augmented reality concept that is explored in this section, enabling the user to choose the scenario where the mixed reality application will take place.

To prepare the game, the player creates a maze in an empty plane (e.g., floor or table). To take advantage of scene orientation (φ_2), the objects composing the maze should be orthogonal as in a Manhattan world for a better detection. The user takes a picture of the scene from the desired position and roughly sketches the floor. This will be used to detect the floor (φ_3) of the maze. The image is then analyzed to find the main vanishing points (φ_1) and scene orientation (φ_2). This process was already described in section 4.3. Currently, there is a previous step of scale adjustment of the snake, but that could be, in the future, inferred by the size of the maze. Finally the snake is positioned on the virtual plane and reacts to the detected floor (the maze in this context) as seen in Figure 5.14.

The initialization of the game requires taking a picture of an open uncluttered space. The user is guided through a simple process where the game level is built. The user roughly selects the area of the game in the picture, the picture is analyzed to build the corresponding 3D model, and finally the game is superimposed to the picture. The main difference of this system is that the scenario for the game is chosen or built by the users in the real world, and the virtual superimposed game obeys to physical constraints such as: scene orientation, perspective and boundaries (e.g., walls, floor, obstacles).

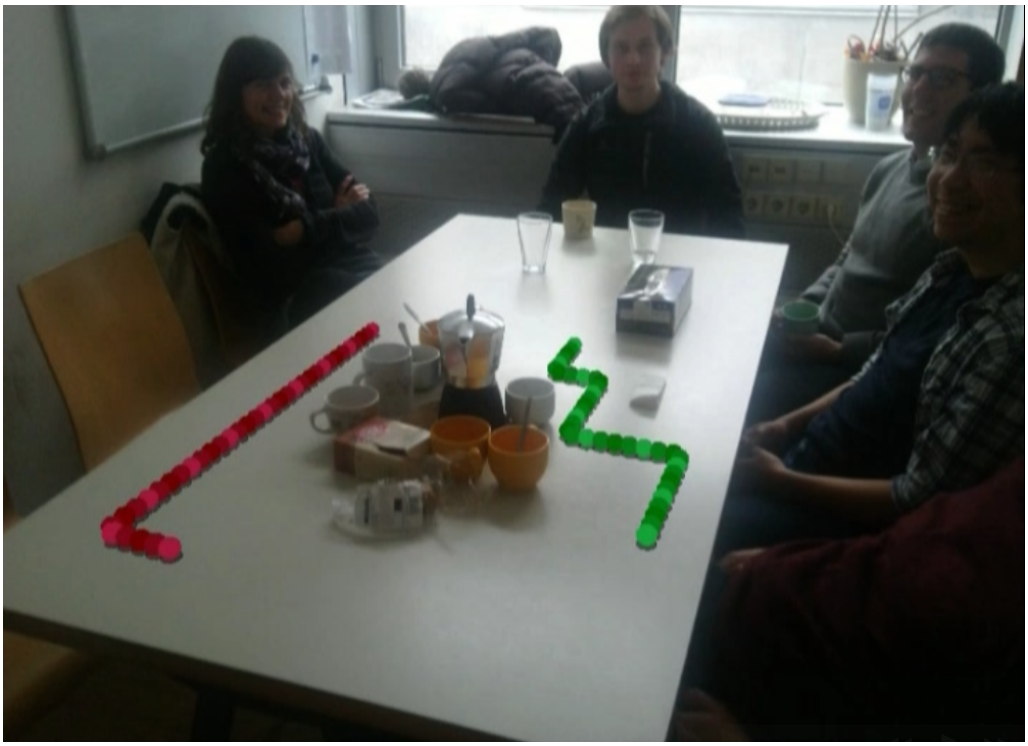


Figure 5.14: Snake AR running in a familiar setting. The snakes must dodge the obstacles in the table. The picture was taken without optimal lightning conditions, with an open area (the table) and several cluttered areas (people and objects).

5.2.2 Interactive Game Interface

The proposed mixed reality system, described in Chapter 4, is designed to extract features from single images and superimpose virtual objects accordingly to these features. The current system supports the automatic detection of the scene floor plane, the main vanishing points, the scene orientation and the floor boundaries.

The proposed game aims to demonstrate the full capabilities and possibilities of the system. The game is targeted to be played at home, in places known by the users (Figure 5.14). The implementation of the game follows the same principles, needs the same libraries and has the same requirements as described in subsection 4.3.7. In the following subsections, the interface of the game is described in detail.

5.2.2.1 The Game

The game takes place in a real-world scenario chosen by the players. Figure 5.15 shows the splash screen, which incorporates this notion, that is, the two snakes

Figure 5.15: Snake game: splash screen.

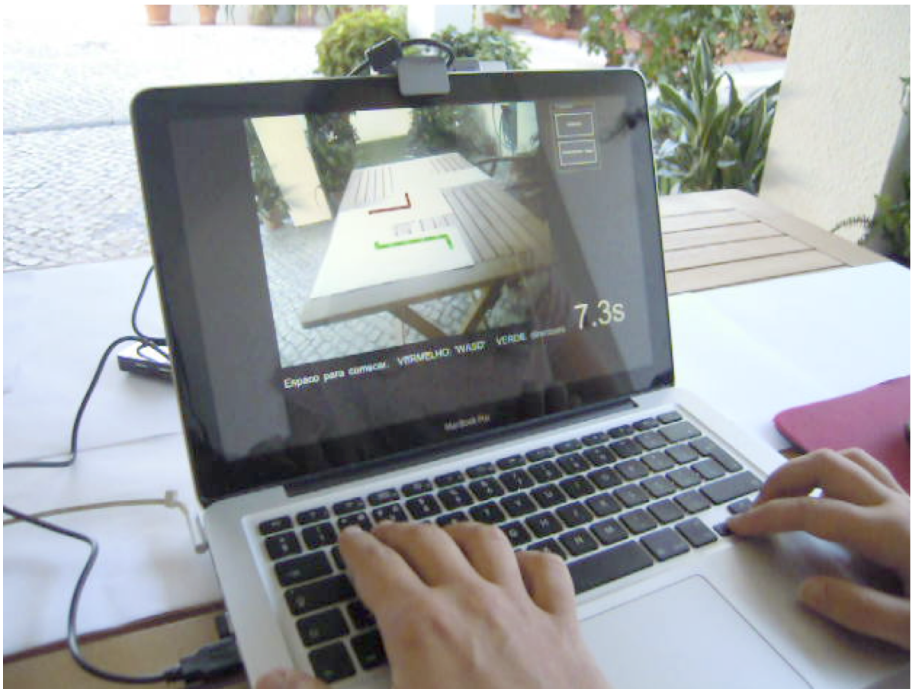
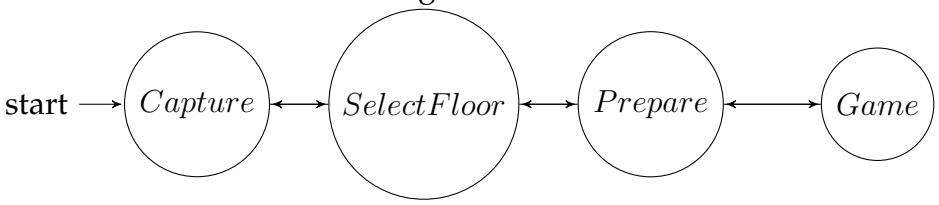


Figure 5.16: Two players playing side by side in the same computer in the Mixed Reality Snake.

are walking around the Louvre main square. Figure 5.16 shows two players playing in a computer.

The players may interact with the game in several different ways. They can build one scenario and play a tournament to see who is the greatest snake master or they can take turns in creating alternative physical game playgrounds and play a limited round of games on each set.

The interaction with the game is done with a standard keyboard using the arrow keys. The initialization process is done using the mouse (or the finger in touch systems). The initialization follows a similar pattern as the Magnetic Objects prototype (5.1.3). The only difference is that this game follows a linear work-flow as seen in the following state machine:



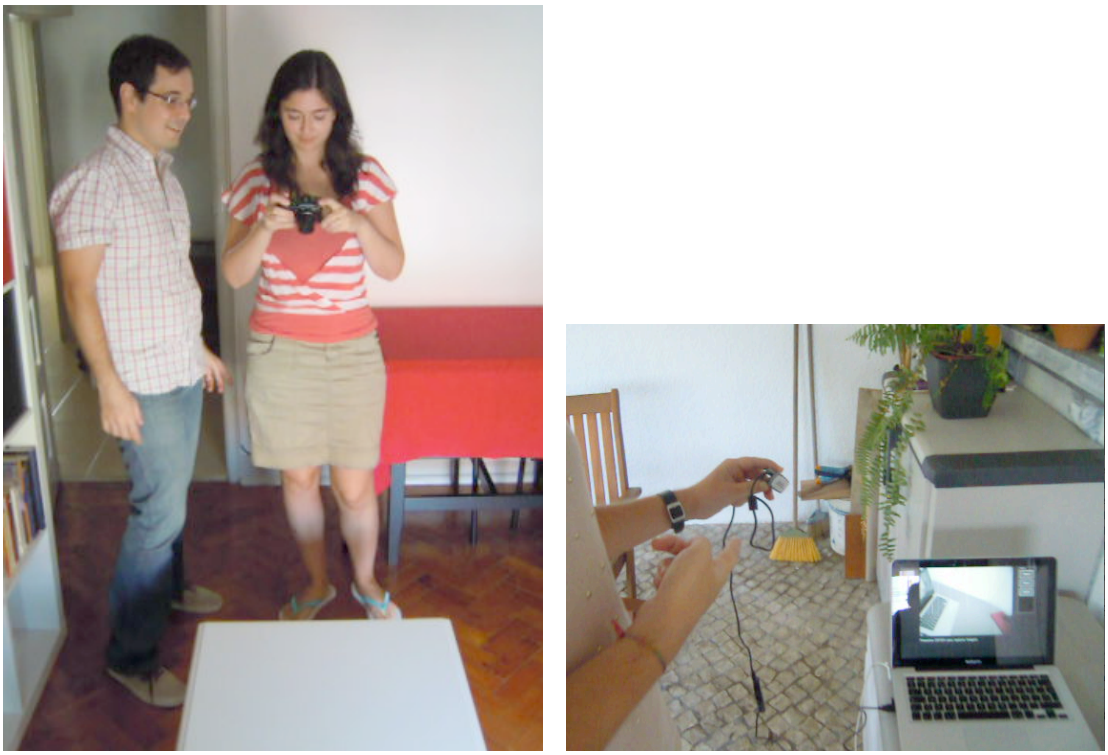
The user captures the image, selects the floor, automatically detects the vanishing points, selects the scale (Prepare node) and then plays the game. This is done in this specific order, with instructions for each step. In the Magnetic Objects prototype, the several steps could be tested in an arbitrary order. The goal is to simplify the process by providing guidance in the initialization until the game is set.

5.2.2.2 Capturing an Image

In this application, as important as the interaction with the game itself, is how users initialize the mixed reality system. The system must be initialized with a picture.

The users are instructed to search for an open uncluttered space with a horizontal plane surface where the game will take place. The scene can be indoor or outdoor as long as the surface that will compose the playground occupies a large portion of the screen. Due to the current implementation, players are advised to look for scenes with several parallel lines. This is usually the case of indoor and modern architecture scenes and is usually described as a Manhattan world [CY99]. Additionally, it is recommended to avoid non-Lambertian [Fol+95] surfaces such as mirrors, windows and reflecting materials.

In the current system users can chose between three methods to insert an image: capture an image with a webcam, use a digital camera with Wi-Fi SD card or load it from disk.



(a) Camera with Wi-Fi SD-card. (b) Webcam mode.

Figure 5.17: Loading the initial image.

Using the webcam (Figure 5.17(a)), the user can take the computer/tablet to the location, take the picture and immediately start to play the game. The main drawback is that the quality of some webcams is not the ideal for the scene detection algorithm. If the augmented reality system were to be implemented in a tablet device with rear camera this would be the main preferable input.

To solve the problem of low quality images from the webcams an alternative input version was used: capturing the image with a 12 Megapixel digital camera equipped with a wireless SD card [4] as seen in Figure (Figure 5.17(b)). This allowed the users to have more freedom in the sense that they could walk around the space without needing to carry a PC or a tablet. This approach generated high quality photos but it is an option that is harder to generalize because wireless SD cards or wireless cameras may not be available for all users.

Loading an image from disk, gives total freedom to take a picture with the user device of choice. Additionally, it is an important feature to load saved scenarios. After photographing the playground scene, the users advance to the next stage.

[4] Eye-Fi Wireless SD card, `http://www.eye.fi/`.

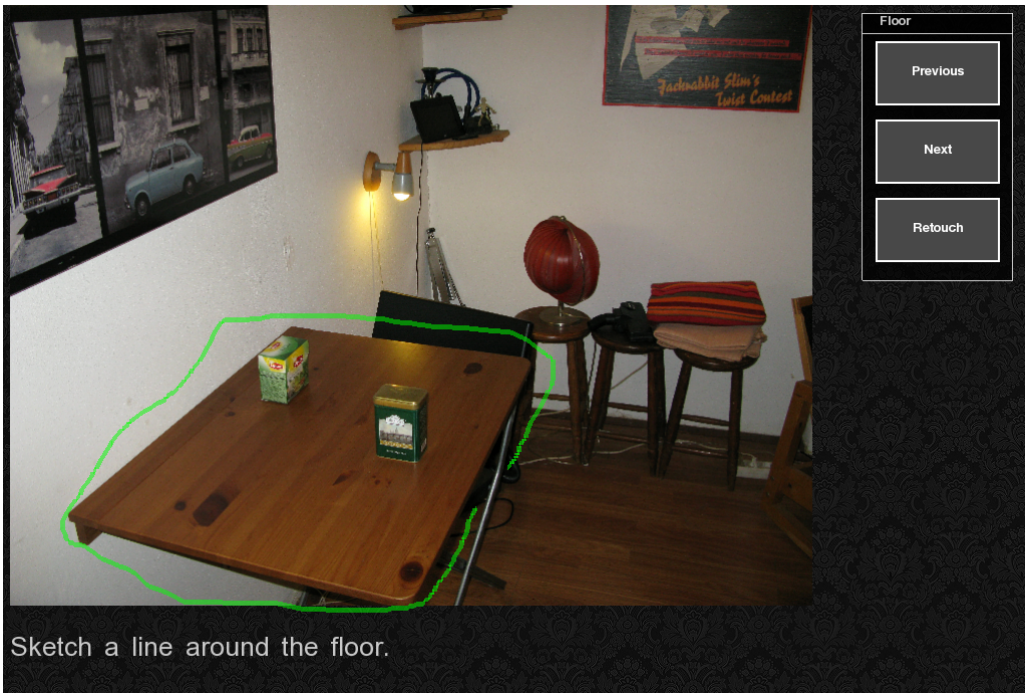


Figure 5.18: Snake game: Selecting the floor. The user draws a line around the playground.

5.2.2.3 Selecting the Playground

After selecting the scenario, there are steps where the user helps the system to initialize the game, much in the same way as in the Magnetic Objects Application described in section 5.1.3. The first step is to indicate to the system the surface where the game will run and what part of that surface should be considered a playground.

To indicate which surface is the playground, one of the players roughly draws a line around the surface (e.g., the ground floor) as seen in Figure 5.18. The system automatically tries to detect all the homogeneous materials on that area, as explained in subsection 4.3.2, and proposes a playground area, presented in transparent red in Figure 5.19. For most users this step is sufficient to detect the expected area and move to the next phase.

Advanced players may want to customize the area by manually adding or removing playground areas. A paint tool exists so that players can redraw another area or retouch the detected area by adding missed parts or removing extras, as explained in subsection 4.3.2. Considering the example of a table with office material on top. Some players might consider using the table as playground and the objects as obstacles; others will consider everything a playground.

This selected area, seen in Figure 5.19, will be where the snakes can move

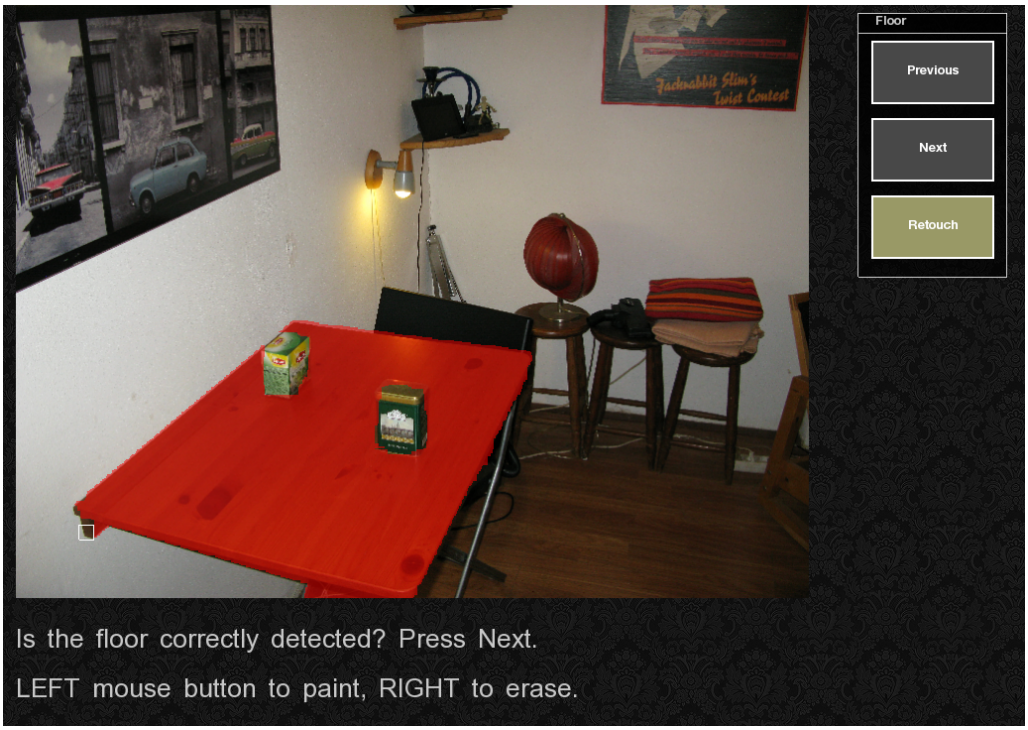


Figure 5.19: Snake game: selected floor. The user can retouch the floor if needed.

through. Leaving this area will be considered a bump in an object and that player will lose. The objective was to create a simple semi-automatic method to chose the playground. Depending on the complexity of the scene, many times the users do not need to retouch the area because the algorithm automatically finds it in a proper way. Also important is the fact that the processing time is never more than three seconds to detect the area (results in subsection 4.4.2), meaning that the users can retry several times until they are satisfied. When the playground is defined players can advance to the next stage.

5.2.2.4 Adjusting Perspective and Scale

In this final screen before the game starts, the mixed reality model M is internally processed and initialized. Again this will only take one or two seconds to load. The photographed scene is displayed with a small 3D wooden box in the center of the scene, as presented in Figure 5.20. If the scene perspective was correctly detected, the box will appear to be on top of the selected surface. Additionally the box will be aligned with the photographed objects in the scene (i.e., the furniture, chairs, and floor lines). In an ideal situation, the game is ready to start without further intervention from the users by pressing "Start Game".

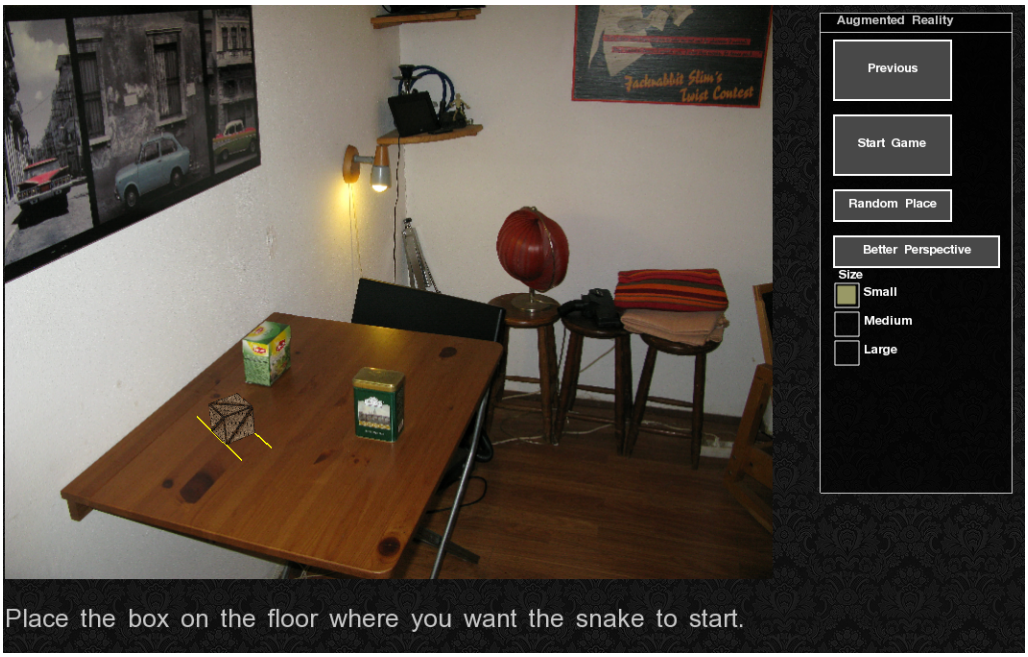


Figure 5.20: Snake game screenshot. The user can choose the starting point of the game, a better perspective and the scale.

