Photometric registration of indoor real scenes using an RGB-D camera with application to mixed reality

Salma Jiddi

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

Salma JIDDI

Photometric Registration of Indoor Real Scenes using an RGB-D Camera with Application to Mixed Reality

Thèse présentée et soutenue à Rennes, le 11/01/2019
Unité de recherche : IRISA – UMR6074
Thèse N° :

Rapporteurs avant soutenance :

Vincent Lepetit	Professeur à l'Université de Bordeaux
Alain Trémeau	Professeur à l'Université Jean Monnet

Composition du Jury :

Rapporteurs :	Vincent Lepetit	Professeur à l'Université de Bordeaux
	Alain Trémeau	Professeur à l'Université Jean Monnet
Examinateurs :	Kadi Bouatouch	Professeur à l'Université de Rennes 1
	Michèle Gouiffès	Maître de Conférences à l'Université Paris Sud
Co-encadrant :	Philippe Robert	Docteur Ingénieur à Technicolor
Dir. de thèse :	Eric Marchand	Professeur à l'Université de Rennes 1

Acknowledgements

This thesis is the culmination of three years of hard work, enriching encounters and fulfilling accomplishments. Although only my name appears on the dissertation cover, a great deal of people were involved in its shaping and realization.

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Eric Marchand and Dr. Philippe Robert, who offered their continuous advice and involvement on a daily basis, from the start of this journey up until its very end. I thank them for the methodical guidance and tremendous effort they put into training me in this stimulating research field. They contributed to a rewarding PhD degree by providing me with intellectual freedom in my work, engaging me in new ideas, and demanding high standards in all of my endeavors. I am forever indebted. Thank you for the latter and so much more.

I would like to thank my dissertation committee members, namely Prof. Vincent Lepetit, Prof. Alain Trémeau, Prof. Kadi Bouatouch and Dr. Michèle Gouiffès, for their invaluable advice and crucial remarks that shaped my final thesis. Thank you.

In full gratitude, I would like to acknowledge my colleagues from Technicolor's Mixed Reality team, namely Pierrick Jouet, Vincent Alleaume, Anthony Laurent, Matthieu Fradet, Fabien Servant, Grégoire Nieto, Tao Luo and Mohammad Rouhani. Their constant help and much pleasant working environment have made my three-year doctoral experience more than perfect. I certainly am going to miss our gripping over-coffee discussions, unequaled team spirit and, most importantly, agreed-on habit of getting those unhealthy yet delicious snacks. Thank you for being such a dream team.

I acknowledge my gratitude to Caroline Baillard, the leader of the Mixed Reality Technical Area, for her heartfelt welcome and encouragements. I thank her for giving me the opportunity to be part of several interesting projects and making my first business trip such a wonderful experience. Thank you.

My profound gratitude also goes to Pierre Houeix, the Media Computing laboratory manager, who offered me the opportunity to join his highly competent team and provided his unconditional support throughout my entire thesis. Thank you for the latter and so much more.

My time at Technicolor was made beyond enjoyable and gratifying in large part thanks to the highly skilled and kind people whom I was honored to meet. The list is far from being exhaustive but I would like to acknowledge Gaël, Philippe, Jean, Thomas, Dmitry,

Rémy, Tristan, Matthieu, Paul, Sandra, Izabela, Julien, Nicolas, François and Fabien. I am thankful for all the support they brought during my thesis. Their passion for their work, their incessant help and much valuable advice were a true pillar.

Equally, I would like to express my gratitude to Technicolor's security team, namely Xavier, Madi and Robert. Their unfailing assistance and support have helped me throughout my tenure, particularly near paper submission deadlines when late night working hours were just part of the deal. A special thank you goes to Harouna for his continuous encouragement and cheerful spirit. Today, when I hear "Copacabana" or "cocotiers", my mind is naturally wired to evoke his name. Thank you.

My earnest thanks go to the Rainbow team at IRISA for their warm welcome and enlightening discussions. I thank both team leaders, Prof. François Chaumette and Prof. Paolo Robuffo Giordano for giving me the opportunity to be part of their outstanding and talented team. I further acknowledge my labmates namely Brian, Firas, Rahaf, Fabien, Julien, Vincent and Helene, for their support, collaboration as well as the gratifying SAV and climbing experiences. Thank you.