For the situations where the perspective was not well detected there is a button that cycles through the second and third best perspective hypothesis. This is done by cycling through the several detected vanishing points as described in subsection 5.1.3. Most of the times the second or third guess from the system is correct.

The goal of the virtual wooden box is to visually determine if the perspective is correct and also to decide where the game will start. The players can also choose between three different scale levels for the game. If the playground is narrow the small scale is more adequate if not a larger scale can be used. The three discreet optimal sizes were picked after several lab tests (subsection 5.1.5) and are easier to choose from than the continuous slider used in the Magnetic Objects prototype. With a continuous slider many users ignore the function and others have doubts about which size is most suitable for the game.

With the start position selected and the correct size, the game is ready to start. All options can always be reconfigured later; the players can even go back and select a different playground.

5.2.2.5 Playing

After configuration, the players are side by side rivalling against each other and the game starts as presented in Figure 5.21. In the current prototype the snakes

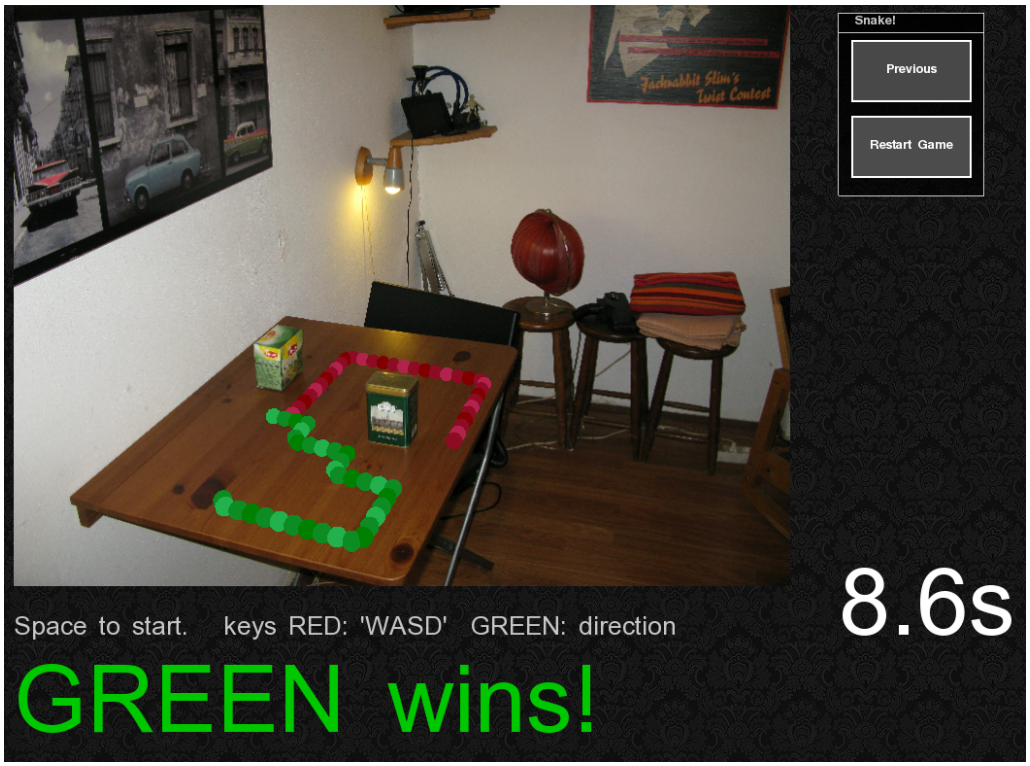
Figure 5.21: Snake game: running. The users control both snakes with the keyboard.

are controlled with the keyboard. To enhance the game immersion there is a retro MIDI sound playing in the background and a timer counting the time that each snake stays alive as seen in Figure 5.22(a). The game logic is simple, each snake will grow over time and the game will get gradually faster in order to speed-up the demise of one of the players.

In the example of Figure 5.22, the snakes should not bump into each other nor the area outside the selected area seen in Figure 5.19. Along with the perspective, this will create the illusion that the snakes are in the room and that the game will end if they bump into the furniture.

Although conceptually simple, the game has proven to be quite addictive and to stimulate competition between players. One of the key factors that differentiate this game from others is that the players can choose different scenarios from real places. This can generate a competitive behavior were the players choose and create the playground, in turns, with each player creating a more difficult maze each time.

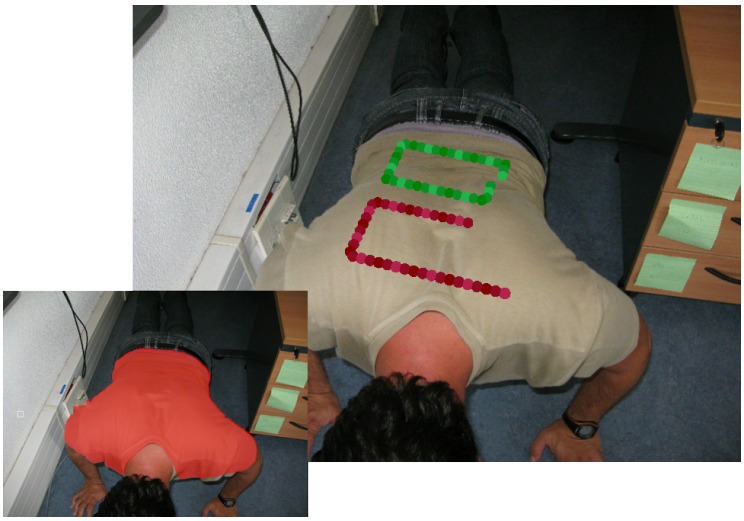
There are several possibilities that can be used to create a playground. Figure 5.23 shows some examples that were created during the user studies (next subsection). Some users prefer to play in wide outdoor areas, pictures taken from



(a) Red hits green snake. Green wins.

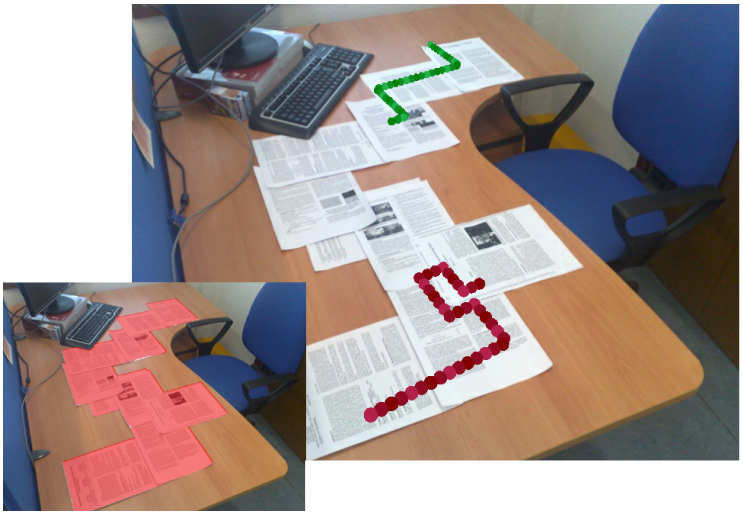
(b) Green hits wall. Red wins

Figure 5.22: Snake game: Gameplay.



(a)

(b)



(c)

Figure 5.23: Snake game. Building a custom level.

windows or images of known places. Others prefer to choose an open spot in a room of the house. There is a large portion of players that enjoy creating a custom playground using obstacles on a table.

Although the processing time of the initialization is rather low, several users spend more time in the construction of the playground (and enjoy it more) than actually playing. Additional considerations about the users' behavior are drawn in the user-study section.

5.2.3 User Studies

To answer the request questions (**RQ2** and **RQ3**) presented in the beginning of this section (section 5.2), two main tests were devised. The first test is a user study with a group in a household environment (subsection 5.2.3.1). The goal was to evaluate the reliability of the application in a real-life situation and improve the interface of the game. The study was conducted in the users' home environment.

With improvements and lessons learned from the first study, a large scale second user study (subsection 5.2.3.2) was conducted to evaluate if the users were capable of initializing the system on their own and were interested and motivated by the game itself. The following subsections detail each user study.

5.2.3.1 Household Environment

The mixed reality snake application was evaluated in a systematic study [NC13a] conducted with 11 users to evaluate if they could perform correctly and in a short period of time the initialization of the AR snake game. This was an initial study, taking place in a household environment. Users were interviewed in their home, mostly in pairs (couples or roommates). The idea was to simulate an indoor environment where the user plays the mixed reality snake game with his/her friends in one of the rooms of the house. The goal of this test was to prove that users are capable of initializing the system and are motivated by the mixed reality application (**RQ3**, from section 1.1).

Experimental design

The experience apparatus took place in the users' households. This was important because part of the goal was to understand how users would use the system in their private spaces, living rooms and bedrooms. The participants had between 20 and 40 years old, and had in general good proficiency with technology although almost none (90.1%), had prior experience with Augmented Reality. Whenever possible the studies were filmed for later analysis.

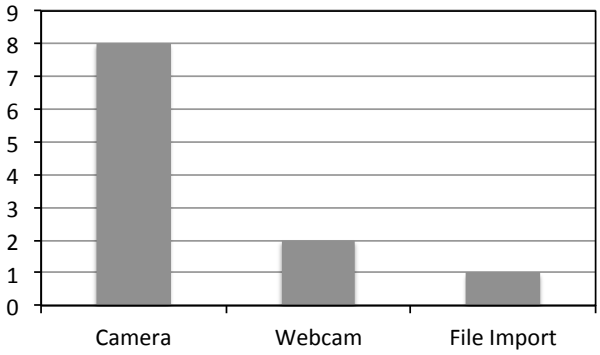


Figure 5.24: The photo camera with Wi-Fi SD Card was the preferred input method for the game.

The experience with the AR Snake was divided into four main parts: (1) creating a physical game level and capturing the image, (2) selecting the playground, (3) adjusting perspective and scale and (4) playing the game. For each step they were asked about how simple the execution was. Additionally, several questions were asked to the users to evaluate the interest and acceptability of such system. The questions were answered in a 5 points Likert-scale.

The users were encouraged to use a flat surface, such as tables or the floor, and to use everyday objects as obstacles, as exemplified by Figures 5.14 or 5.23. To capture the scenario, they could use a photographic camera with wireless SD card, a webcam or import an image from the disk. The users had to experiment all the alternatives, as demonstrated in Figure 5.17. After that, the users had to follow the game steps as previously described in the Interactive Game Interface subsections (subsections 5.2.2.3, 5.2.2.4 and 5.2.2.5). The users were given some time (between 5 to 20 minutes) to familiarize themselves with the system before answering the questionnaire.

Results and Discussion

The questionnaire's results can be observed in Table 5.1 and Figure 5.24 presents the preferred input method according to the users. Most users enjoyed more using the camera (Figure 5.17(a)) because it allowed more freedom than the webcam (Figure 5.17(b)). File import was interesting to use in holidays' pictures and images obtained in the Internet.

All players concluded the entire process successfully and within one or two minutes (the time increased with experience). Some complained about the mouse (too fast or too slow), others (27.2%) took a while to understand the floor selecting system, namely that the selection line should be roughly around the floor and not near the floor borders.

Table 5.1: Questionnaire on household Snake user study. Statements with Likert-scale answers being 1 - Disagree and 5 - Agree. Median with Lower and Upper Quartile Deviation and Mean with Standard Deviation.

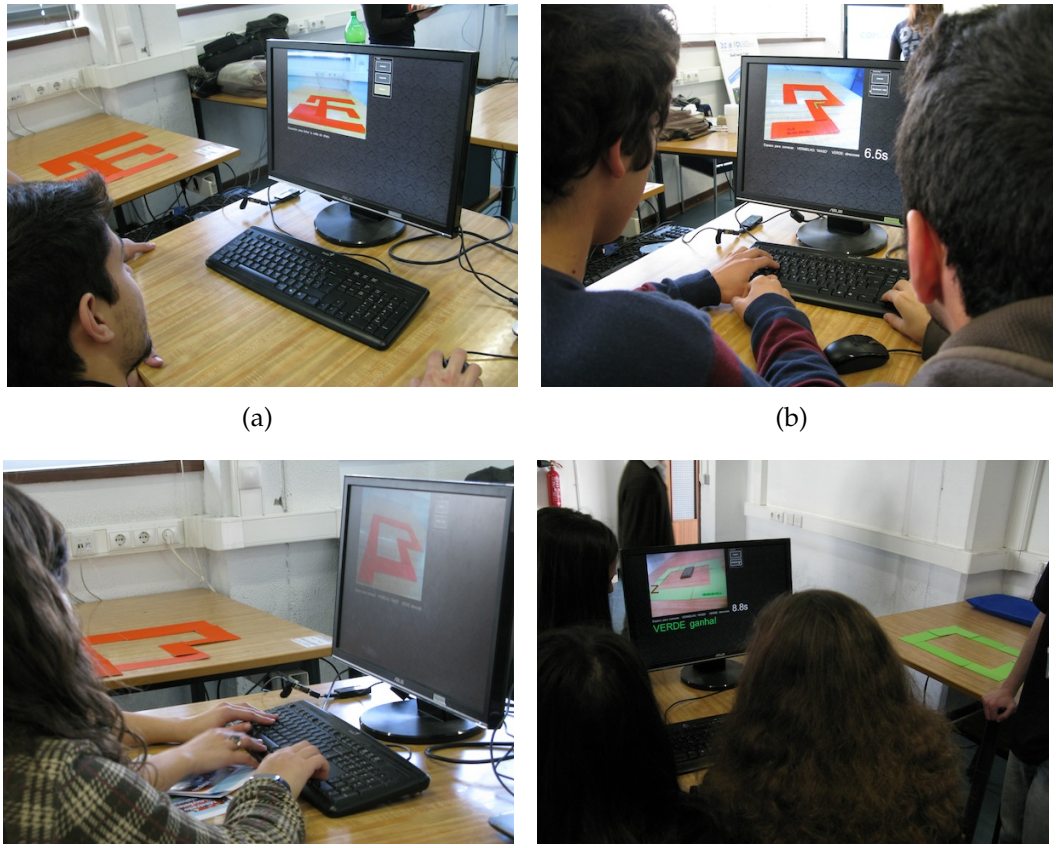
	Statement brief description	**Median**	**LQ**	**UQ**	**Mean**	**SD**
1	**User's opinions:**					
1.1	**(1st step)** Capturing the image was a simple task.	5	-0.5	0	4.73	0.47
1.2	**(2nd step)** The selection of the game floor was a simple task.	4	0	+1	4.18	0.75
1.3	**(3rd step)** Adjust the starting point and the perspective was a simple task.	4	-1	+1	3.91	1.30
1.4	**(final game)** The game looks integrated with the photo.	5	0	0	4.91	0.30
1.5	All initialization steps were simple to execute.	4	0	+1	4.27	0.65
1.6	The final game was interesting and exciting.	4	0	+1	4.45	0.52
1.7	Playing in a photographed space chosen by the user increases the interest of the game.	5	0	0	5.00	0.00
2	**How would you use a mixed reality application such as this?**					
2.1	Tablet with back camera.	5	-1	0	4.36	0.81
2.2	Smartphone with back camera.	4	-0.5	+1	4.09	1.04
2.3	Computer application with webcam.	4	0	+1	4.27	0.79
2.4	Computer application with photographic camera.	4	0	+1	4.09	0.94
2.5	Facebook application.	4	-2.5	0	3.09	1.58
2.6	Web application.	4	-1	+0.5	3.73	1.01
3	**Application Preferences and would you use this system?**					
3.1	I think it is interesting to use virtual objects in real scenarios.	5	0	0	4.82	0.40
3.2	Imagining a furniture application (...) using this mixed reality system.	5	-0.5	0	4.55	0.82

The first time users selected the floor they were really careful and slow (2nd step on Table 5.1), but after a while, their confidence in the system grew and they started drawing faster and less elaborated sketches. Some initial critiques were done to the fact that the snakes do not pass behind objects, but most of this changed as the game started and players understood the limitations of the system. When the initial perspective was not entirely correct users took a while to try the improve perspective button (3rd step of Table 5.1). The answers to the questionnaire were very positive although some difficulties were expressed regarding steps 2 and 3 (Table 5.1)

Most users were highly motivated and satisfied with the game (e.g., 4.45 mean in question 1.6), and state that the fact that the game runs in a space that they prepared and photographed improves the game immersion and involvement (5 mean in question 1.7). When asked to score devices and environments they would use, in a AR game/system such as this one, the users chose in average by this order (question 2 of Table 5.1): tablet, computer application with webcam, smartphone with back camera, computer application with wireless digital camera, web application and application on a social network such as Facebook. In question 3, most users were very positive about the prospect of using this system to introduce virtual objects in real scenarios (4.82 mean in question 3.1) and to use it in a furniture testing application (4.55 mean in question 3.2).

One of the most positive aspects of the snake game is the competitive factor of being a two-player game. This meant that most users stayed to play long after the proposed time for the user study. Some were not so interested in playing the game, stating that they were not good at games, but were interested in constructing the model. The creation of the game level using everyday objects was one of the most important aspects of the game. Most users show much interest on this aspect (as stated by the unanimous 5 score in question 1.7). Many want to find improbable places to play such as the belly of a cat or old black and white photos. Although the detection system is not prepared to deal with all types of places (as shown by the results in subsection 4.4), there is a learning factor where after a while users understand that the system performs better in uniform and uncluttered surfaces with defined line limits.

The study results were encouraging and show that the players can create multiple game levels from different scenes in different environments. The game itself was received with enthusiasm by all users.



(a) (b)

(c) (d)

Figure 5.25: Large scale user study with 81 users. The users created a maze using card boards and then played the snake game on the maze.

5.2.3.2 Large Scale User Experiment

The previous study (subsection 5.2.3.1) was important to reveal the feasibility of the mixed reality snake game and to identify problems in the implementation and in the study. After refining the prototype a similar study was conducted in a larger scale with 81 users. The goals of this study are the same as the ones from the previous household study but with a higher number of users.

Experimental design

The study follows the same logic and apparatus as the household study in the previous subsection (subsection 5.2.3.1). The main differences are related with logistics. The study took place in a classroom during an open science day in the University. For this reason, most of the users were high-school students (80%, with ages between 15 and 19) and were visiting the campus. The students played in pairs with their friends and colleagues and interacted with the system between 10 and 20 minutes, as seen in Figure 5.25. The interaction happened in a desktop

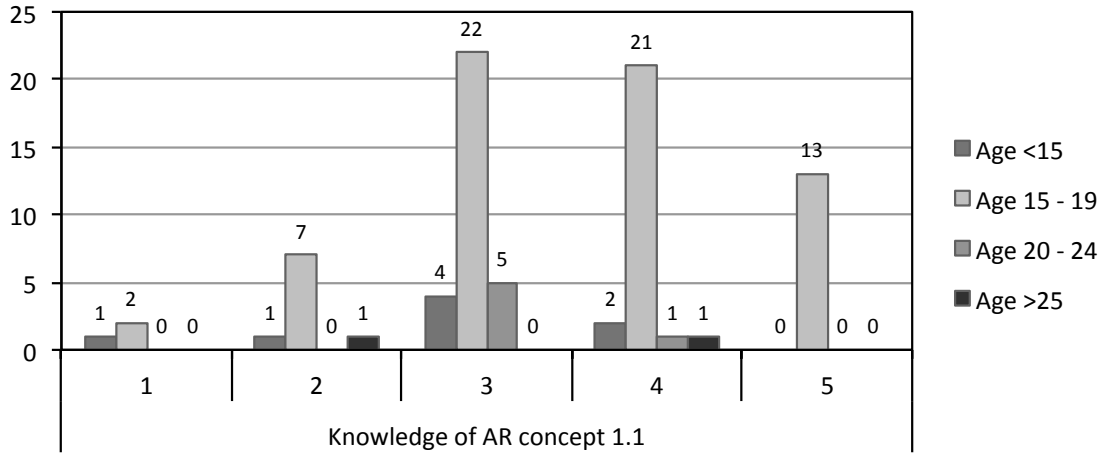


Figure 5.26: Question 1.1, knowledge of the AR concept. Values by age range for each score.

computer with a webcam (no photographic digital camera this time as in the last study). Next to the computer there was a table with several pieces of paper (they looked like Tetris pieces). The players were instructed to create a maze with the papers and to initialize the game with that maze. In large groups several informal contests were organized to find the best player. Each user had to initialize and interact with the game, and in the end, answer a questionnaire.

Results and Discussion

The questionnaire's results can be observed in Table 5.2. This questionnaire is very similar with the questionnaire presented in the previous subsection in Table 5.1. The results are still quite positive and very similar but there are some aspects, which present some differences, which will be discussed next.