A heartfelt thank you goes to a handful of important people in my life, my friends: Hadrien (petit poney), Anas (l'ananas), Fatma (fatouHR) and Jérémie (la tata). None of this would have been possible without your unconditional love and ceaseless support. Your valuable presence throughout the good, and bad, helped me keep my goals in perspective. I greatly value your contribution and deeply appreciate your belief in me. Thank you so much.

Last but by no means least, I acknowledge the people who mean everything to me, my family. To my wonderful sisters Imane, Yasmine, Kouchia, my meow-brother Arès and my loving parents. There are no words to describe how grateful I am. Perhaps, the emotion we shared on the eleventh of January is its most faithful representation. Thank you for all the selfless love, care, understanding and sacrifices you've willingly granted me all these years. Thank you for your dedicated efforts which contributed to this beautiful achievement. In my turn, I humbly dedicate this thesis to you.

To you.
1997 - 2009

Contents

Notations

Unless otherwise stated, the main conventions used in the notation of this Thesis are described in the following.

General

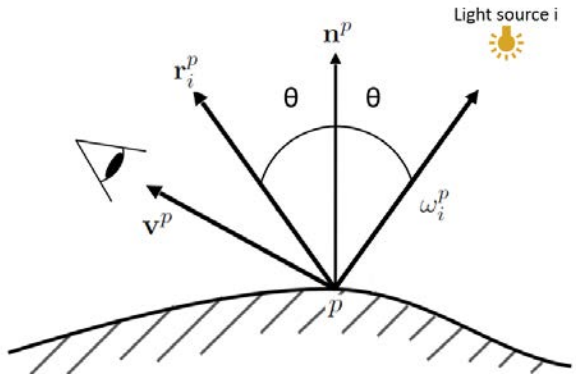
- $\mathcal{F}$: Cartesian frame.
- a: scalar.
- $\mathbf{a}$: column vector.
- $\mathbf{a}^\top$: row vector and the transpose of column vector $\mathbf{a}$.
- $\mathbf{A}$: matrix.
- $\mathbf{A}^{-1}$: inverse of matrix $\mathbf{A}$.
- $\mathbf{A}^\top$: transpose of matrix $\mathbf{A}$.
- $\mathbf{a}.\mathbf{b} = \mathbf{ab}$: scalar product of column vectors $\mathbf{a}$ and $\mathbf{b}$.
- $\mathbf{a} \times \mathbf{b}$: cross product of column vectors $\mathbf{a}$ and $\mathbf{b}$.

Euclidean Geometry

- $\bar{\mathbf{X}} = (X, Y, Z)^\top$: coordinates of a point in the Euclidean space.
- $\mathbf{X} = (X, Y, Z, 1)^\top$: homogeneous coordinates of a point in the Euclidean space.
- ${}^i\mathbf{X}$: coordinates $\mathbf{X}$ expressed in frame $\mathcal{F}_i$.
- $\bar{\mathbf{x}} = (x, y)^\top$: image point coordinates in pixels.
- $\mathbf{x} = (x, y, 1)^\top$: image point coordinates in homogeneous pixels formulation.
- ${}^i\mathbf{T}_j$: homogeneous transformation matrix from $\mathcal{F}_j$ to $\mathcal{F}_i$.
- ${}^i\mathbf{R}_j$: rotation matrix from frame $\mathcal{F}_j$ to frame $\mathcal{F}_i$.
- ${}^i\mathbf{t}_j$: translation vector from frame $\mathcal{F}_j$ to frame $\mathcal{F}_i$.

Reflection Model

The color of each pixel is represented by a normalized RGB column vector and rescaled into the range from 0 to 255 for visualization.



- p: point in the scene.
- ω_i^p: light direction vector with respect to light source i.
- $\mathbf{n}^p$: normal vector of point p.
- $\mathbf{r}_i^p$: perfect reflection vector with respect to light source i.
- $\mathbf{v}^p$: viewpoint vector.

Introduction

1

Since the late 1960s, the creation of realistic computer-generated images (CGI) has been a major focus of the computer graphics research. Achieved work has led to various algorithms which deliver outsandingly realistic images of *fully* modeled virtual worlds. The possibility of creating *any* virtual world has contributed, 30 years later, to the emergence of the first Mixed Reality (MR) system [Rosenberg, 1993]. In 1992, Louis Rosenberg introduced the concept of virtual fixtures as an overlay of virtual information on a given workspace in the context of military tasks. The core idea of superimposing or mixing real and virtual information was first coined by Tom Caudell [Caudell and Mizell, 1992] and later referred to as Mixed Reality (MR) within the Reality-Virtuality (RV) continuum framework [Milgram and Kishino, 1994] .