The goal of the first question (1.1) in Table Table 5.2 was to evaluate how familiar were the users with the concept of Augmented Reality (Mixed Reality is much less known so it was not asked). The results detailed in Figures 5.26 and 5.27, show that there is a large dispersion of results with no significant difference between gender or age (mode is 3).

The initialization questions (questions 2.1, 2.2 and 2.3) have similar positive results as the household study. As expected the capture (2.1) decreased because this time only the webcam was available. The selection was one of the interface aspects, which were improved (better mouse interaction), and the results from question (2.2) prove that users are more satisfied this time.

The last four questions from group 2 show that there is a large satisfaction with the game and the mixed objects concept. In Figure 5.28, it can be observed

Table 5.2: Questionnaire on a large scale user study about SnakeAR. There are some differences from Table 5.1. Statements with Likert-scale answers being 1 - Disagree and 5 - Agree. Median with Lower and Upper Quartile Deviation and Mean with Standard Deviation.

	Statement brief description	**Median**	**LQ**	**UQ**	**Mean**	**SD**
1	**User's background:**					
1.1	I am quite familiar with the concept of Augmented Reality.	3	0	+1	3.44	1.01
2	**User's opinions:**					
2.1	**(1st step)** Capturing the image was a simple task.	5	0	0	4.62	0.66
2.2	**(2nd step)** The selection of the game floor was a simple task.	4.5	-0.5	+0.5	4.31	0.83
2.3	**(3rd step)** Adjust the starting point and the perspective was a simple task.	4	0	+1	4.24	0.76
2.4	**(final game)** The game looks integrated with the photo.	5	-1	0	4.47	0.69
2.5	All initialization steps were simple to execute.	5	-1	0	4.36	0.75
2.6	The final game was interesting and exciting.	4	0	+1	4.30	0.68
2.7	Playing in a photographed space chosen by the user increases the interest of the game.	5	-1	0	4.45	0.72
3	**How would you use a mixed reality application such as this?**					
3.1	Tablet with back camera.	4	-1	+1	3.91	1.23
3.2	Smartphone with back camera.	4	-1	+1	3.94	1.18
3.3	Computer application with webcam.	4	-1	+1	3.92	1.17
3.4	Facebook application.	3	-1	+1	3.18	1.37
3.5	Web application.	3	0	+1	3.31	1.24
4	**Application Preferences, would you use these systems?**					
4.1	I think it is interesting to use virtual objects in real scenarios.	5	-1	0	4.51	0.65
4.2	Imagining a furniture application (...) using this mixed reality system.	4	0	+1	3.95	0.89
4.3	Imagining a game in the living room (...) which interacts with furniture.	4	0	+1	4.25	0.80
4.4	Imagining a redecorating system which allows you to change your room and share it with friends.	4	0	+1	4.20	0.84

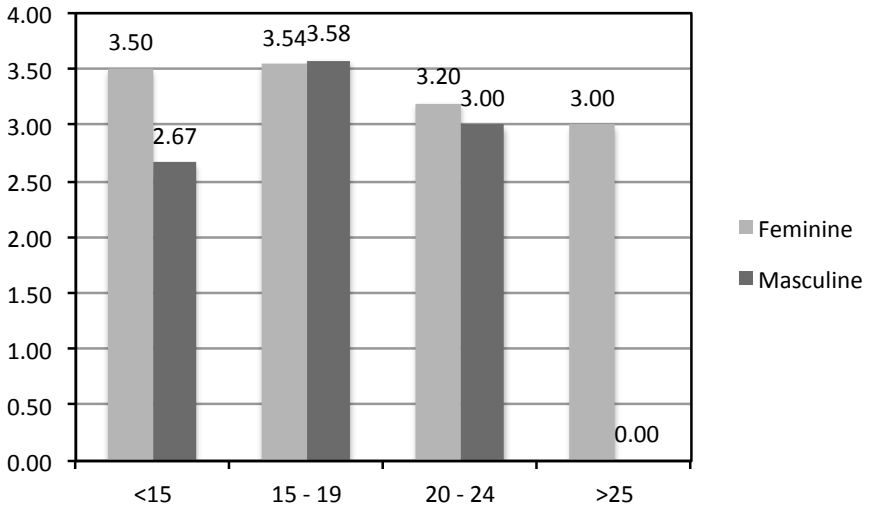


Figure 5.27: Question 1.1, knowledge of the AR concept. Mean score by age range.

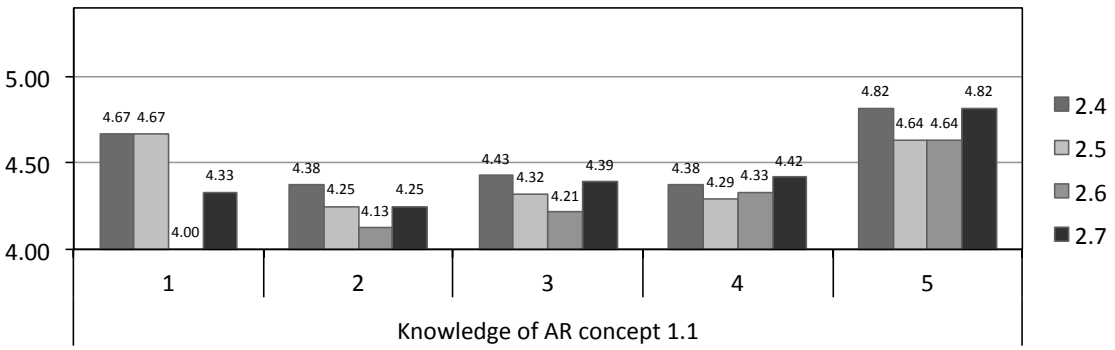


Figure 5.28: Mean score in questions 2.4, 2.5, 2.6 and 2.7 according to the knowledge of the AR concept (1.1).

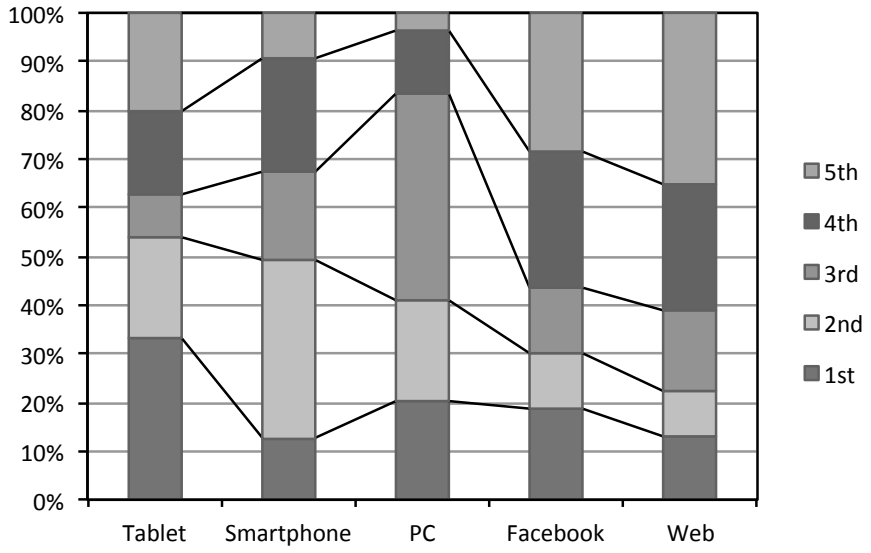


Figure 5.29: Platform preferences. For each platform the bar presents the percentage of each choice starting by the first in the bottom.

that this satisfaction is higher for users that have more knowledge of augmented reality and for users with no knowledge at all.

In the third group of Table 5.2, users were asked which platform they would be interested in using this type of games. The answers ordered by mean are: smartphone, tablet, computer, website and Facebook. In addition to the Likert-scale questions the users had to order the platforms by preference from 1 to 5 to evaluate if the above order was significant. The results, shown in Figures 5.29 and Figure 5.30, present an absolute and accumulated view of preferences. Looking only at first and second choices there is a clear indication that users would like to play this game in tablets and smartphones.

In the final group of Table 5.2, the users show a high interest in using virtual objects in real scenarios (question 4.1) and are very interested in the proposed scenarios in questions 4.2, 4.3 and 4.4. Figure 5.31 presents the average results for questions 4.2, 4.3 and 4.4. There is a small dominance of the scenario proposed in question 4.3, a game which interacts with the furniture of the living room, next comes the redecorating system (4.4) and the furniture application (4.2). These preferences probably reflect the younger age of the users, as older users could be more interested in the furniture application.

These results confirm the interest and acceptance that the game had in the household user study (previous subsection). Most of the students were very interested in the game and in its concept. The game also performs quite well with large groups, with users spontaneously organizing leagues and playing in turns

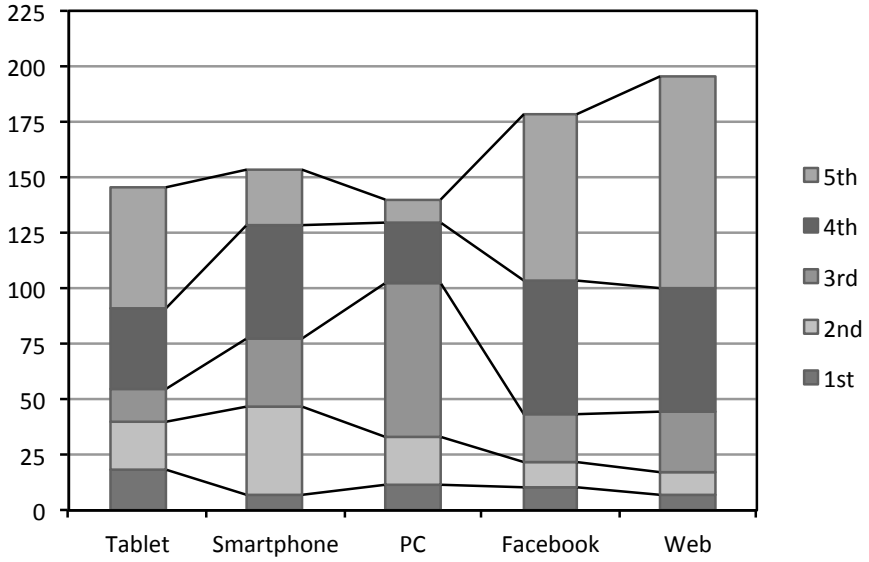


Figure 5.30: Accumulated choice: each bar is obtained by multiplying the number of users that chose a technology (n_i) by the choice position. The final value for each technology is $(n_{1st} * 1) + ... + (n_{5th} * 5)$. The lowest aggregated bar is the most preferable.

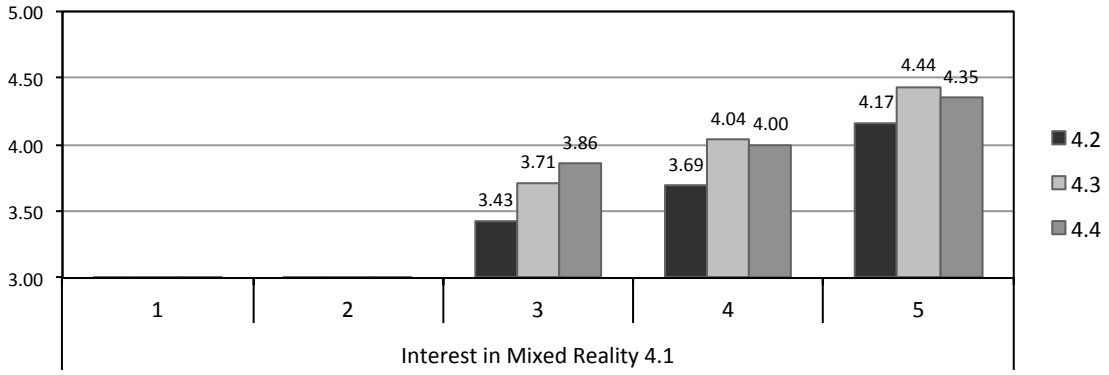


Figure 5.31: Average results for questions 4.2, 4.3 and 4.4 according to the interest in objects in real scenarios (question 4.1).

and rebuilding the paper maze several times. This user study, together with the household study (subsection 5.2.3.1) shows that the users are capable and motivated to initialize several times the virtual model required to play the snake game (**RQ3**, section 1.1).

5.3 Past Museum Exhibition Navigation Through Overlapping Images

The previous sections have shown the feasibility of mixed reality applications based mainly on a single photo exploration. The previous applications (sections 5.1 and 5.2) were entirely based on the framework presented in Chapter 4. In this section, a new approach is presented to explore a space using several photos from that same space. Building upon the concepts detailed in Chapter 3 a prototype was implemented to build relations between images (sections 3.1 and 3.2) and construct a web exploration system of a physical place.

The prototype focuses on building an interactive exploration of a modern art exhibition, which occurred in the year of 1957. This was the first modern art exhibition organized by Gulbenkian Foundation in Portugal[5], from which there are only several black and white images, a catalogue (in paper) and a film of the opening.

The exploration, detailed in the following subsection, is an alternative example of what kind of applications can be implemented using images of a certain place. While the previous examples focused more on real-time captured scenes, in this section the most interesting aspect is the exploration of archive photographs.

The main questions that were addressed in this project were, first, the creation of a digital documental framework able to store all the information, and secondly, the creation of a multimedia visual interface that can recreate an exhibition from the past. This also relates with the first research question (**RQ1**), presented in section 1.1, where the goal is to evaluate the feasibility of creating an interactive application from non predefined scenarios. In this section the goal is to build an interactive exploration from photos for any scenario, even a scenario which does not exist any more.

Creating a visualization system for an exhibition that occurred many years ago, poses additional challenges that need to be addressed. The first obvious difficulty is that it is impossible to visit the exhibition; sometimes even the building

[5]Biblioteca de Arte Fundação Calouste Gulbenkian, `http://www.biblarte.gulbenkian.pt/`.

where it occurred is structurally different or no longer exists. The reconstruction is entirely dependent of documental photos, from that time, old building blueprints, and archive listings. The old images from the exhibition are analyzed using computer vision techniques to detect overlaps and continuities between them. These continuities can later be explored in the semi-automatic reconstruction of the exhibition spaces.

The visualization module is a web-based multimedia application with an interactive interface to virtually explore the exhibition and was mostly developed in the context of a master thesis [Nob12]. This dissertation is more focused in the developed techniques required for the visual homography relation between images (section 3.2) and in the panoramic exploration, which will be presented later (subsection 5.3.3).

The current prototype [Nó+12] was developed by a multi-disciplinary team of art historians, designers and engineers. The art historians include Leonor Oliveira and Raquel Henriques da Silva from IHA/FCSH/UNL[6]. The website interface was designed by Ana Bárbara Teixeira and the implementation and documentation was executed by Carlos Nobre, Rui Nóbrega and Nuno Correia from CITI/FCT/UNL[7]. Figure 5.32 presents a preview of the homography calculus prototype and the web-based application.

Figure 5.32: Preview of the museum exploration.
(Video: `http://img.di.fct.unl.pt/rpn/phdthesis/gulbenkian.mp4`)
(instructions in Appendix B).

The application can be accessed in the following internet addresses:

Development version: `http://img.di.fct.unl.pt/expo1957`
Production version: `http://expo1957.fcsh.unl.pt`

[6]Instituto de História de Arte, Faculdade de Ciências Humanas e Sociais, Universidade Nova de Lisboa, `http://iha.fcsh.unl.pt`.
[7]Center for Informatics and Information Technologies, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, `http://citi.di.fct.unl.pt`.

5.3.1 Exhibition Reconstruction: Motivation

The preservation of the historical heritage has been one of the major roles of museums internationally [Eli+07; Min+01; Sty07]. Art museums in particular have been responsible for the acquisition, inventory, study and exhibition of a large number of cultural and artistic artifacts through the last two centuries. Currently, the memory and heritage of these old exhibitions lives mostly unorganized in museum archives, or in the minds of some of the organizers and participants.

Several museums around the world have a web-based system [Min+01] where users can visit or preview the archives and the exhibitions' material. This creates a large heterogeneous distributed archive of the world's cultural heritage. Web systems are usually focused on the current exhibition that is being presented in the museum. This means that past temporary exhibitions get lost forever from the physical world and from the virtual web space. But what if these temporary exhibitions could still be visited online for a longer period? And what if those past exhibitions, from which only a few of records remain, could be revisited?

Studying the history of such museums allows the creation of bridges to the past in areas such as art collecting, heritage policies, artistic creation and exhibitions' models and patterns. The study of the same museological spaces in different time frames creates a better understanding of past points of view regarding the design of exhibitions.

This project proposes a framework to organize a digital catalogue from an art exhibition. The old images from the exhibition are analyzed using computer vision techniques to detect overlaps and continuities between them. These continuities can later be explored in the semi-automatic reconstruction of the exhibition spaces. The visualization module is a web-based multimedia application with an interactive interface to virtually explore the exhibition. This module also supports an administrator role where specialized users (artists, historians, archivists) can add and edit the exhibition information.

5.3.2 Design Principles

The design principle was to create a tool to present the information about an historical event or exhibition, and create a form of visualization and navigation that would enable access for a wider audience. This relates with the third research question (**RQ3**, section 1.1) about if users are capable and motivated to use and initialize this system. The expert users can be historians, archivists or artists with

access to archive data. The data is composed of images, videos, artwork descriptions or texts. Images from the exhibition floor and from the artworks will contribute to a rich navigational environment.

The expert users supply the system with all archive data and some relational information. With the archive data and especially with the images, the system should be able to infer some relations on its own. On each exhibition floor image the artworks can be automatically identified, by comparing the floor images with the artwork images. By comparing the floor images among each other, it should be possible to infer which pictures belong to the same room or which pictures represent a contiguous area. Since most of these images come from historical archives, and cannot be repeated, they can have a low quality or may not exist in enough quantity. As such, the automatic detection system is not sufficient to create a full reconstruction of the exhibition. Some of the relations between images may have to be introduced manually after the automatic process runs.

The main input to this project was the offline analysis of the images to search for homography relations, as explained in section 3.2. The offline analysis of the images was done using a server based on OpenCV [Ope13]. Using the relations between images several panoramic reconstructions were created and incorporated in a virtual navigation system.

The navigation in the reconstructed scene is based on the overlaps between each photo. Using photo-stitching techniques [SSS06] (SIFT + RANSAC + Homography) (section 3.2) a prototype was constructed, as presented in Figure 3.10, to find the relation between images and the corresponding distortion necessary to present a photo-stitched panoramic view of the scene. This is important to show the surroundings of the scene and increase the immersion of the user. The current system assumes that a large quantity of images per room exists, and there is some degree of overlapping in the photos. The entire set of archive images is pre-analyzed offline to detect continuities on different photos.

Using the collected documents and the extracted information, the goal is to use the database to visualize different aspects of the museum. It will also enable public authorities, museum directors and professionals to reflect on museum activities by providing easy access to a historical background.

5.3.2.1 Case-study and Implementation

With the partnership of a major art museum (Fundação Calouste Gulbenkian), a web-based prototype was implemented to reconstruct the 1957 art exhibition. This project required a significant effort to search for missing documents and

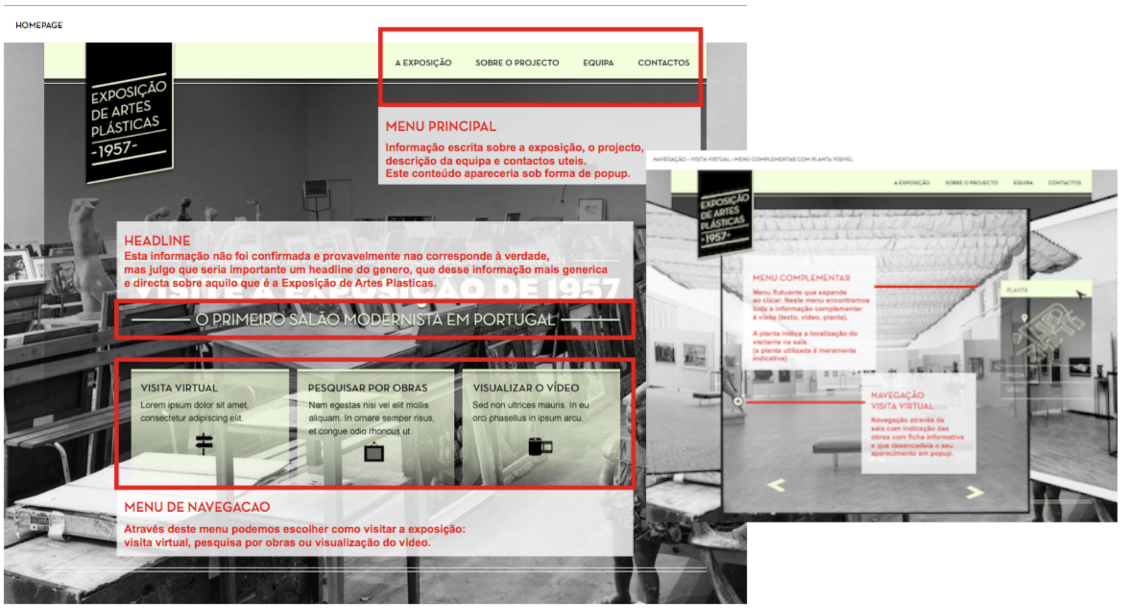


Figure 5.33: Design concept sketch of the proposed interface (Designer: Ana Bárbara Teixeira).

references. As a result, an archive of black and white photos was collected representing the exhibition rooms. The catalogue with the information about all the artworks was also recovered and digitized. The map from the exhibition had to be inferred from the photos because the building interiors are slightly different now. The web application design was custom made for this exhibition but takes advantage of the developed framework and database.

For the creation of the graphic interface the main concept was to maintain the aesthetic of that particular period when the exhibition took place, so that the user experience would be as immersive as possible. Intense research of the historical documentation allowed the use of graphic elements (such as, fonts, icons and visual elements applied in the navigation) from the exhibition's period.

The reconstruction of this virtual environment was improved by the photographic/videographic archive that was gathered and, as such, the graphical user interface and the platform navigation system give special significance to the photographic dimension. Figure 5.33 presents the preliminary design sketch of the interface where every feature is described.

The architecture was implemented with several technologies, most of them already described in Chapter 3 in section 3.7.

The graphical and visual part of the application use HTML 5 and Processing.js[8]. Internally the interface and the management system of the application, which communicates with the database and interface, were constructed using Java Server Pages and Rich Faces[9]. The data and its relations were stored in a XML database. The database is organized around two main entities, artwork and exhibition floor images. Each artwork has several fields with information, and the floor images where they appear. Each floor image has the artworks that it contains, its location and spatial relation with other floor images.

5.3.3 Visual Interface

The main contribution in the visual interface was the introduction of the panoramic navigation mode, which takes advantage of the technologies described in the previous subsection. A more traditional navigation approach was also implemented to have a comparable alternative. The two alternatives can be seen in Figures 5.34 and 5.35. The remaining sections of the interface are presented in the next subsection.

Entering the navigation section of the website there are two modes of navigating: strip or panoramic. In strip navigation there are pictures that represent the rooms where the artworks are in. The main picture, represented on the center of Figure 5.34, is the one being visited at that moment. Inside, several artworks can be consulted by selecting the small plus icons. There are two other adjacent images, on the right and left that represent the next and previous images of the room. This means that the rooms are represented by a circular list that can be accessed through these pictures by clicking on them. There is also a map that indicates the user's current position in the exhibition and enables the selection of other rooms to visit. After selecting one of the rooms, the application will redirect the current place to the place selected.

The panoramic navigation is the proposed navigation mode in this project. In the panoramic navigation presented in Figure 5.35, it is possible to navigate through the images using the spatial relation between them. This spatial relation was extracted using the analysis techniques described in subsection 5.3.2. Although very similar with the strip navigation mode, this navigation mode uses images that appear overlapped to the main picture. These are usually pictures from the same area, but taken from different points of view. Cycling through the

[8]Processing JS, Javascript version of the graphic framework Processing, `http://processingjs.org/`.

[9]Rich Faces, Java Ajax library, `http://www.jboss.org/richfaces/`.

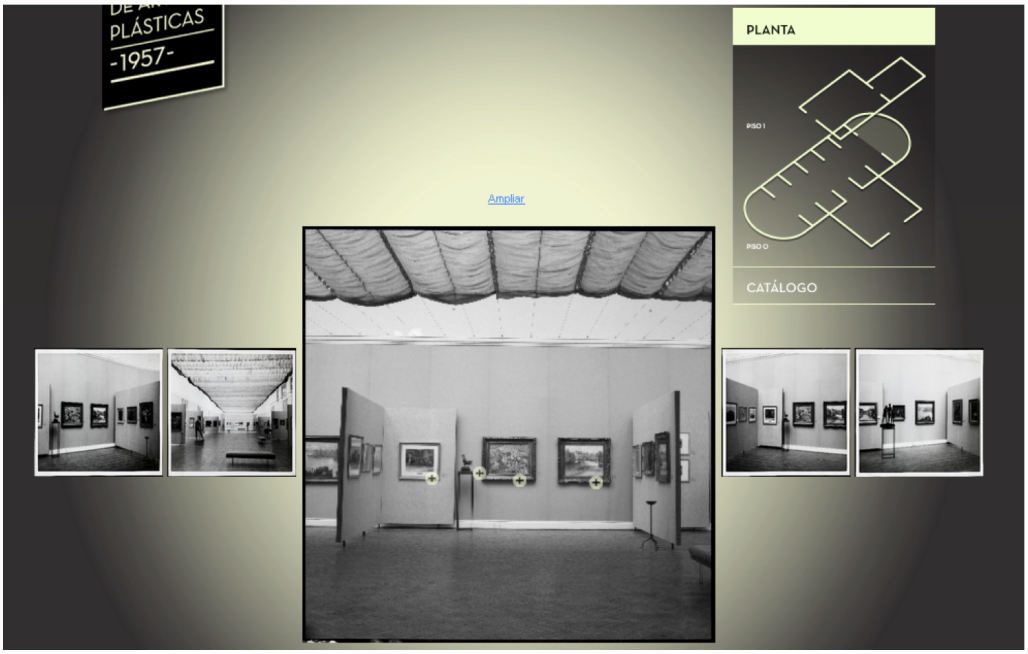


Figure 5.34: Strip navigation example. At the center, the current scene is shown with the two next and previous images of that room. At the top-right a navigation map gives direct room access. Pressing the (+) next to each artwork will present the detail page.

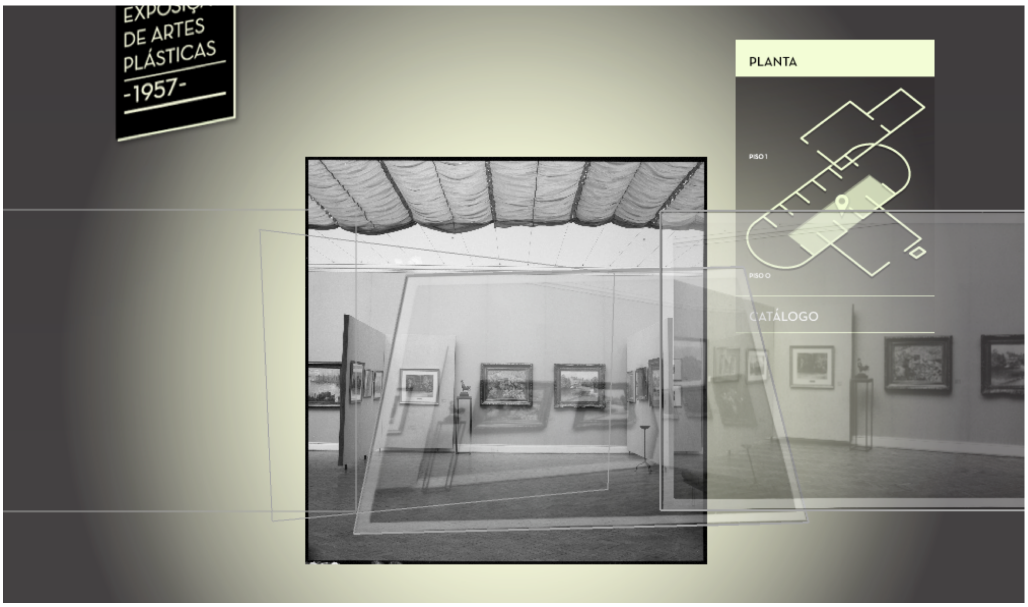


Figure 5.35: Panoramic navigation example, it has the same functionalities of the strip navigation but adds the navigation through the overlapping images seen in half transparency.

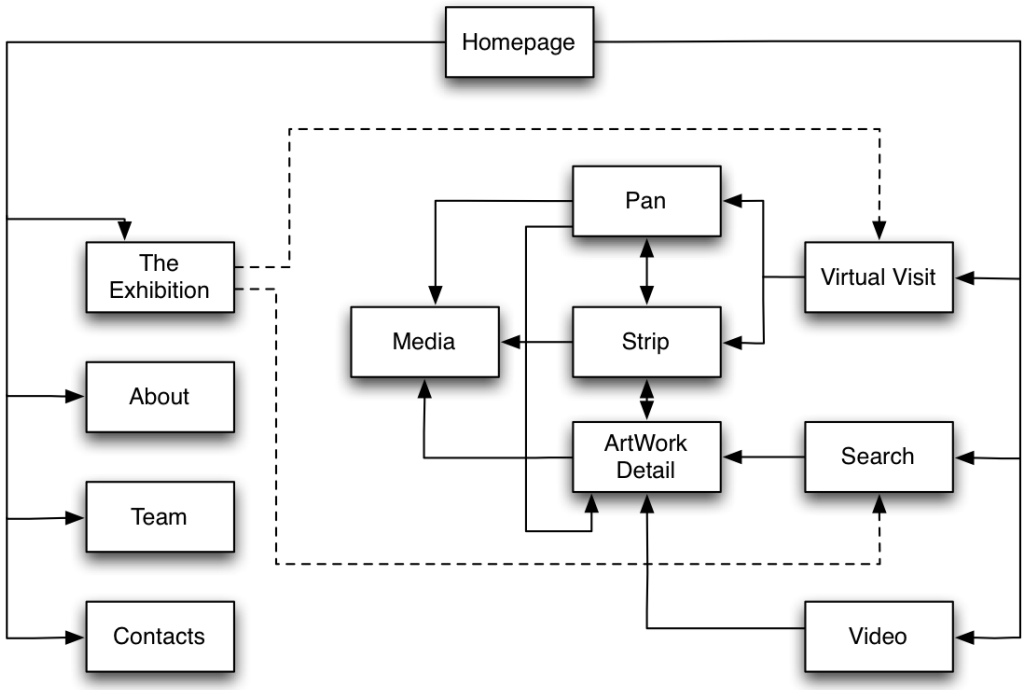


Figure 5.36: Navigation map of the user interface.

images is, as if, exploring a panoramic image, although the user can only see the points of view that are in the archive pictures As mentioned before, these two navigation modes are partly constructed with automatically detected spatial information. The goal with these navigation modes is to preserve the notion of space from the original exhibition. This provides contextual knowledge, which may be important for some historical researchers and for the public.

5.3.3.1 Interface Details

The virtual exhibition must be easily understandable by the visitor, so the navigation is supported by links that connect the main components. Figure 5.36 presents the navigation map of the exhibition Web application.

There are five main sections in which this exhibition is divided. The sections are: (1) Video, (2) Navigation, (3) Search, (4) Details and (5) Administration. The (1) video section contains media files that display, when running, dynamic information about what is shown in the movie at that moment. The (2) navigation section allows the user to visit the exhibition through pictures and observe the artworks. The (3) search for a specific word or artwork name is possible by consulting the search section where there are methods that enable fast access to information. The information that is going to be shown to the user will include some

Figure 5.37: Video links. Artworks in the table beneath highlight when they appear in the video. Some details about the artwork appear on the right with a link to the full detail panel.

specific details (4) where it is possible to obtain information about the current artwork, such as the artist name or the artwork title. In the (5) administration area there are options to control the artworks information and the navigation visualization.

The (1) video section, depicted in Figure 5.37, presents videos with contextual information about the filmed artworks. If one specific artwork is being shown in the video the correspondent information on the right panel will be displayed, allowing to obtain more detail about the artwork. It is also possible to navigate through the artworks that are in the table below the video. The table includes all the artworks that are present in the movie and when clicking on one of them the movie will jump to the instant where the artwork appears.

The (3) search panel (Figure 5.38) was constructed with the goal to provide information about the artworks in a straightforward fashion. Clicking on the desired artwork will redirect the user to the corresponding information detail section. The (4) detail panel presents information about any artwork in the system. This information will be displayed in a standard format for every artwork of the exhibition, as seen in Figure 5.39. In this panel it is possible to see the artwork image, in low resolution format and zoom to higher resolutions. There are also links below the image that indicate the rooms where this artwork is in. Clicking on one of these rooms, the system will redirect to that specific room in the

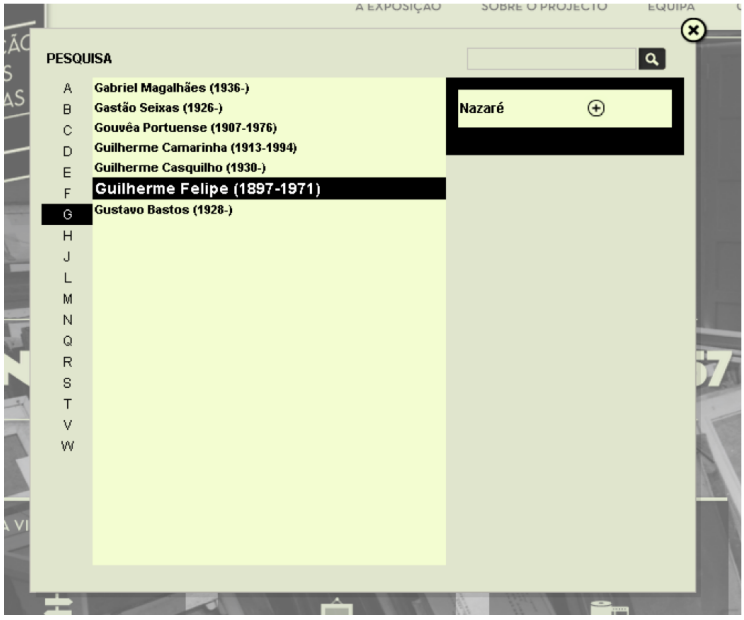


Figure 5.38: Artists search information panel. Search can be done alphabetically or using the text box. The right column presents the links to the artist's artworks (+).

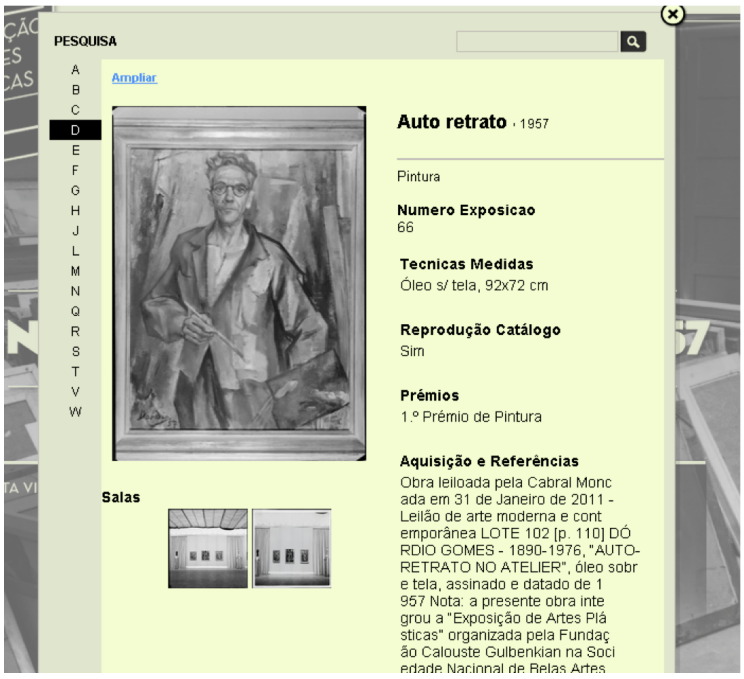


Figure 5.39: Artwork detailed information panel. Below, thumbnails link to the exhibition pictures where they appear and above a link provides access to a high-resolution image.

panoramic navigation mode. The (5) administration section has the control of specific sections of the exhibition and allows introducing all the information.

5.3.4 User Studies

The implemented prototype [Nó+12] presented in this section was constantly evaluated by a restricted group of expert users, namely historians, museum archivists and curators. The informal input was iteratively used to improve the interface and the proposed features.

The general reaction of the experts to the system is positive and enthusiastic. There is a general consensus in the field that the collected information from exhibitions and events held in museums should be archived and catalogued. The archiving process should take into consideration all media forms possible, including images, texts, maps and important location references. This information should be mostly digital or digitally available and searchable.

After several implementation iterations a more robust user study was conducted to evaluate the proposed interface. The main website user study (presented in detail in Nobre's M.Sc thesis [Nob12]) was performed with the help of 22 users aged between 18 and 60 years old with different backgrounds. The distribution of the genres was: female 36.4% and male 63.6%. All users had experience with Internet and web browsing.

The user study consisted in several exploratory tasks and a written questionnaire, presented in Table 5.3. Most of the user tests were conducted online. The questionnaire was divided into seven parts. The first five parts evaluated the main sections of the interface: Search, Detail, Video, Strip Navigation and Panoramic Navigation. The sixth part evaluated the entire website and the last part includes several user characterization questions.

Each part included a simple task to familiarize the user with the desired panel (e.g., "Find the painting X painted by Y"). After carrying out that task the user had to classify each statement in a 5-point Likert-based scale score. The scale has five options/choices and the extremes are classified as Disagree and Agree (with the statement). Additionally, whenever it made sense, there were several questions where the user had to introduce their preferences between several choices. In the end there was an open question to capture the opinion of the user about the system.

Results

The statements for each topic are presented in Table 5.3. This table presents

Table 5.3: Questionnaire. Statements with Likert-scale answers being 1 - Disagree and 5 - Agree. Median with Lower and Upper Quartile Deviation and Mean with Standard Deviation.

	Statement brief description	**Median**	**LQ**	**UQ**	**Mean**	**SD**
1	**Search Panel:**					
1.1	It is easy to find artwork X.	4	0	+1	4.23	0.81
1.2	Search results from the search box are the expected.	5	-1	0	4.50	0.74
1.3	Search panel is easy to understand.	4.5	-0.5	+0.5	4.45	0.60
2	**Artwork Detail Panel:**					
2.1	The artwork fields are clear and explicit.	4	0	+1	4.23	0.81
3	**Video Panel:**					
3.1	It is easy to go to the exact instant where the artwork X is shown.	3	-1	+1	3.00	1.31
3.2	You were aware that the detail information was being updated (at each artwork) when the video was running.	4	-1	+1	4.00	0.87
3.3	It is easy to understand the interface.	3.5	-0.5	+0.5	3.45	0.96
4	**Strip Navigation:**					
4.1	Is it easy to change the room photo.	5	-1	0	4.32	1.13
4.2	It is easy to change the room.	4	0	+1	4.14	0.99
4.3	It was simple to get the detail from each artwork.	4.5	-1.5	+0.5	4.23	0.87
4.4	Strip navigation was pleasant.	4	-1	0	3.73	0.77
5	**Panoramic Navigation:**					
5.1	The image overlap is simple and comprehensible.	4	-1	0	3.59	0.85
5.2	It was simple to consult artworks and navigate through the rooms.	4	0	0	3.77	1.15
5.3	This method is better than the Strip navigation.	3	0	+1	3.14	1.08
6	**Overall:**					
6.1	The web navigation quality is high.	4	-1	0	3.59	0.91
6.2	The historical content quality is high.	4	0	0	3.95	0.79

Table 5.4: Emotional engagement: Number of times each word was chosen. Users could choose an unlimited amount of words.

	#		#		#
Attractive	14	Frustrating	0	Stimulating	2
Complex	6	Confusing	3	Useless	0
Motivating	4	Satisfying	11	Simple	7
Boring	1	Irritating	0	Predictable	2
Impressive	0	Ideal	3	Addictive	0
Funny	8	Innovative	10	Tiring	1
Exciting	1	New	8	Immersive	1

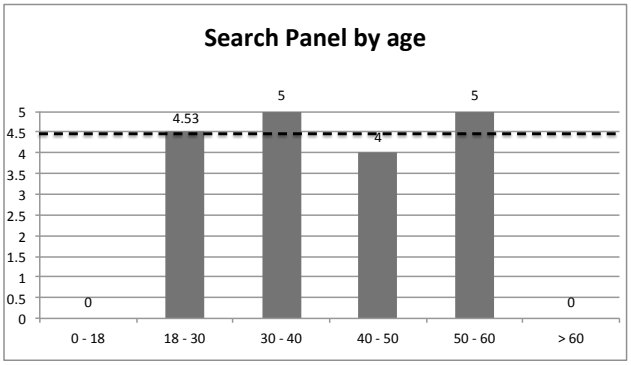
two central value measures, the median and the mean of the answers, which allow to evaluate what was the reaction to each statement. All users completed all the proposed tasks.

The final results were generally very good, but there were small differences in each section, allowing to say that some sections were more successful than others. Observing Table 5.3, the search panel was the most successful item with the highest score (statement 1.2) and the lowest deviation in answer (statement 1.3). The worst section was the Video section where it was not clear to most users that the information displayed on the side of the video was linked to the content of the video (statement 3.1).