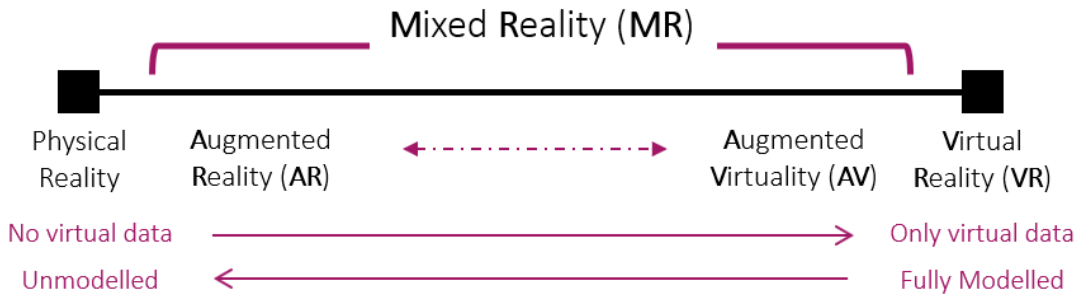


Figure 1.1 – The Reality-Virtuality Continuum goes from the Physcical Reality (real environment) which does not contain any virtual data to the Virtual Reality world where everything is virtual and fully modeled.

The RV continuum (Figure 1.1), as presented by Milgram and Kishino in [Milgram and Kishino, 1994], goes from the physical real world where no virtual data exists to a Virtual Reality (VR) world where everything is virtual and modeled. Within this continuum, MR was defined as "*...anywhere between the extrema of the RV continuum*". Hence, it comprises all the configurations which span the continuum from Augmented Reality (AR) where the virtual augments the real to Augmented Virtuality (AV), where the real augments the virtual.

In this Thesis, we are interested in MR systems which are positioned along the segment between AR and AV technologies in the RV continuum. These systems are known to achieve a *seamless* visual blending between physical and digital worlds. Specifically, computer-generated data in the context of this thesis are never considered to be 2D figures, text or basic geometric primitives (e.g., lines, circles, etc.). In our case, the virtual world can be composed of one or multiple 3D objects. In addition to the virtual world's content, such MR systems must consistently align, in 3D and in real time, both

real and virtual worlds [Azuma, 1997] [Schmalstieg and Hollerer, 2016].

Experiencing MR requires the use of a display device through which the blending of the physical world with the digital world can be visualized. These devices have been mainly categorized into head mounted (HMD) and non-head mounted displays [Azuma, 1997][Fuchs and Ackerman, 1999][Costanza et al., 2009]. The former were introduced by Ivan Sutherland [Sutherland, 1968] in the late 1960s and have been the mainstay of a significant number of MR systems to date. They owe their popularity to the amount of freedom they provide while experiencing MR applications. Efforts in this research area have led to various high-end wearables that offer outstanding MR experiences thanks to their high resolution, low latency and sensory components [Microsoft, 2016][MetaVision, 2017][Leap, 2018]. On the other hand, non-Head Mounted Displays such as phones and tablets have become commonly used in MR applications thanks to their pricing point and increasing computational power capabilities.

As the involved technology in MR matures, such systems move towards being affordable and user-friendly products. Consequently, the number of scenarios and contexts in which MR can be useful is *limitless* [Costanza et al., 2009][Mekni and Lemieux, 2014]. Perhaps, the first objective behind the information conveyed by the virtual objects is to make a task easier for a human to perform [Brooks, 1996]. Accordingly, the first field to welcome this technology was the military one through the work of Rosenberg [Rosenberg, 1993] where virtual data was overlaid to enhance the user's telerobotic experience. Since then, MR systems have been used to help operators better locate points of interest (e.g., streets, airports, railroads) and improve their situational awareness [Calhoun et al., 2005]. In the medical field, Mixed reality can be a useful tool for doctors and surgeons [Botella et al., 2010]. For instance, it can provide them with crucial overlayed information regarding their patient (e.g., records, imaging tumors) [Thomas, 2016]. Also, it has been massively considered as an efficient tool for certain phobia treatments and cooperative surgical scenarios (Figure 1.2).

Figure 1.2 – Examples of mixed reality applications: (left) a cooperative surgical scenario, (middle) a capture of Chemistry AR application and (right) the most successful gaming application (*PokémonGO*).