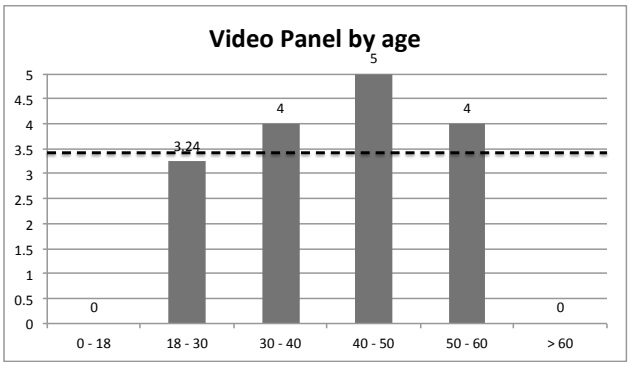
Comparing the Strip Navigation and the Panoramic Navigation, the first seems to have a better score revealing that probably the Panoramic Navigation needs to be visually simplified, but curiously when asked if the Panoramic is better than the Strip, more users chose the Panoramic although not by a large margin (statement 5.3). In this statement, the Likert inspired scale was modified to not directly map the numbers into the navigation styles. So the scale was Strip in one end and Panoramic in the other end (no numbers). The goal was to reduce the bias of the question. Many users reported that they were unfamiliar with the Panoramic Navigation and for that reason could not see the added value to it, although they enjoyed the smooth animated transitions between images.

The statements 1.3, 3.3, 4.4 and 5.2 evaluate directly each section. Figure 5.40 presents the results by age category. In the Search panel there is little variance but in the Video panel, younger users tend to expect a different type of interaction than older ones. In the Panoramic Navigation users between 40 and 50 give a lower score.

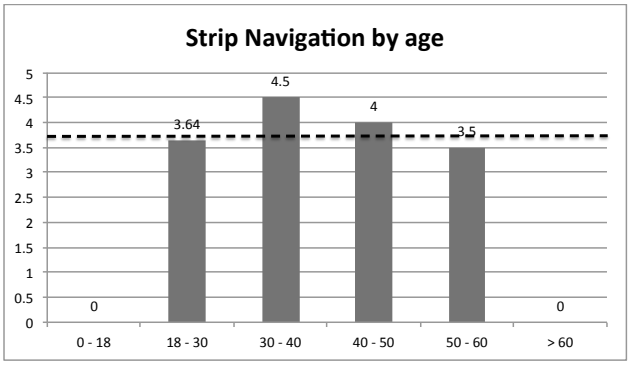
There were several different types of questions; in the detail section the users



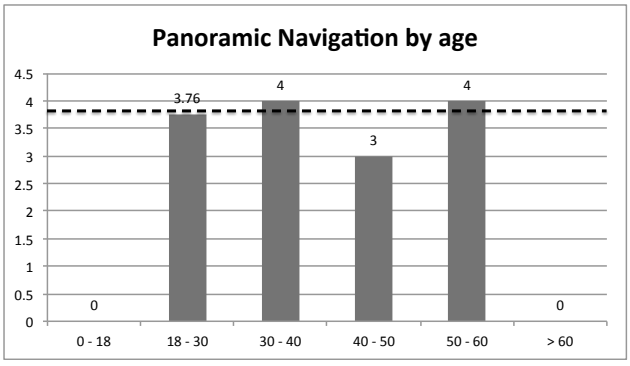
(a) statement 1.3



(b) statement 3.3



(c) statement 4.4



(d) statement 5.2

Figure 5.40: Evaluation of the Search, Video and Navigation panels by the users by age from 1 (not good) to 5 (good). The line represents the mean.

Figure 5.41: Most relevant adjectives describing the website (using Wordle.net).

were asked if they wanted other kind of zoom functions and what kind would they be interested in. 32% of the users answered that they were interested in a new zoom (most wanted a full screen zoom without distortions). In the general part, the users were asked about what kind of emotions the website triggered, using a subset of the Microsoft Reaction cards [BM02]. The results can be seen in Table 5.4 and are summarized in Figure 5.41. The best positive results were Attractive (14 times), Innovative (10) and Satisfying (11) and the most significant negative results (there were less negative than positive) were Complex (6) and Confusing (3).

Discussion

From this user study it can be concluded that the website is fully functional but some aspects need to be improved namely the Video panel interface. The Panoramic Navigation proved to be reliable, with the navigation through related images working as predicted. Additional work needs to be done in order to convey to the user what actions are possible. Some of the users were not familiarized with the quads, which represented the other images (images which overlap the current). For this reason many users did not use the full potential of the interface. From the reactions cards results summarized in Figure 5.41 it is possible to observe that the reaction is quite positive.

5.3.5 Lessons Learned

In this project, the semi-automatic recognition of relations between images and scene reconstruction were some of the most positive aspects referred by the expert users. For large datasets, automatic or semi-automatic reconstruction methods are an increasing necessity. Automating the process leaves more time and resources to focus on details, such as artwork information and cataloguing. The visualization of museum images in an order that highlights the original arrangement allows the exploration of the virtual exhibition with the visual codes and concepts of the original one. Experts can study the arrangement of the room, why certain artworks were placed next to each other at that time or how were the paintings presented. In our use-case, the original exhibition was held in a neo-gothic building from the beginning of the twentieth century. The search capabilities of the interface were also highly appreciated. Having a multimedia generic tool that allows the exploration of the archive by authors or artworks is important for researchers.

The building, shown in Figure 5.42, has many ornamental details but for the modern art exhibition that is being considered here, all walls, ornaments and painted windows were hidden behind fake walls. Some of the room layouts were different at that time; some of the historians and architects were surprised to discover this from the space-aligned photos. The construction of the map, that supports the navigation presented in subsection 5.3.3, was itself constructed after the semi-automatic organization of the photos. This was in part because of the lack of information of what rooms were used during the exhibition and also because of the constant changes that the interior of the building has suffered through the years.

The positive aspects of the system mentioned by the users were the automatic discovery of relations between images, the integrated media interface (combining images, video, information and scene reconstruction), the zoom ability (important for detail studying) and the standardized artwork detail panel. The custom made design of the Web application was also highly appreciated. The engaging front-end is important for the contextual immersion.

The negative aspects were essentially related with missing information from certain artworks, or the inexistence of high-resolution color images of certain pieces. Some fields were suggested to improve the standard artwork sheet and the given importance of the order in which they are presented was discussed. Although important, these are problems that concern essentially archivists. The current platform is intended to be a generic system for exhibition reconstruction,

Figure 5.42: Now and then: images of the museum from today and from the 1957 modern art exhibition.

there are always issues on each project that need to be addressed specifically. The current system proposes a set of common fields to describe artworks, but can easily be extended to accommodate extra information.

5.3.6 Related Projects

The research presented in this section is focused on the digital presence of virtual museums online, how artworks are presented and visualized, and what challenges museums face. Additionally, methods for image analysis and scene reconstruction were studied in order no enhance the visual interface model (section 3.2). The exploration of photos from the past was also subject of discussion. Some of the related projects presented in this section were already mentioned in the related work chapter (Chapter 2), others provide a deeper insight into the virtual museum's navigation systems.

The Google Art Project[10] is a project developed by Google in cooperation with several museums around the world. With the Street View technology it is possible to visit some of the largest art galleries in the world. The Street View [11] technology offers panoramic views, from many positions, of the real world streets. All the images presented by this system were captured by instrumented vehicles and with the help of other technologies covering the area around some specific place. This technology was adapted for museums by mounting the cameras and sensors into a small trolley. In some museums, such as the Museu Nacional de Arte Antiga[12] it is possible to make a virtual visit through the installations using

[10]Google Art Project, `http://www.google.com/culturalinstitute/project/art-project`.

[11]Google Maps Street View, `https://www.google.com/maps/views/`.

[12]Museu Nacional de Arte Antiga, National Ancient Art Museum, Portugal, `http://www.`

panoramic pictures, similar to the Street View method, where the user chooses the place to observe. One of the technologies commonly used is Krpano[13], a Flash plug-in, which allows the construction of virtual museum environments. The orientation cursors and the zoom buttons facilitate the walk through the virtual area. The navigation in this world is made inside a room selected from a list of rooms where each one is a section of different artworks types. Each room has a panel that shows the information related with it. The possibility of showing high-resolution artwork images is one of the factors to strengthen the exhibition presentation. Every picture has a panel showing information about it.

There are some developed projects that integrate the construction of interactive and virtual museums such as: the Hermitage Museum [Min+01], multi-touch interactive catalogue in museums [Cor+10], digital dossiers [Eli+07], the Info-Gallery [Grø+06] or the University of Ioannina web-platform [Sty07]. To model the construction of 3D exhibitions on the web Costagliola et al.[Cos+02] developed a tool that provides a domain specific language for virtual exhibitions.

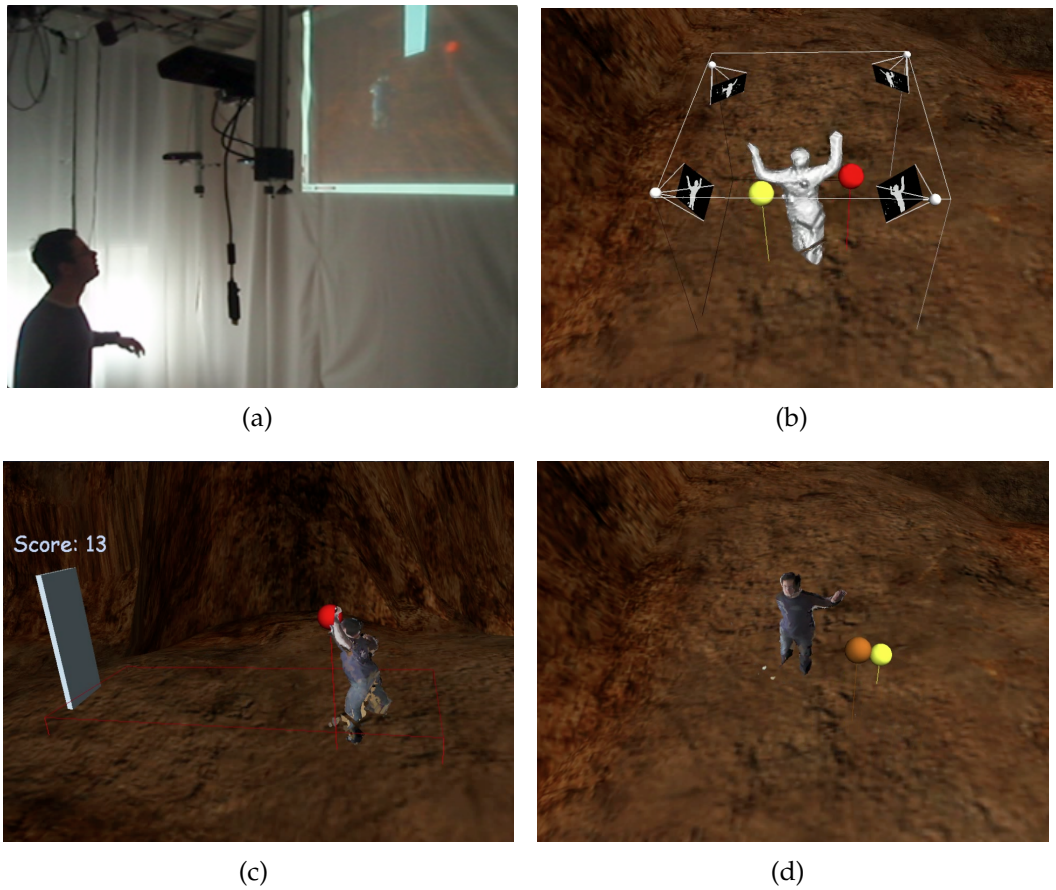
To analyze and visualize large sets of contextual images, several projects were observed as presented in sections 2.1 and 2.2. The creation of panoramic pictures through image stitching has been extensively studied and researched [BL06; IAH95; LHG08] as presented in section 2.1. A brief description of a solution to this problem will be given in the next section. Brown and Lowe [BL06] present a detailed survey of solutions and commercial products for photo-realistic panoramic images (section 3.2). There are also several commercial applications that use panoramic images . One of the most relevant projects in this area is PhotoTourism [SSS06; SSS08], where a large amount of unsorted photos are photo-stitched. This work was in the origin of PhotoSynth[14], an application that allows users to virtually explore a set of images by navigating in three-dimensions.

This prototype also supports the notion of temporal exploration, since its goal is to recreate a 50-year-old exhibition from old pictures. The process of computational re-photography, has been explored by Bae et al. [BAD10], and consists of using computer vision techniques to visually match ancient photos with recent pictures. The goal is to have a computer-assisted photo shooting system that helps the user to retake a photo with the same scenario of the ancient photo. The result is a photo overlap system with a time travel feature. This experience can be explored in the New York Changing [LYG04] project where re-photography is used to testify the passage of time in several streets of New York.

museudearteantiga.pt/.

[13]Krpano, panoramic Flash plug-in, http://krpano.com/ .

[14]PhotoSynth, http://photosynth.net.



(a) (b)

(c) (d)

Figure 5.43: Interactive OmniKinect prototype. Volume rendering based on the input of several Kinects. Extension for interaction between rendered volume and virtual objects.

5.4 Other Related Prototypes

Aside from the three main applications that were described, other related projects were addressed and some are still in execution. Some of this work was already briefly mentioned in section 1.5 and is detailed in this section.

Interactive OmniKinect

The research conducted at ICG - Institute of Computer Graphics and Vision[15] in the Technical University of Graz[16] in Austria, led to the development of an interactive application based on the OmniKinect system [Kai+12].

The OmniKinect is a system, developed at ICG, where a room is equipped with several depth sensing cameras (e.g., Microsoft Kinect) with the goal of acquiring in three dimensions all the objects and persons in a room. The result of

[15]ICG, `http://www.icg.tu-graz.ac.at/`
[16]TUGraz, Technical University of Graz, `http://www.tugraz.at/`

the 3D acquisition can be rendered in real-time in a 3D application. The 3D acquisition results from merging the 3D depth information from each Kinect. The room is currently equipped with 8 Kinects, and in the example in Figure 5.43(c), 4 Kinects are used.

The OmniKinect [Kai+12] was originally designed for volume rendering of real objects, and it represents an excellent example of scene acquisition using depth sensing cameras. To showcase the full capabilities of the system, an Interactive application was developed in the context of this dissertation. The Interactive OmniKinect game, seen in Figure 5.43, allows the interaction of the volume rendered user, with virtual objects in the virtual scenario.

The setup of the system includes a room with the OmniKinect system of calibrated Kinects and a large projector where the user can see the virtual world. Figure 5.43(a) presents the user interacting in the room, Figure 5.43(b) presents the detected volume render of the user, while Figures 5.43(c) and 5.43(d) present a textured version of the volume render in the game.

The game is very simple, but very effective to demonstrate the system. The user hits the ball against an obstacle. The ball bounces back in a virtual box. A physics engine gives extra realism to the ball interaction. Whenever the ball hits the obstacle the score increases. Since the room is completely covered by the cameras, the interaction can be done from all sides.

This project demonstrated the feasibility of creating interactive multimedia applications using a scenario captured by multiple cameras with depth sensors. From an application point of view it presents several interaction challenges. Different users react differently when they see themselves in the third person. The user is captured from all sides with much detail. For the future, the goal is to understand how can this be used in interactive applications and what kind of interaction is interesting between the real world and the virtual world.

Logo detector

Other interesting project, related with this dissertation, is the creation of a smartphone application to detect known logotypes or commercial brands. The prototype seen in Figure 5.44 was implemented using the technologies described in sections 3.1 and 3.2.

The prototype uses a dual-core 1Ghz phone running Android OS to automatically detect in real time a pre-programmed logo. This can be useful to create interactive application with the involvement of real world objects associated with commercial brands. The application, presented in Figure 5.44, is currently being developed for commercial purposes.

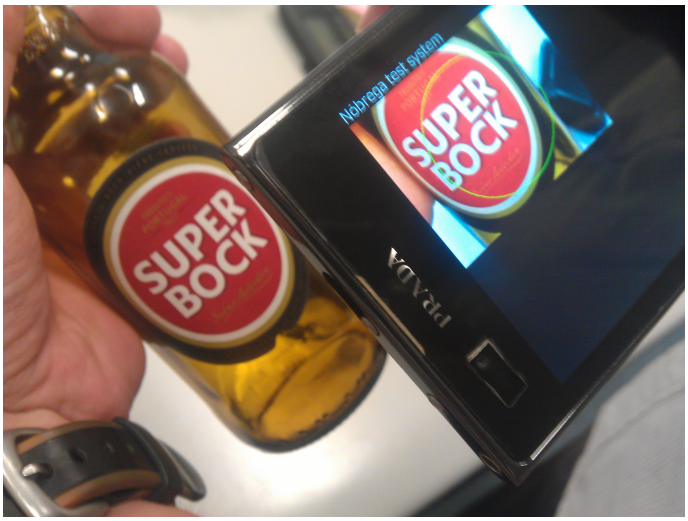


Figure 5.44: Real-time logo detection using an Android smartphone.

SEN: Shareable, Editable and Navigable Spaces

Having a framework where virtual objects can be added in images (Chapter 4) means that there are multiple applications possible, especially in areas where traditional photography is important. An interesting concept would be the edition of photographs from spaces belonging to friends and relatives.

The advent of the Internet and especially social networks brought a new culture of sharing personal aspects of ones life, which were previously reserved to a inner circle of close friends and relatives. The level of sharing varies from person to person, but usually involves images, videos and texts. Texts appear in the form of annotations in images, comments, blogs or micro-blogs. Images and videos are essentially non-interactive media, people can comment on, but not interact with it. Bringing interactivity to user-shared elements such as photos would foster new forms of collaboration and socialization that are currently not available. This interactivity could be the introduction of virtual elements and objects in other people's photos [NC12e].

There are many motivations for sharing photos; one of them is showing a certain physical space to others. This can be an holiday spot, a part of the new house or just an interesting place. However, sometimes the users may want more, they may want to share a place and allow it to be editable and explorable by others. This can correspond to asking an opinion about a place or to do cooperative work. A photo is a single instantaneous revelation of a certain point of view in time. Sharing a virtual space, could be more interesting in the sense that it could be explored in three-dimensions or from several points of view and in different temporal time-frames. The roles of time, place, and the relation between photos

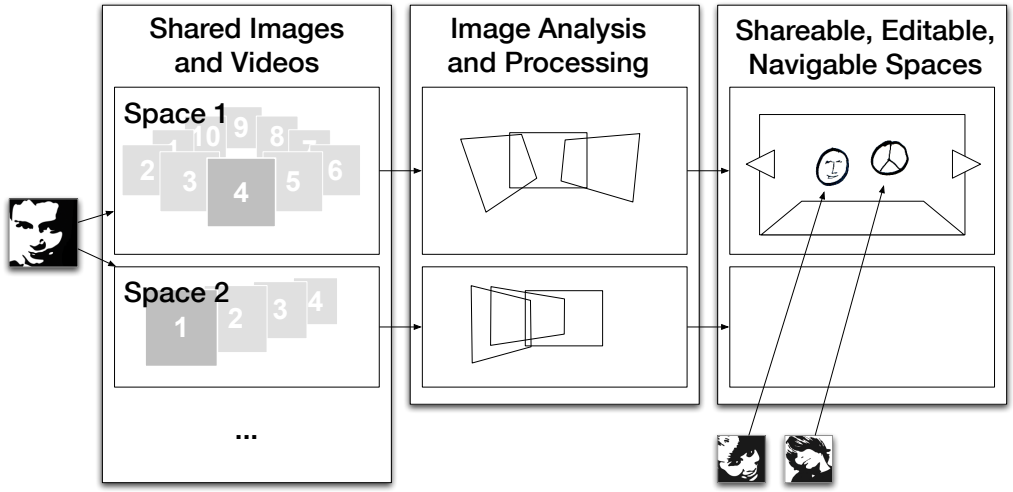


Figure 5.45: Design architecture of the system. Using the images and videos of a certain place, shared by the user it is possible to create the concept of interactive spaces. The user and his or her friends could edit these spaces and superimpose them with virtual objects that adapt to the content of the image.

are explored in Stelmaszewska et al.[SFB08].

Imagine a world with exploratory spaces where it is possible to see back in time, where objects can be added to the scenario and the result shared with others. The concept proposed [NC12e] is an approach for sharing in social networks, using a different type of media, interactive places represented with panoramas or 3D images from user generated content. Most participative mechanisms currently available in social networks are positive and negative reinforcements (like buttons), comments on forums and the act of propagating the content through the network. In the shared places the users would be allowed to graphically reshape a physical place collaboratively.

There is a large number of applications that allow social interaction with editing and sharing of virtual places, one of them being Second Life[17]. Although the interaction is performed in visually rich environments, mainly created by users, these environments were not created automatically from user content such as photos. The automatic creation of an editable environment based on user photos is one of the main goals of this work.

The main requirements for these interactive places are that they should be explorable, editable and shareable, as detailed in Figure 5.45.

The exploration should allow some form of change in the point of view so that users can see the surroundings of the main scene or observe the scene from different angles to get a better idea of the perspective. The level of exploration is

[17]Second Life,Virtual world application, `http://secondlife.com/`.

directly related with the number of photos and videos available of that specific place. The system should be based on data that can be collected by a common user with no special skills using regular cameras. Since the photos can be from the same place but of a different time, a temporal exploration could also be important.

Scene editing should be done with traditional elements such as text labels and annotations on the scene. Additionally, it should be possible to add graphical elements such as images or three-dimensional objects that remain attached to scene elements, following them when the scene changes. The three-dimensional objects positioning by the user should take into consideration the three-dimensional structure of the scene, as in the Magnetic Objects concept previously detailed in section 5.1.