Mixed reality is without any doubt going to play a major role in shaping our realities in the near future, not only because of its various use cases but also because of the time-saving, productivity and economic growth that it brings. In [Tang et al., 2003], authors found that when integrating MR in the manufacturing field, the error rate for an assembly task was reduced by 82% compared to using a printed manual. Enhancing the user's surrounding with an adequate and adaptive amount of information helps better understand the assembly task. In [Herpich et al., 2017], a comparative analysis

demonstrates how MR frameworks can help improve traditional classroom techniques by offering more individualized and flexible learning. To illustrate, Chemistry AR (Figure 1.2) allows student to visualize and interact with virtual molecules. All scenarios being considered, the most impacted field is the entertainment one. According to a recent Goldman Sachs report [GoldmanSachs, 2016], the MR consumer market in entertainment was the first to develop and is believed to grow to 216 million users by 2025. By offering outstanding virtual contents, MR has impacted most of the entertainment industries such as gaming, cinema and live events.

1.1 Thesis Context

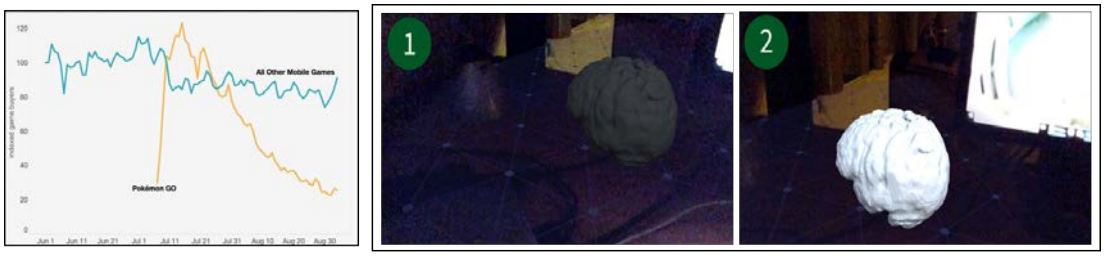
This work belongs to the entertainment field. It has been conducted within the frame of a CIFRE[1] industrial partnership with Technicolor, a multinational company that delivers services and products to entertainment industries, and the IRISA laboratory. This thesis was carried out in the MR technical area which focuses on providing innovative solutions with an emphasis on interactive, real-time and *realistic* content.

Although several industrial actors have already proposed real-time MR solutions (ARKit, Vuforia, Wikitude) able to geometrically align the real world with the digital world, none of them tackled the problem of achieving a seamless and *realistic* blending. Indeed, when using such systems, the virtual objects are often easily distinguishable from the real ones as their appearances do not match. Since immersion is an important aspect of MR systems [Roussou and Drettakis, 2003], this is clearly a problem. In fact, human visual cues are sensitive to the global coherence within an image. Hence, absence or incorrectly rendered virtual shadows, confused color perception such as an exuberantly bright virtual object are all elements which may not help an MR user interact and commit to a target application.

To illustrate, within the large panel of existing gaming applications, *PokémonGO* has remained the first one to bring MR widely to the mainstream since 2016. As shown in figure 1.3-a, this application has accomplished in a very short period of time what all previously proposed applications did not: over 750 million global downloads. Nevertheless, its success has drastically fallen down after a short period following its launch. According to [Polygon, 2018][Gamerant, 2018], it is mainly because it lacked essential immersion features such as *real-world occlusions* and *realism*.

It is only recently that Google presented the ARCore Software Development Kit (SDK) as an example of what realistic MR *could* be. In their demonstration (Figure 1.3-b), one can see that the virtual object's appearance approximately fits the global lighting in the scene. The overall rendering of the virtual object is adapted by estimating an average brightness over the current image of the real world. Nonetheless, this is clearly not sufficient since primary cues to achieve realistic MR such as shadows, are still missing. These cues not only improve the realism of the virtual object itself but also help the user determine spatial relationships between real and digital worlds (in figure

[1] Convention Industrielle de Formation par la Recherche



(a) (b)

Figure 1.3 – (a) The image represents the impact of *PokémonGO* compared to previously existing gaming applications. (b) The images show an augmented scene (virtual white object) using ARCore. The synthetic object fits the overall brightness with the capture of the scene. Nonetheless, it is still distinguishable from the real environment.

1.3-b, the virtual object appears to be "floating" even though it is geometrically well located on the planar surface).