Finally, the interactive places should be shareable with the users' friends. By creating a web-based application, the interaction and exploration can be done with additional users that can enrich the interactive places with their own generated content. Also, it is a much more interesting paradigm, where users can create their own versions of other places, giving their own view of how it should look like.

This concept requires a large number of photos from a certain place. Most of this data is already available on current social networks. As explained in Figure 5.45, the user takes a large amount of photos and videos of a certain place. Then the photos are pre-processed to detect relations, visual elements and three-dimensional structure. Finally the interactive places are available for other users to explore, create alternative versions with virtual content or edit collaboratively the scene.

This application concept presents a different type of shareable media: interactive places that can be navigated, edited and shared. These can be achieved using single images through the use of the Magnetic Objects concept and the framework for virtual insertion of objects (Chapter 4), or through multiple images using photo-stitching or stereo vision techniques (Chapter 3).

The main novelty of this concept proposal is a framework that empowers the user to virtualize a certain place and share it online, with all the social implications of instantly sharing these interactive and sometimes intimate places [NC12e].

5.5 Discussion

In this chapter several applications were presented and studied, with special focus for the two applications that derive directly from the framework presented in Chapter 4. These are the applications for the insertion of virtual objects in photos: Magnetic Augmented Objects in Photos and Mixed Reality Snake.

In the Magnetic Augmented Objects application (section 5.1), the user can configure different settings and parameters to obtain the best interaction results. It has several image capturing methods to obtain different input images, and an interactive workflow, which allows recapturing and repeating the process quickly. This implementation proves the feasibility of the creation of interactive mixed reality applications (**RQ1**). The Magnetic Objects concept is demonstrated, showing that virtual objects can interact with the properties of the scenario (**RQ2**). The user studies show that the users can have a performance increase with this concept and are able to use the proposed technology (**RQ3**).

The Mixed Reality Snake game (section 5.2), took the Magnetic Objects to another level with a coherent user interface guiding the user through the initialization of the mixed reality system. The studies show that the users were able to initialize the system themselves and, with the correct interface, were motivated to use the mixed reality system (**RQ3**).

Analysing the combined results from the users studies from these applications (sections 5.1.5 and 5.2.3), there is a high degree of satisfaction with the idea of using objects inside photos and users are satisfied with the implementation, especially in the latest snake game.

5.6 Summary

The main goal of this chapter was to present several interactive applications that take advantage of the spatial acquisition of information from images and cameras. Three main applications were implemented and were thoroughly described. Additionally, several application concepts and small prototypes were presented. The goal of these applications was to test the concepts and algorithms presented in the previous chapters.

The first two sections, present direct implementations of the single image based framework presented in Chapter 4. Section 5.1 introduces the Magnetic Augmented Objects concept with an application which implements its principles. The following section, presents a mixed reality snake game, designed to explore

the full capabilities of the framework. It contains a guided initialization method, to capture and annotate a photo, which is then used as input for a competitive two-player game.

The sections 5.3 and 5.4 present alternative techniques to the acquisition and presentation of a certain space through images and cameras. The reconstruction of a past museum exhibition (section 5.3), presents a navigation system based on the analysis of several images. The interactive navigation is an interface based on the spatial relation between each image and presents augmented links above each painting. The construction is semi-automatic, partially fulfilling **RQ1**. The last section briefly presents additional implemented application with special focus in the interactive space acquisition based on special hardware, such as depth sensing cameras, and in the shareable space exploration.

6

Conclusions

The main challenge of this work was to study solutions for interactive applications, which acquire a certain scene from photos captured by the users, and seamlessly introduce virtual elements in the same scene. The proposed solution was to detect high-level features in the photographed scenario that could be used to assist the insertion of virtual objects in the context of the scene.

In this thesis, the main research problems (section 1.1) are approached from several different perspectives and techniques. The proposed solution is presented in Chapter 4. It is a framework for the semi-automatic analysis and insertion of virtual objects in a single image. This framework can analyze a photo captured by the user and detect high-level features such as vanishing points, floor and room orientation. Using these features it is possible to create interactive applications in which the behavior depends on certain interactions with the scenario (e.g., implement 3D gravity and bouncing of objects against the floor).

Using the mixed reality framework described in Chapter 4, two main applications (Chapter 5) were implemented. The first, presented in section 5.1, introduces and demonstrates the concept of Magnetic Augmented objects which interact with the photographed scenario. The second, presented in section 5.2, is an interactive application, a "Snake" game in mixed reality. This game demonstrates the possibilities of the proposed system, which can be explored in interface and interaction design. The game takes advantage of the detected high-level features to simulate virtual objects (snakes) that have physical properties, such as traversing the floor in three-dimensions and colliding with furniture and walls.

Beyond the main mixed reality system based on single photo analysis, other alternatives were explored and some implemented. The most relevant alternative was the construction of a virtual museum exhibition, which occurred in the past, using several archive photos. This application presented in section 5.3, explores the automatic virtualization of spaces from several photos. It is a website with a panoramic navigation system built by finding automatic relations between images in the photo set, using matching features. Other different approaches were studied using different input paradigms such as stereo vision (subsection 3.5), video input (subsection 3.4), 3D depth cameras and smartphone camera interaction (both in subsection 5.4). The complete list of contributions and deliverables was already detailed in sections 1.3 and 1.5.

6.1 Main Results

Recalling the research questions from section 1.1, the main research question, **RQ**, was answered from several perspectives as stated before. The main proposed solution to **RQ** was the mixed reality framework developed in this thesis and explained in detail in Chapter 4.

From the broader problem statement three research questions were defined. In the first research question, **RQ1**, the main goal was to provide answers about the feasibility of creating interactive mixed reality applications where virtual objects interacted on non pre-defined real world scenarios. The mixed reality framework described in Chapter 4 presents a solution that can be used in mixed reality applications. It is based in single photos, which can be captured in any non pre-defined scenario without any special marker in the real world. Results presented in section 4.4 show a high degree of detection (in Manhattan scenes) of the image's vanishing points which are essential to detect the orientation of the scene. The system's redundancy analysis with multiple combinations of parameters increases the detection success under several conditions, as demonstrated in subsection 4.4.1 where the algorithm was tested with an external image database. The framework's high level features can be used to create several applications in which virtual objects interact with the captured non-predefined scenarios. The virtual museum navigation system (section 5.3) provided an example of how a non-predefined space can be virtualized using several photos.

The second research question **RQ2** was related with the possibility of virtual objects simulating real world properties in photographs. The mixed reality applications presented in sections 5.1 and 5.2 demonstrate examples of virtual objects

that interact with the detected photo scenario. The magnetic augmented objects (section 5.1) interact with the floor and the limits of the floor. The snakes from the mixed reality Snake game (section 5.2) interact with the walls and obstacles in the game area.

Finally, the third research question, **RQ3**, is related with the users and the interface required for them to interact with the mixed reality system. The mixed reality Snake application (section 5.2) demonstrated that it is possible to implement engaging applications using the framework presented in Chapter 4. The two user studies presented in subsection 5.2.3 show a high level of user satisfaction. More importantly the studies demonstrated that users were able to initialize the mixed reality system on their own and were motivated enough to do it several times. The user studies show that the users are wiling to use such a system especially in the game setup, meaning that the content of the application matters. From the same study, it can be concluded that using a small set of instructions, the users can create the scene model using cameras (with a short learning curve). The study with the Magnetic Objects (subsection 5.1.5) demonstrated that these can be used to improve the user performance by aiding in the positioning of objects in 3D in photographs.

6.2 Discussion

Many implications result from the presented work. The mixed reality framework presented in Chapter 4 has many applications with several benefits and some limitations.

The main benefits of the proposed system are related with the fact that it can be used in any space and without any marker or annotation in the real world. There are several possible capturing methods, which can allow interaction with photos, which were collected before, or are instantly acquired from a physical space. Using pre-captured photos allows the creation of applications which interact with old archive photos or photos from holidays.

A single image based system has the advantage of having a separate acquisition, processing and interaction workflow. This means that users can capture now and interact later or use captured photos for extended and comfortable interaction as opposed to real-time AR systems where users must maintain the camera pointed to a target while interacting.

The high-level of detection in several illumination conditions means that the proposed system is ready for interactive multimedia applications that are tolerant

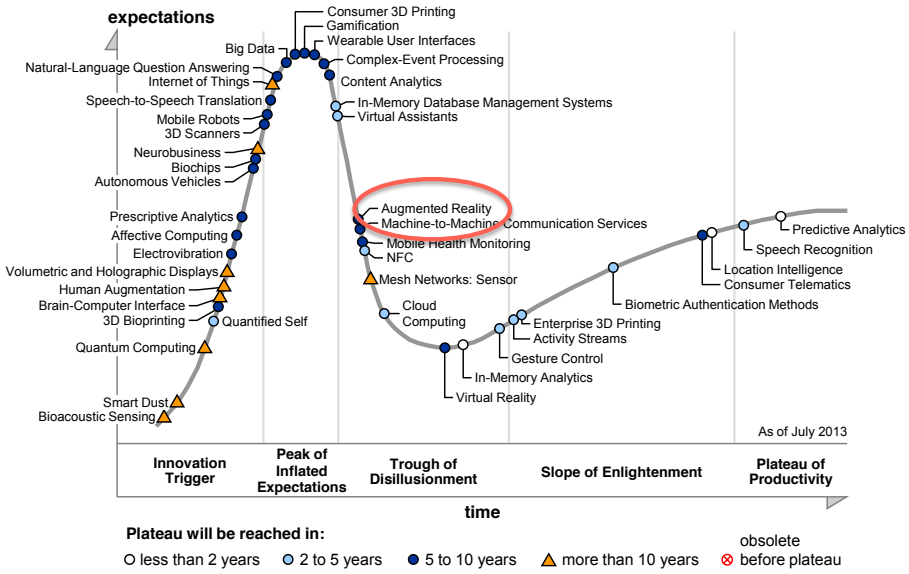


Figure 6.1: The 2013 Emerging Technologies Hype Cycle by Gartner Inc.. (Source: Gartner Inc. `http://www.gartner.com/`)

to some degree of mismatch (not to be used in critical applications) where the user can be asked for another photo if the system does not work.

The present system has, due to its nature, several limitations. The single image input system, means that there are no other points of view from which to extract additional information. This means that it is very difficult to deal with occluding objects unless they are manually annotated. Another important aspect already mentioned, is that this system does not detect the space in real-time, meaning that the user needs to re-initialize the process when it is necessary to change the point of view of the scene. Finally the detection system, can detect multiple scene elements through manual annotations. Annotations will increase the realism of the detection but make the initialization more difficult for the user.

6.3 Future Directions

The acceptance by the general public of mixed and augmented reality applications has traditionally been difficult. Users are generally very curious about the technology but after the first impact of surprise they usually (with some exceptions) ignore the applications. Figure 6.1 presents a chart that represents the current development stage of several technologies. It can be seen that Augmented Reality (the closest technology to Mixed Reality) after an initial hype, mostly related with the appearance of ARToolKit [ART03], is currently in the Trough of Disillusionment. This means that the technology has matured but the interest in

it has diminished. What also can be observed is that it can enter the Slope of Enlightenment at any moment with the increase of reliable applications which deliver consistent and useful results. Currently, new tools, libraries and software development kits are appearing commercially [Met13; Vuf13] with new developments in real-time 3D detection of scenarios and superimposition of virtual objects. These developments will increase the opportunities for the development of interactive mixed and augmented reality applications making this technology probably enter the Slope of Enlightenment. Mixed reality applications will also improve with the appearing of new solutions for the living room where televisions have cameras or are associated with game consoles which have high-definition cameras. These interactive systems will probably have a larger impact in entertainment and gaming applications while mobile applications can have additional impact in advertising products and physical spaces.

Mixed reality systems where users interact with single photos as in the proposed solution are less common than their augmented reality counterparts, but can equally be important in entertainment, design and architecture applications. The main advantage, already stated, is that the interaction can be much more focused on the virtual objects inside the real world photo without the additional complication of carrying the smartphone/tablet/webcam and pointing it to some specific place. The interaction with a single photo can be more relaxed with careful attention to details. This characteristic is its main strength, users experimenting objects in a room, desire precision while calmly interacting without the pressure of real-time video. Probably an interesting idea would be to combine the two types of systems. In certain applications it would be useful to acquire a certain place by moving around as usually in AR, while having a freeze functionality where users could edit virtual objects in a single frame image. As previously stated in the mixed reality related work section (section 2.3), there are several applications especially from design, architecture and photo editing, which explore these ideas.

Using the research presented in this thesis there are many possibilities for the future. Some are questions which were left open in this work while others are opportunities which arise from it. Future work includes:

Detect additional single-image high-level features. There are many additional elements that can be detected and incorporated in the mixed reality applications. These include walls, polygons, shadows, textures, colors or specific objects. Many of these objects or elements can be detected with additional annotation. This includes occluding objects, cluttering objects or

empty spaces.

Application development. The current developed SnakeAR application is an application which showcases the framework. Using the information from the floor and the room orientation it is possible to create many more interactive applications for entertainment, advertising or design.

Detect scale in scenario. The automatic detection of scale of the real world is one of the main problems of mixed reality systems, which do not depend on a physical marker with known size. Possible solutions to this problem can be the automatic detection of known elements or scene objects from a knowledge base, use additional hardware or introduce the user in the loop by allowing the annotation of the size of an element.

Improve computer graphics elements. The current prototypes were created with special focus on geometric correctness and engaging interactivity. The computer graphics quality is minimal. An interesting research path would be on finding ways to improve the quality of the textures, illumination and shadows of virtual objects in order to blend them with the photograph.

Extend framework to mobile platforms. The current implementation is based on regular desktop computers. It is geared for image input captured from webcams, photographic cameras and disk files. Extending this work to mobile platforms will concentrate the acquisition, processing and interaction process in a single device.

Improve system detection. The presented algorithms work in most of the expected situations but can still be improved, especially in non Manhattan scenarios. A possible solution could be using a prior knowledge-base to have different detection methods according with the type of image (e.g., indoor vs outdoor). Another possible solution could involve additional annotations by the user.

Improve input method. The current input method is based on a single picture. Many times the captured picture is not ideal because it has not enough interaction space or it is blurred. It would be interesting to have a video based capturing interface where the system automatically chooses, which of the images is best. The system could even benefit with the additional three-dimensional information acquired by moving the camera.

Automatically augmented photos This implemented framework can be used to add virtual objects in 3D context when a photo is captured with a camera. This can be used to add automatic labels or interesting virtual objects (e.g, gnomes) to create augmented photos automatically.

Rephotograpy, explore photos from the past. This system detects the structure of a scene independently from when it was captured or what method it was used. Vanishing points can be detected in paintings, black and white photos and technical drawings. This means that virtual objects can be used to alter or enhance the reality of archive photos. There can be a redefinition of the past or a reconstruction of old scenarios with virtual content.

In this research, the main aspects of introducing virtual content in photos have been presented. Using the presented techniques it is possible to design and develop different interactive applications based on images.

Bibliography

[Aga+11] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. "Building Rome in a day". In: *Communications of ACM* 54.10 (2011), pp. 72–79.

[AKR11] C. Arth, M. Klopschitz, and G. Reitmayr. "Real-time self-localization from panoramic images on mobile devices". In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 55–64.

[ART03] ARToolKit. `http://www.hitl.washington.edu/artoolkit/`. (last access October 2013). 2003.

[AZK08] S. Ay, R. Zimmermann, and S. Kim. "Viewable scene modeling for geospatial video search". In: *Proceedings of the 16th ACM international conference on Multimedia (MM '08)*. Vancouver, BC, Canada: ACM, 2008, pp. 309–318.

[Azu97] R. Azuma. "A survey of augmented reality". In: *Presence-Teleoperators and Virtual Environments, MIT Press* 4 (1997), pp. 355–385.

[BAD10] S. Bae, A. Agarwala, and F. Durand. "Computational rephotography". In: *ACM Transactions on Graphics (TOG)* 29.3 (2010), pp. 1–15.

[BSS11] S. Bao, M. Sun, and S. Savarese. "Toward coherent object detection and scene layout understanding". In: *Image and Vision Computing, Elsevier* 29.9 (2011), pp. 569–579.

[BDS10] L. Baronti, M. Dellepiane, and R. Scopigno. "Using Lego pieces for camera calibration: a preliminary study". In: *Proceedings of Eurographics Conference 2010*. Norrköping, Sweden: Eurographics Association, 2010, pp. 97–100.

[Bay+08] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. "SURF: Speeded up robust features". In: *Computer Vision Image Understanding, Elsevier* 110.3 (2008), pp. 346–359.

[BM02] J. Benedek and T. Miner. "Measuring desirability: new methods for evaluating desirability in a usability lab setting". In: *Proceedings of the Usability Professionals Association Conference 2002*. Orlando, USA, 2002.

[Ben11] B. Benfold. "Stable multi-target tracking in real-time surveillance video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Vol. 1. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 3457–3464.

[BVZ99] Y. Boykov, O. Veksler, and R. Zabih. "Fast approximate energy minimization via graph cuts". In: *Proceedings of the IEEE 7th International Conference on Computer Vision (ICCV'99)*. Kerkyra, Greece: IEEE Computer Society, 1999, pp. 377–384.

[BL06] M. Brown and D. G. Lowe. "Automatic panoramic image stitching using invariant features". In: *International Journal of Computer Vision, Springer* 74.1 (2006), pp. 59–73.

[Bun+12] P. Bunnun, D. Damen, A. Calway, and W. Mayol-Cuevas. "Integrating 3D object detection, modelling and tracking on a mobile phone". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 273–274.

[Cal+10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "BRIEF: Binary robust independent elementary features". In: *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*. Hersonissos, Crete, Greece: Springer-Verlag, 2010, pp. 778–792.

[Can86] J. Canny. "A computational approach to edge detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 (1986), pp. 679–698.

[CKD11] Y. N. Chang, R. K. C. Koh, and H. B. L. Duh. "Handheld AR games: a triarchic conceptual design framework". In: *Proceedings of the IEEE International Symposium On Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 29–36.

[CS08] B. Chen and P. Sen. "Video carving". In: *Proceedings of Eurographics Conference 2008*. Hersonissos, Crete, Greece: Eurographics Association, 2008, pp. 63–66.

[CLL06] L. H. Chen, Y. C. Lai, and H. Y. Liao. "Video scene extraction using mosaic technique". In: *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR'06)*. Hong Kong, China: IEEE Computer Society, 2006, pp. 18–21.

[CLY09] W. T. Chu, C. H. Lin, and J. Y. Yu. "Feature classification for representative photo selection". In: *Proceedings of the 17th ACM international conference on Multimedia (MM '09)*. Beijing, China: ACM, 2009, pp. 509–512.

[CM09] K. Cordes and O Muller. "HALF-SIFT: High-accurate localized features for SIFT". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. Miami, FL, USA: IEEE Computer Society, 2009, pp. 31–38.

[Cor+10] N. Correia, T. Mota, R. Nóbrega, L. Silva, and A. Almeida. "A multitouch tabletop for robust multimedia interaction in museums". In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS'10)*. Saarbrücken, Germany: ACM, 2010, pp. 117–120.

[Cos+02] G. Costagliola, S. D. Martino, F. Ferrucci, and F. Pittarello. "An approach for authoring 3D cultural heritage exhibitions on the web". In: *Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE '02)*. Ischia, Italy: ACM, 2002, pp. 601–608.

[CY99] J. M. Coughlan and A. L. Yuille. "Manhattan world : compass direction from a single Image by bayesian inference". In: *Proceedings of the International Conference on Computer Vision (ICCV '99)*. Vol. 2. Kerkyra, Greece: IEEE Computer Society, 1999, pp. 1–10.

[Cra+11] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. "Discrete-continuous optimization for large-scale structure from motion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 3001–3008.

[CRZ00] A. Criminisi, I. Reid, and A. Zisserman. "Single view metrology". In: *International Journal of Computer Vision, Springer* 40.2 (2000), pp. 123–148.

[DTM96] P. E. Debevec, C. J. Taylor, and J. Malik. "Modeling and rendering architecture from photographs : a hybrid geometry and image-based approach". In: *Proceedings of the 23rd annual conference on Computer Graphics and Interactive techniques (SIGGRAPH'96)*. New Orleans, LA, USA: ACM, 1996, pp. 11–20.

[Del+11] L. Del Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard. "Sampling bedrooms". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 2009–2016.

[DB09] A. Delong and Y. Boykov. "Globally optimal segmentation of multi-region objects". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 285–292.

[Dix+03] A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*. 3rd. Prentice Hall, 2003, p. 834.

[DR08] P. Dragicevic and G. Ramos. "Video browsing by direct manipulation". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI'08)*. Florence, Italy: ACM, 2008, pp. 237–246.

[Eli+07] A. Eliëns, Y. Wang, C. van Riel, and T. Scholte. "3D digital dossiers – a new way of presenting cultural heritage on the Web". In: *Proceedings of the twelfth international conference on 3D web technology (Web3D '07)*. Vol. 1. 212. Perugia, Umbria, Italy: ACM, 2007, pp. 157–161.

[FSB09] J. Fernquist, G. Shoemaker, and K. S. Booth. "'Oh snap'– helping users align digital objects on touch interfaces". In: *INTERACT'11 Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction*. Uppsala, Sweden: Springer-Verlag, 2009, pp. 338–355.

[FB81] M. A. Fischler and R. C. Bolles. "Random sample consensus: A paradigm for model Fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[FG11] P. Fite-Georgel. "Is there a reality in industrial augmented reality?" In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 201–210.

[Fol+95] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C*. Addison-Wesley Professional, 1995, p. 1200.

[For13] D. Forsyth. "Understanding pictures of rooms". In: *Communications of the ACM* 56.4 (2013), p. 91.

[FP02] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[Fur+09a] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. "Manhattan-world stereo". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. Miami, FL, USA: IEEE Computer Society, 2009, pp. 1422–1429.

[Fur+09b] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. "Reconstructing building interiors from images". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 80–87.

[Gau+12] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Hollerer. "Live tracking and mapping from both general and rotation-only camera motion". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 13–22.

[GPM10] J. Gimeno, F. Pardo, and P. Morillo. "A mobile augmented reality system to enjoy the Sagrada Familia". In: *Proceedings of Eurographics Conference 2010*. Norrköping, Sweden: Eurographics Association, 2010, pp. 57–64.

[Gio+08] R. G. V. Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. "On straight line segment detection". In: *Journal of Mathematical Imaging and Vision, Springer* 32.3 (2008), pp. 1–45.

[GJR07] R. von Gioi, J. Jakubowicz, and G. Randall. "Multisegment detection". In: *Proceedings of the IEEE International Conference on Image Processing(ICIP'07)*. San Antonio, TX, USA: IEEE Computer Society, 2007, pp. 1–4.

[GFK09] S. Gould, R. Fulton, and D. Koller. "Decomposing a scene into geometric and semantically consistent regions". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 1–8.

[Gra+12] R. Grasset, T. Langlotz, D. Kalkofen, M. Tatzgern, and D. Schmalstieg. "Image-driven view management for augmented reality brows ers". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 177–186.

[Grø+06] K. Grønbæk, A. Rohde, B. Sundararajah, and S. Bech-Petersen. "InfoGallery : Informative art services for physical library spaces". In: *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries (JCDL '06)*. Chapel Hill, NC, USA: ACM, 2006, pp. 21–30.

[GDH11] R. Guo, Q. Dai, and D. Hoiem. "Single-image shadow detection and removal using paired regions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 2033 –2040.

[Gup10] A. Gupta. "Blocks world revisited: Image understanding using qualitative geometry and mechanics". In: *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*. Vol. 125. Lecture Notes in Computer Science 1-2. Hersonissos, Crete, Greece: Springer-Verlag, 2010, pp. 482–496.

[Gup+11] A. Gupta, S. Satkin, A. Efros, and M. Hebert. "From 3D scene geometry to human workspace". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 1961–1968.

[GWCO09] M. Guttmann, L. Wolf, and D. Cohen-Or. "Semi-automatic stereo extraction from video footage". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 136–142.

[HS88] C. Harris and M. Stephens. "A combined corner and edge detector". In: *Proceedings of the 4th Alvey Vision conference*. Manchester, UK, 1988, pp. 147–152.

[HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004, p. 672.

[HH09] V. Hedau and D. Hoiem. "Recovering the spatial layout of cluttered rooms". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 1849–1856.

[HHF12] V. Hedau, D. Hoiem, and D. Forsyth. "Recovering free space of indoor scenes from a single image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. Providence, RI, USA: IEEE Computer Society, 2012, pp. 2807–2814.

[HS05] A. Hertzmann and S. M. Seitz. "Example-based photometric stereo: shape reconstruction with general, varying BRDFs." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 27.8 (2005), pp. 1254–64.

[Hig+09] T. Higo, Y. Matsushita, N. Joshi, and K. Ikeuchi. "A hand-held photometric stereo camera for 3-D modeling". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 1234–1241.

[HEH07] D. Hoiem, A. a. Efros, and M. Hebert. "Recovering surface layout from an image". In: *International Journal of Computer Vision, Springer* 75.1 (2007), pp. 151–172.

[Hou62] P. Hough. *Method and means for recognizing complex patterns*. U.S. Patent 3.069.654. 1962.

[IAH95] M. Irani, P. Anandan, and S. Hsu. "Mosaic based representations of video sequences and their applications". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV'95)*. Cambridge, MA, USA: IEEE Computer Society, 1995, pp. 605–611.

[Ish+11] T. Ishikawa, K. Thangamani, M. Kourogi, A. P. Gee, W. Mayol-Cuevas, J. Hyun, and T. Kurata. "Interactive 3-D indoor modeler for virtualizing service fields". In: *Virtual Reality, Springer* (2011), pp. 1–21.

[Iza+11] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon. "KinectFusion: real-time dynamic 3D surface reconstruction and interaction". In: *Proceedings of the ACM SIGGRAPH 2011 - Talks*. Vancouver, BC, Canada: ACM, 2011, p. 23.

[Kai+12] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg. "Omnikinect: real-time dense volumetric data acquisition and applications". In: *Proceedings of the 18th ACM symposium on Virtual reality software and technology (VRST '12)*. Toronto, Canada: ACM, 2012, pp. 25–32.

[KHF11] K. Karsch, V. Hedau, and D. Forsyth. "Rendering synthetic objects into legacy photographs". In: *ACM Transactions on Graphics (TOG)* 30.6 (2011), pp. 1–12.

[KS04] Y. Ke and R. Sukthankar. "PCA-SIFT: a more distinctive representation for local image descriptors". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*. Vol. 2. Lecture Notes in Computer Science 3. Washington, DC, USA: IEEE Computer Society, 2004, pp. 506–513.

[KM07] G. Klein and D. Murray. "Parallel tracking and mapping for small AR workspaces". In: *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*. Vol. 07. Nara, Japan: IEEE Computer Society, 2007, pp. 1–10.

[KCG11] A. Kowdle, Y. Chang, and A. Gallagher. "Active learning for piecewise planar 3D reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 24–26.

[KB11] D. Kurz and S. Benhimane. "Gravity-aware handheld augmented reality". In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 111–120.

[Lai+11] J. Lai, C. Chen, P. Wu, and C. Kao. "Tennis real play: an interactive tennis game with models from real videos". In: *Proceedings of the 19th ACM international conference on Multimedia (MM '11)*. Vol. 3. Scottsdale, AZ, USA: ACM, 2011, pp. 483–492.

[LHK09] D. C. Lee, M. Hebert, and T. Kanade. "Geometric reasoning for single image structure recovery". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. Miami, FL, USA: IEEE Computer Society, 2009, pp. 2136–2143.

[LGH10] D. Lee, A. Gupta, and M. Hebert. "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces". In: *NIPS, NIPS Foundation* 1 (2010), pp. 1–9.

[LC09] J. A. Lee and A. Y. S. Chia. "Robust matching of building facades under large viewpoint changes". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 1258–1264.

[Leo08] W. K. Leow. "Localization and mapping of surveillance cameras". In: *Proceedings of the 16th ACM international conference on Multimedia (MM '08)*. Vancouver, BC, Canada: ACM, 2008, pp. 369–378.

[LYG04] D. Levere, B. Yochelson, and P. Goldberg. *New York Changing: Revisiting Berenice Abbott's New York*. Princeton Architectural Press, 2004.

[Li+12] B. Li, K. Peng, X. Ying, and H. Zha. "Vanishing point detection using cascaded 1D Hough Transform from single images". In: *Pattern Recognition Letters, Elsevier* 33.1 (2012), pp. 1–8.

[LG10] B. Liu and S. Gould. "Single image depth estimation from predicted semantic labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. San Francisco, CA, USA: IEEE Computer Society, 2010.

[Liu+11] C. Liu, T. Huang, M. Chang, and K. Lee. "3D cinematography principles and their applications to stereoscopic media processing". In: *Proceedings of the 19th ACM international conference on Multimedia (MM '11)*. Scottsdale, AZ, USA: ACM, 2011, pp. 253–262.

[LHG08] F. Liu, Y. H. Hu, and M. L. Gleicher. "Discovering panoramas in web videos". In: *Proceedings of the 16th ACM international conference on Multimedia (MM'08)*. Vancouver, BC, Canada: ACM, 2008, pp. 329–338.

[Liu+08] H. Liu, S. Jiang, Q. Huang, and C. Xu. "A generic virtual content insertion system based on visual attention analysis". In: *Proceedings of the 16th ACM international conference on Multimedia (MM '08)*. Vancouver, BC, Canada: ACM, 2008, pp. 379–388.

[LBC10] D. Lovi, N. Birkbeck, and D. Cobzas. "Incremental free-space carving for real-time 3d reconstruction". In: *Proceedings of 5th International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'10)*. Paris, France, 2010.

[Low04] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision, Springer* 60.2 (2004), pp. 91–110.

[LK81] B. D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision". In: *International Joint Conferences on Artificial Intelligence (IJCAI'81)* 130 (1981), pp. 674–679.

[ML11] R. L. Mandryk and C. Lough. "The effects of intended use on target acquisition". In: *Proceedings of the 29th annual SIGCHI conference on Human factors in computing systems (CHI '11)*. Vancouver, BC, Canada: ACM, 2011, pp. 1649–1652.

[Mat+04] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions". In: *Image and Vision Computing* 22.10 (2004), pp. 761–767.

[Met13] Metaio. `http://www.metaio.com/`. (last access October 2013). 2013.

[MS05] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 27.10 (2005), pp. 1615–1630.

[Mil+94] P. Milgram, H. Takemura, A. Ustimi, and F. Kishino. "Augmented Reality: A class of display on the reality-virtuality continuum". In: *Telemanipulator and Telepresence Technologies* 2351 (1994), pp. 282–292.

[Min+01] F. Mintzer, G. W. Braudaway, F. P. Giordano, J. C. Lee, K. A. Magerlein, S. D'Auria, A. Ribak, G. Shapir, F. Schiattarella, J. Tolva, and A. Zelenkov. "Populating the Hermitage museum's new web site". In: *Communications of the ACM* 44.8 (2001), pp. 52–60.

[MK10] B. Mičušík and J. Košecká. "Multi-view superpixel stereo in urban environments". In: *International Journal of Computer Vision, Springer* 89.1 (2010), pp. 106–119.

[Mor80] H. P. Moravec. "Obstacle avoidance and navigation in the real world by a seeing robot rover". PhD thesis. Stanford, CA, USA, 1980.

[ML09] M. Muja and D. G. Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration". In: *Proceedings of the International Conference on Computer Vision Theory and Application (VISAPP'09)*. Lisboa, Portugal: INSTICC Press, 2009, pp. 331–340.

[MSS12a] A. Mulloni, H. Seichter, and D. Schmalstieg. "Indoor navigation with mixed reality world-in-miniature views and sparse localization on mobile devices". In: *Proceedings of the 2012 International Working Conference on Advaced Visual Interfaces (AVI'12)*. Capri Island, Naples, Italy: ACM, 2012, pp. 212–215.

[MSS12b] S. Myojin, A. Sato, and N. Shimada. "Augmented reality card game based on user-specific information control". In: *Proceedings of the 20th ACM international conference on Multimedia (MM'12)*. Nara, Japan: ACM, 2012, pp. 1193–1196.

[NBK12] T. Narumi, Y. Ban, and T. Kajinami. "Augmented perception of satiety: controlling food consumption by changing apparent size of food with augmented reality". In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. Austin, TX, USA: ACM, 2012, pp. 109–118.

[Ngu+12] V. Nguyen, M. Tran, T. Le, Q. Bui, and A. Duong. "Augmented media for traditional magazines". In: *Proceedings of the Third Symposium on Information and Communication Technology (SoICT '12)*. Da Nang, Vietnam: ACM, 2012, pp. 97–106.

[NCP97] L. Nlatthies, B. Chen, and J. Petrescu. "Stereo vision, residual image processing and Mars rover localization". In: *Proceedings of the International Conference on Image Processing (ICIP'97)*. Santa Barbara, CA, USA: IEEE Computer Society, 1997, pp. 248–251.

[Nob12] C. Nobre. "Sistema para Navegação Web usando Imagem e Vídeo". M. Sc. Lisboa, Portugal: Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2012.

[NC09] R. Nóbrega and S. Cavaco. "Detecting key features in popular music : case study – singing voice detection". In: *Second International Workshop on Machine Learning and Music at ECML-PKDD'09*. Bled, Slovenia, 2009, pp. 7–12.

[NC12a] R. Nóbrega and N. Correia. "Smart interface for reshaping photos in 3D". In: *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces (IUI'12)*. Lisboa, Portugal: ACM, 2012, pp. 3–4.

[NC11] R. Nóbrega and N. Correia. "Design your room: adding virtual objects to a real indoor scenario". In: *Extended Abstracts of the 2011 annual ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. Vancouver, BC, Canada: ACM, 2011, pp. 2143–2148.

[NC12b] R. Nóbrega and N. Correia. "Inserção dinâmica de objectos virtuais no contexto de fotografias tiradas por utilizadores". In: *Livro de Atas do 20º Encontro Português de Computação Gráfica (EPCG)*. Viana do Castelo, Portugal: EPCG, 2012, pp. 103–106.

[NC12c] R. Nóbrega and N. Correia. "Interactive insertion of virtual objects in photos and videos". In: *Proceedings of the Eurographics Conference 2012*. Cagliari, Sardinia, Italy: Eurographics Association, 2012, pp. 7–8.

[NC12d] R. Nóbrega and N. Correia. "Magnetic augmented reality: virtual objects in your space". In: *Proceedings of the 2012 International Working Conference on Advanced Visual Interfaces (AVI'12)*. Capri Island, Naples, Italy: ACM, 2012, pp. 332–335.

[NC12e] R. Nóbrega and N. Correia. *SEN : Shareable , editable and navigable spaces through image reconstruction and inter-connection*. Tech. rep. Lisboa, Portugal: CITI,FCT,UNL, 2012.

[NC13a] R. Nóbrega and N. Correia. "Dynamic Insertion of virtual objects in photographs". In: *International Journal on Creative Interfaces and Computer Graphics (IJCICG)* 1 (2013), to be published.

[NC13b] R. Nóbrega and N. Correia. "Photo-based multimedia applications using image features detection". In: *Proceedings of International Conference on Computer Graphics Theory and Applications (GRAPP'13)*. Barcelona, Spain: INSTICC Press, 2013, pp. 298–307.

[Nó09] R. Nóbrega. "Building interactive spatial and temporal models using multimodal data". Thesis Plan. Lisbon, Portugal: Faculdade Ciências e Tecnologia, Universidade Nova de Lisboa, 2009.

[Nó11] R. Nóbrega. "Modeling places for interactive media and entertainment applications". In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI '11)*. Vancouver, BC, Canada: ACM, 2011, pp. 1081–1084.

[Nó+12] R. Nóbrega, N. Correia, C. Nobre, A. B. Teixeira, L. Oliveira, and R. H. da Silva. "Navigation in past museum exhibitions using multimedia archives". In: *Proceedings of the 2012 Working conferecence on Advanced Visual Interfaces (AVI'12)*. Capri Island, Naples, Italy: ACM, 2012, pp. 778–779.

[OF12] O. Oda and S. Feiner. "3D referencing techniques for physical objects in shared augmented reality". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 207–215.

[OS05] J. Oh and W. Stuerzlinger. "Moving objects with 2D input devices in CAD systems and Desktop Virtual Environments". In: *Proceedings of Graphics Interface (GI'05)*. Victoria, BC, Canada: ACM, 2005, pp. 195–202.

[Ope13] OpenCV. `http://opencv.org`. (last access October 2013). 2013.

[ope13] openFrameworks. `http://www.openframeworks.cc`. (last access October 2013). 2013.

[PAR11] Q. Pan, C. Arth, and E. Rosten. "Rapid scene reconstruction on mobile phones from panoramic images". In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 55–64.

[PH09] G. Peters and K. Häming. "Take three snapshots - a tool for fast freehand acquisition of 3D objects". In: *INTERACT '09 Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction*. Uppsala, Sweden: Springer-Verlag, 2009, pp. 842–843.

[Pol+07] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. "Detailed real-time urban 3D reconstruction from video". In: *International Journal of Computer Vision, Springer* 78 (2007), pp. 143–167.

[PKG98] M. Pollefeys, R. Koch, and L. V. Gool. "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*. Santa Barbara, CA, USA: IEEE Computer Society, 1998, pp. 90–95.

[PKV98] M. Pollefeys, R. Koch, and M. Vergauwen. "Automatic generation of 3d models from Photographs". In: *Proceedings of Virtual Systems and MultiMedia (VSMM'98)*. Gifu, Japan, 1998.

[PKVP09] Y. Pritch, E. Kav-Venaki, and S. Peleg. "Shift-map image editing". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 151 –158.

[RAPP06] A. Rav-Acha, Y. Pritch, and S. Peleg. "Making a long video short: dynamic video synopsis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 1. New York City, NY, USA: IEEE Computer Society, 2006, pp. 435–441.

[RA+08] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. "Unwrap mosaics: a new representation for video editing". In: *ACM Transactions on Graphics (TOG)* 27.3 (2008).

[Ra+04] T. Romão, N. Correia, E. Dias, J. Danado, A. Trabuco, C. Santos, R. Santos, A. Câmara, and E. Nobre. "ANTS—Augmented environments". In: *Computers & Graphics, Elsevier* 28.5 (2004), pp. 625–633.

[RD05] E. Rosten and T. Drummond. "Fusing points and lines for high performance tracking". In: *Proceedings of the IEEE 5th International Conference on Computer Vision (ICCV'05)*. Beijing, China: IEEE Computer Society, 2005, pp. 1508–1515.

[Rot02] C. Rother. "A new approach to vanishing point detection in architectural environments". In: *Image and Vision Computing, Elsevier* 20.1 (2002), pp. 647–655.

[RK04] C. Rother and V Kolmogorov. "GrabCut — Interactive foreground extraction using iterated Graph Cuts". In: *ACM Transactions on Graphics (TOG)* (2004).

[Rou+12] A. Roussos, C. Russell, R. Garg, and L. Agapito. "Dense multibody motion estimation and reconstruction from a handheld camera". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 31–40.

[Rub+11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *Proceedings of the International Conference on Computer Vision (ICCV'11)*. Barcelona, Spain: IEEE Computer Society, 2011, pp. 2564–2571.

[San+09] J. Santolaria, D. Guillomía, C. Cajal, J. a. Albajez, and J. J. Aguilar. "Modelling and calibration technique of laser triangulation sensors for integration in robot arms and articulated arm coordinate measuring machines". In: *Sensors* 9.9 (2009), pp. 7374–96.

[SSN09] A. Saxena, M. Sun, and A. Y. Ng. "Make3D: learning 3D scene structure from a single still image." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 31.5 (2009), pp. 824–40.

[She04] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. 3rd Editio. Chapman & Hall/CRC, 2004.

[ST94] J. Shi and C. Tomasi. "Good features to track". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. Vol. 94. Seattle, WA, USA: IEEE Computer Society, 1994, pp. 593–600.

[Shi+12] M. Shirose, M. Hirose, K. Oku, M. Koide, and N. Hirai. "Passage+". In: *Proceedings of the 20th ACM international conference on Multimedia (MM '12)*. Nara, Japan: ACM, 2012, p. 1497.

[Sho+11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. "Real-time human pose recognition in parts from single depth images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, USA: IEEE Computer Society, 2011, pp. 1297–1304.

[Sim06] G. Simon. "Automatic online walls detection for immediate use in AR tasks". In: *Proceedings IEEE International Symposium On Mixed and Augmented Reality (ISMAR'06)*. Santa Barbara, CA, USA: IEEE Computer Society, 2006, pp. 4–7.

[SB02] G. Simon and M.-O. Berger. "Pose estimation for planar structures". In: *Computer Graphics and Applications, IEEE* 22.6 (2002), pp. 46–53.

[SFZ00] G. Simon, A. W. Fitzgibbon, and A. Zisserman. "Markerless tracking using planar structures in the scene". In: *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR'00)*. Vol. 9. Munich, Germany: IEEE Computer Society, 2000, pp. 120–128.

[SSS09] S. N. Sinha, D. Steedly, and R. Szeliski. "Piecewise planar stereo for image-based rendering". In: *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society, 2009, pp. 1881–1888.

[SSS06] N. Snavely, S. Seitz, and R. Szeliski. "Photo tourism: exploring photo collections in 3D". In: *ACM Transactions on Graphics (TOG)* 80.2 (2006), pp. 189–210.

[SSS08] N. Snavely, S. Seitz, and R. Szeliski. "Modeling the world from internet photo collections". In: *International Journal of Computer Vision, Springer* 80.2 (2008), pp. 189–210.

[SFB08] H. Stelmaszewska, B. Fields, and A. Blandford. "The roles of time, place, value and relationships in collocated photo sharing with camera phones". In: *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*. BCS-HCI '08. Swinton, UK, UK: British Computer Society, 2008, pp. 141–150.

[Stu11] StudiertubeTracker. `http://handheldar.icg.tugraz.at/stbtracker.php`. (last access October 2013). 2011.

[Sty07] G. D. Styliaras. "A web-based presentation framework for museums". In: *Proceedings of the 2007 Euro American conference on Telematics and information systems (EATIS '07)*. Faro, Portugal: ACM, 2007, p. 13.

[SFE12] M. Sukan, S. Feiner, and S. Energin. "Manipulating virtual objects in hand-held augmented reality using stored snapshots". In: *2012 IEEE Symposium on 3D User Interfaces (3DUI'12)*. Costa Mesa, CA, USA: IEEE Computer Society, 2012, pp. 165–166.

[Sze96] R. Szeliski. "Video mosaics for virtual environments". In: *Computer Graphics and Applications, IEEE* (1996), pp. 22–30.

[Sze10] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[TP12] Y. Takeuchi and K. Perlin. "ClayVision: The (elastic) image of the city". In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. Austin, TX, USA: ACM, 2012, pp. 2411–2420.

[TRD09] S. Taylor, E Rosten, and T Drummond. "Robust feature matching in 2.3 microseconds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. Miami, FL, USA: IEEE Computer Society, 2009, pp. 15–22.

[TM11] A. B. Tillon and I. Marchal. "Mobile augmented reality in the museum: Can a lace-like technology take you closer to works of art?" In: *Proceedings of the IEEE International Symposium On Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 41–47.

[TSS11] M. Turcsanyi-Szabo and P. Simon. "Augmenting experiences bridge between two universities". In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 7–13.

[TM08] T. Tuytelaars and K. Mikolajczyk. "Local invariant feature detectors: a survey". In: *Foundations and Trends in Computer Graphics and Vision* 3.3 (2008), pp. 177–280.

[Uch11] H. Uchiyama. "Toward augmenting everything: Detecting and tracking geometrical features on planar objects". In: *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'11)*. Basel, Switzerland: IEEE Computer Society, 2011, pp. 17–25.

[UTM12] H. Uchiyama, V. Teichrieb, and E. Marchand. "Texture-less planar object detection and pose estimation using depth-assisted rectification of contours". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 297–298.

[Val98] J. Vallino. *Interactive augmented reality*. 1998.

[VJ01] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*. Vol. 1. C. Kauai, HI, USA: IEEE Computer Society, 2001, pp. 511–518.

[Vuf13] Vuforia. `https://www.vuforia.com/`. (last access October 2013). 2013.

[WSB09] D. Wagner, D. Schmalstieg, and H. Bischof. "Multiple target detection and tracking with guaranteed framerates on mobile phones". In: *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'09)*. Orlando, FL, USA: IEEE Computer Society, 2009, pp. 57–64.

[Wag+10a] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. "Real-time detection and tracking for augmented reality on mobile phones". In: *IEEE Transactions on Visualization and Computer Graphics* 16.3 (2010), pp. 355–368.

[Wag+10b] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg. "Real-time panoramic mapping and tracking on mobile phones". In: *Proceedings of the 2010 IEEE Virtual Reality Conference (VR'10)*. Waltham, MA, USA: IEEE Computer Society, 2010, pp. 211–218.

[WG10] H. Wang and S. Gould. "Discriminative learning with latent variables for cluttered indoor scene understanding". In: *Proceedings of the 11th European Conference on Computer Vision (ECCV'10)*. Hersonissos, Crete, Greece: Springer-Verlag, 2010, pp. 497–510.

[WZ02] T. Werner and A. Zisserman. "New techniques for automated architectural reconstruction from photographs". In: *Proceedings of the 7th European Conference on Computer Vision (ECCV'02)*. Copenhagen, Denmark: Springer-Verlag, 2002, pp. 541–555.

[Xio+11] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert. "3-D scene analysis via sequenced predictions over points and regions". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'11)*. Shanghai, China: IEEE Computer Society, 2011, pp. 2609–2616.

[YC12] X. Yang and K. Cheng. "LDB: An ultra-fast feature for scalable Augmented Reality on mobile devices". In: *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12)*. Vol. 2. Atlanta, GA, USA: IEEE Computer Society, 2012, pp. 49–57.

[YHC12] C. You, Y. Hsieh, and W. Cheng. "AttachedShock: facilitating moving targets acquisition on augmented reality devices using goal-cross ing actions". In: *Proceedings of the 20th ACM international conference on Multimedia (MM'12)*. Nara, Japan: ACM, 2012, pp. 1141–1144.

[YM08] S. X. Yu and J. Malik. "Inferring spatial layout from a single image via depth-ordered grouping". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08) - Workshops*. Anchorage, AK, USA: IEEE Computer Society, 2008, pp. 1–7.

[Zic+02] T. E. Zickler, E. Engineering, N. Haven, P. N. Belhumeur, and D. J. Kriegman. "Helmholtz stereopsis : Exploiting reciprocity for surface reconstruction". In: *International Journal of Computer Vision, Springer* 49 (2002), pp. 215–227.

List of Websites

The last access for every website is October, 2013.

A.1 Libraries and Frameworks

ARToolKit, AR library based on markers,
`http://www.hitl.washington.edu/artoolkit/`.

FAST, Features from Accelerated Segment Test,
`http://www.edwardrosten.com/work/fast.html`.

FastCV, Mobile-optimized computer vision library,
`https://developer.qualcomm.com/mobile-development/mobile-technologies/computer-vision-fastcv`.

Metaio, AR platform with library based on image markers,
`http://www.metaio.com/`.

Pointcloud, 3D SLAM library for smartphones from 13thLab,
`http://pointcloud.io`.

Processing JS, Javascript version of the graphic framework Processing,
http://processingjs.org/.

Rich Faces, Java Ajax library,
http://www.jboss.org/richfaces/.

OpenCV, Open Computer Vision library,
http://opencv.org.

openFrameworks, open source C++ graphic toolkit,
http://www.openframeworks.cc/.

StudiertubeTracker, AR library based on markers,
http://handheldar.icg.tugraz.at/stbtracker.php.

Vuforia, AR library from Qualcomm,
https://www.vuforia.com/.

A.2 Products

Atelier Pfister, smartphone application for furniture design,
http://www.atelierpfister.ch/app.

Ball Invasion, Smartphone game based on PointCloud and PTAM,
http://13thlab.com/ballinvasion/.

Eye-Fi, Wireless SD card,
http://www.eye.fi/.

Eye Pet, Sony Playstation camera game,
http://www.eyepet.com/.

DesignMyRoom, Interior design application,
http://designmyroom.com.

Google Art Project, Street View exploration of museums,
http://www.google.com/culturalinstitute/project/art-project.

Google Glass, Interactive glasses that can be used for AR,
http://www.google.com/glass/.

Ikea 2014 catalogue,
http://www.ikea.com/ms/en_AA/customer_service/catalogue/catalogue_2014.html.

Junaio, GPS augmented reality application,
http://www.junaio.com/.

Kinect,Microsoft Kinect camera,
http://www.microsoft.com/en-us/kinectforwindows/.

Krpano, panoramic Flash plug-in,
http://krpano.com/.

Layar, GPS augmented reality application,
http://www.layar.com/.

PhotoSynth, 3D reconstruction from several photos,
http://photosynth.net.

A.3 Websites

1957 Gulbenkian's modern art exhibition, Implemented website,
http://img.di.fct.unl.pt/expo1957/,
http://expo1957.fcsh.unl.pt/.

Calouste Gulbenkian Foundation Library, Calouste Gulbenkian Foundation Library,
http://www.biblarte.gulbenkian.pt/.

CITI, Center for Informatics and Information Technologies , Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa,
http://citi.di.fct.unl.pt/.

Flickr, Photo sharing website,
http://www.flickr.com/.

ICG, Institute of Computer Graphics and Vision at TUGraz,
http://www.icg.tu-graz.ac.at/.

ICVSS'11, International Computer Vision Summer School 2011,
http://svg.dmi.unict.it/icvss2011/.

MNAA, Museu Nacional de Arte Antiga,
http://www.museudearteantiga.pt/.

Rui Nóbrega's, Ph.D. blog,
http://doutorandoemfilosofia.blogspot.com/.

TUGraz, Technical University of Graz,
http://www.tugraz.at/.

UMa, Universidade da Madeira,
http://www.uma.pt/.

WARM'11, Winter Augmented Reality Meeting,
http://studierstube.icg.tugraz.at/WARM13/.

YorkUrban DB, York Urban Line Segment Database,
http://www.elderlab.yorku.ca/YorkUrbanDB/.

B

List of Videos

List of videos from the applications presented in Chapter 5. Most videos can be seen directly on modern browsers which read the .mp4 format. Additional suggestions include the VLC video player[1]. The QR codes were created using goQR[2] and can be used to launch the videos from tablets or smartphones. Possible reader applications include QR Droid for Android or QR Reader for iOS.

Magnetic Augmented Objects in Photos

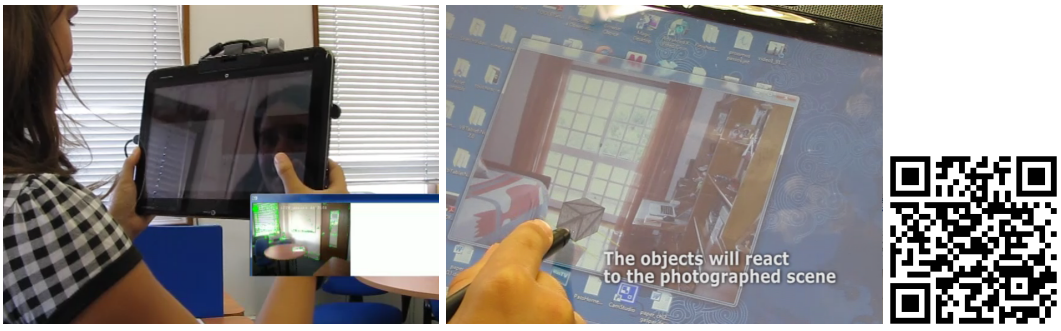


Figure B.1: Preview of the magnetic augmented objects in photos application. (Video: `http://img.di.fct.unl.pt/rpn/phdthesis/magnetic.mp4`)

[1]VLC, video player, `http://www.videolan.org/vlc/`.
[2]goQR, online QR-code creator, `http://goqr.me/`.

Mixed Reality Snake

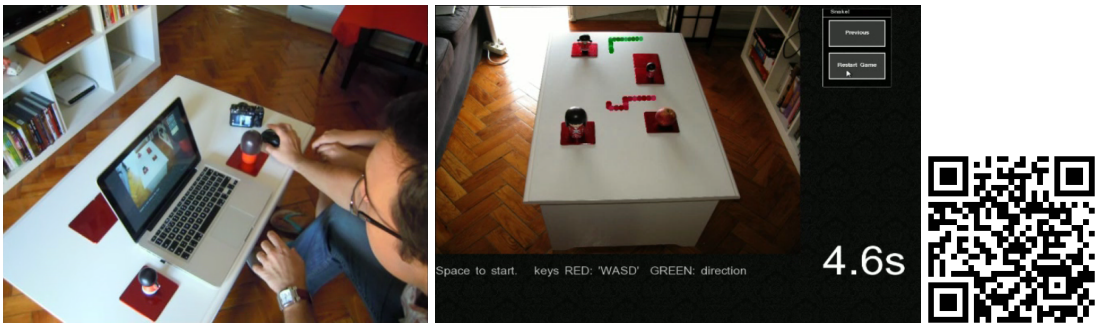


Figure B.2: Preview of the mixed reality snake application.
(Video: `http://img.di.fct.unl.pt/rpn/phdthesis/snake.mp4`)

Past Museum Exhibition Navigation

Figure B.3: Preview of the museum exploration.
(Video: `http://img.di.fct.unl.pt/rpn/phdthesis/gulbenkian.mp4`)

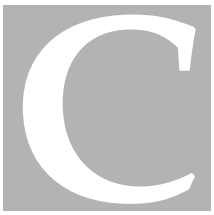


Image Datasets

List of used image datasets:

General: Mixed collection of photos used to test the algorithm, which includes indoor and outdoor photos, old, blurred and pixelated images. Images in Figure C.1.

Lab: Photos from the interior of a building and from the FCT/UNL faculty campus. Images in Figure C.2.

Flickr-Mix: Photos obtained from Flickr[1] with keywords such as "House", "House Interior" and "Buildings". It includes some landscape images with nature. Images which did not constitute a picture of a physical space where removed (e.g., animals or people). Images in Figure C.3.

YorkUrbanDB: The York Urban Line Segment Database[2] is an online database used by Coughlan and Yuille [CY99], which contains images of urban environments consisting mostly of scenes from the campus of York University and downtown Toronto, Canada. Most images follow the rule of the Manhattan world. This is the database is used in subsection 4.4.1, as ground truth because of its annotated vanishing points. Images in Figure C.4.

[1] Flickr, Photo sharing website, `http://www.flickr.com`

[2] York Urban Line Segment Database, `http://www.elderlab.yorku.ca/YorkUrbanDB/`.

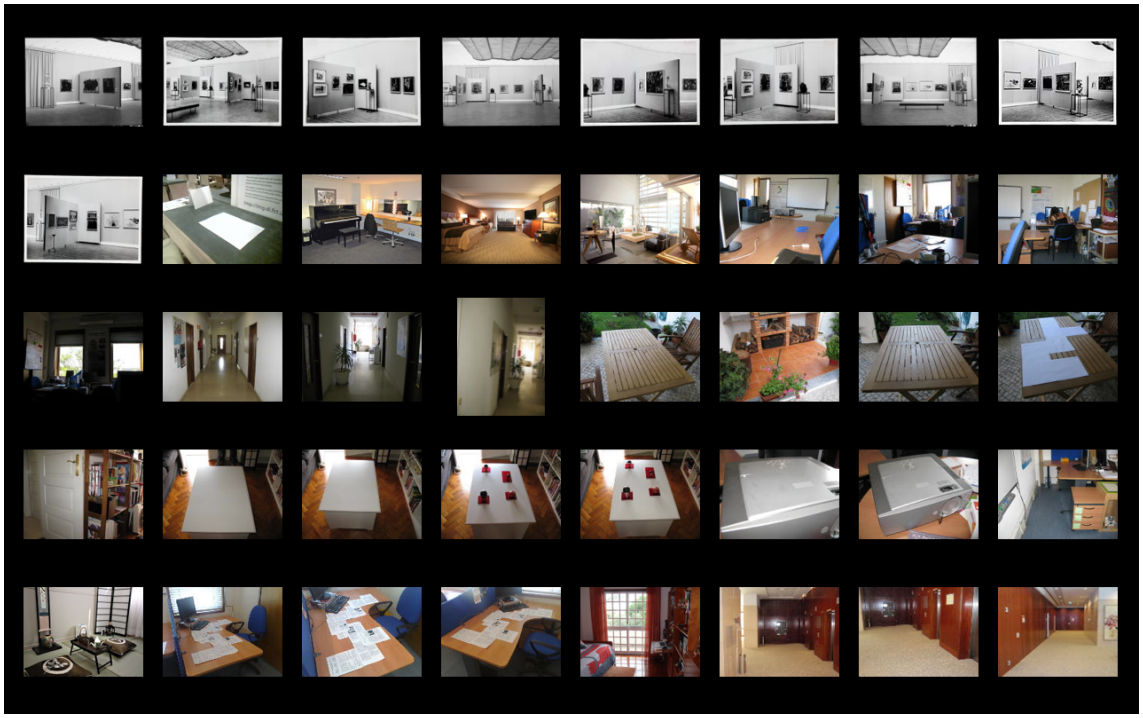


Figure C.1: Images of the the CITI lab, old photos, hotel lobby, Caparica beach and various places.

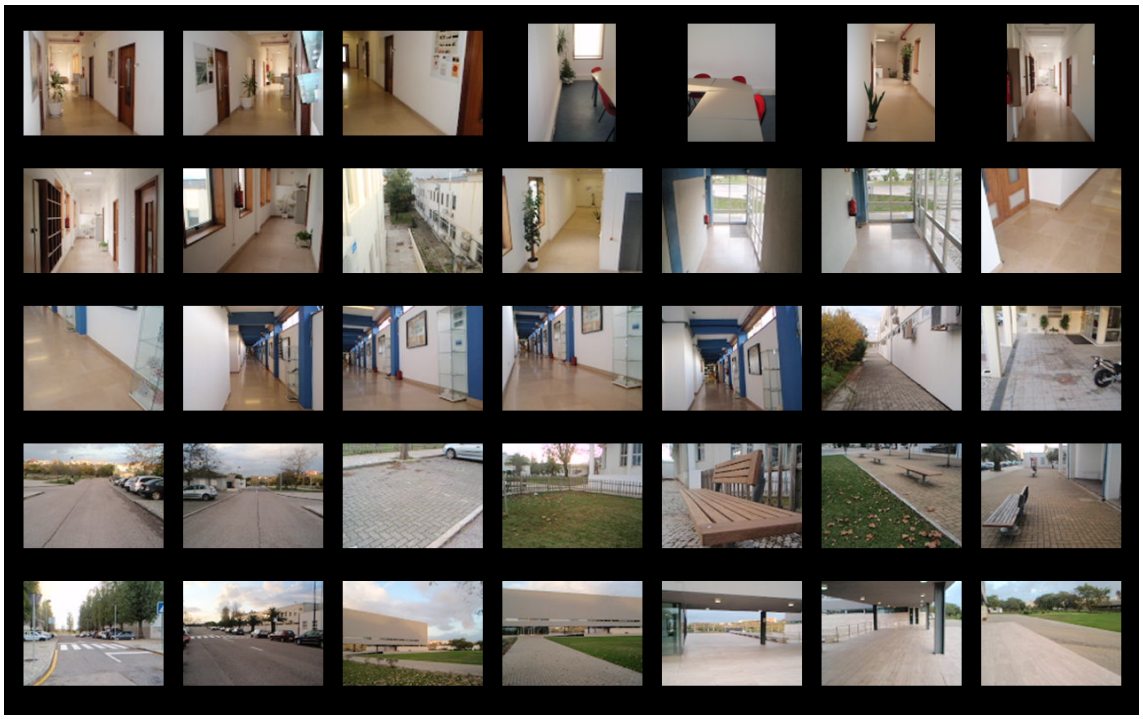


Figure C.2: Images of the the CITI lab and FCT/UNL faculty campus.

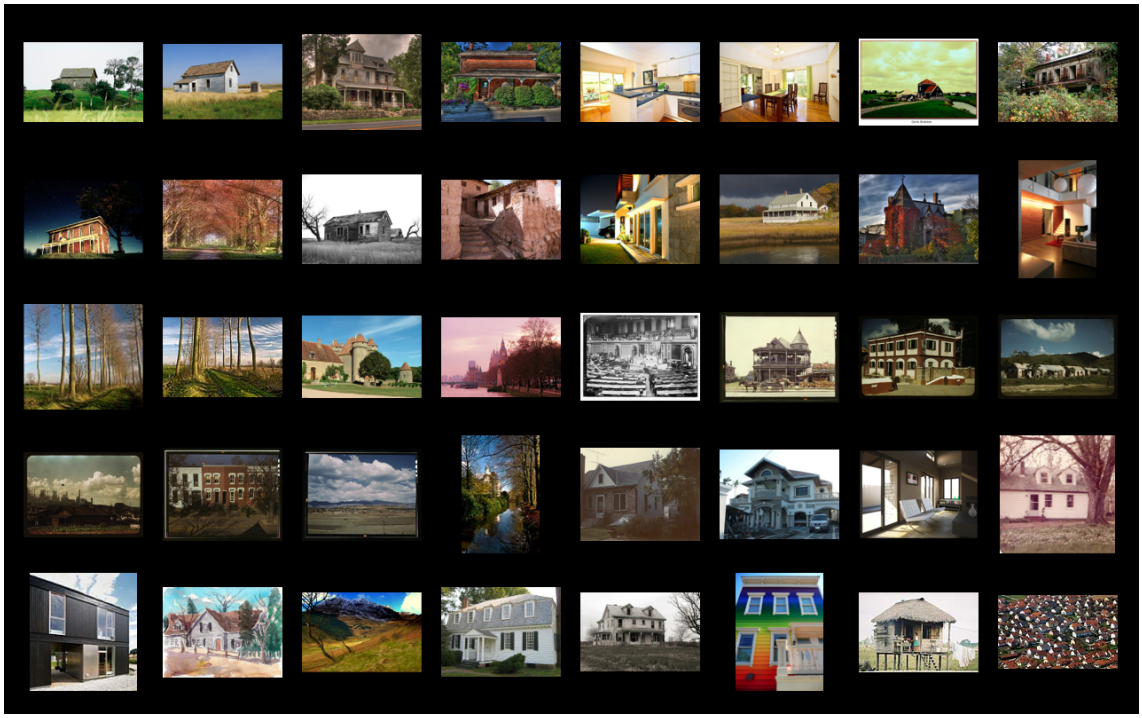


Figure C.3: Random images from Flickr retrieved with tags such as "house", "indoor" or architecture".

Figure C.4: Images from the YorkUrban database.