1.2 Problem Description

The goal of a realistic mixed reality application is to make computer-generated objects almost indistinguishable from the real environment when merged into a single image. In order to achieve this goal, several challenging tasks must be handled. Figure 1.4 shows the key steps within the workflow of a mixed reality ecosystem:

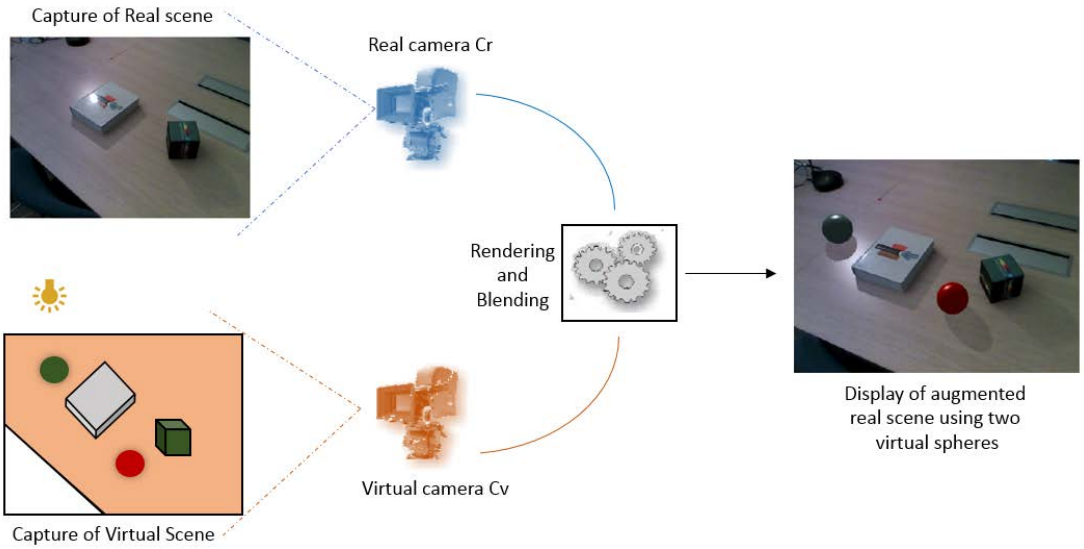


Figure 1.4 – Achieving *realistic* mixed reality requires modeling the camera (field of view, position, orientation) and the real world (geometry, illumination and reflectance). Hence, recovered models can be used within the virtual world to make both worlds seem as if they coexisted in the same environment and were seen through the same camera.

The image captured by any camera (Cr or Cv in figure 1.4) is the result of an interaction between three main quantities: geometry, illumination and reflectance. Geometry

corresponds to the underlying 3D structure of the scene. This structure is often represented by a set of 3D points or oriented polygons. Illumination corresponds to all sources of light in the scene which include direct illumination (bulbs, neon lights, the sun) and indirect illumination (e.g., very reflective surfaces such as mirrors). Illumination is often represented by a set of 3D rays which travel through the 3D space to interact with scene geometry. When an incident light ray hits the surface of a given geometry, the amount of light reflected depends on the reflectance property of the surface. The captured image represents therefore the amount of incident light reflected from the scene towards the camera. In order to achieve realistic MR two challenges must be handled:

1. **Geometric Registration**: the main goal of this registration is to give the MR user the illusion that real and synthetic objects coexist in the same 3D space. In order to achieve this goal, we must first recover the geometry of the real scene. This is an important step as it allows us to handle collisions and occlusions between both worlds (which of the real or synthetic object must be rendered in the camera frustum). Furthermore, the virtual camera Cv which is used to generate a 2D image of the synthetic 3D world must retain the same properties as the real camera Cr (field of view, 3D position, orientation). This step concerns camera calibration and pose estimation.

2. **Photometric Registration**: real objects are illuminated by light sources in the real environment. The way these objects interact with light sources depends on their reflectance properties. In order to achieve a *seamlessly realistic* compositing between real and digital worlds, we must recover the illumination and reflectance of the real scene. This is important as we aim at illuminating virtual objects with virtual light sources which mimic the real ones. Also, to enhance the immersion experience of the MR user, illumination-based interactions between both worlds must be accounted for (e.g., a virtual object casting a shadows that occludes a real specular reflection, color bleeding). When both illumination and reflectance are estimated, we can then virtually add a synthetic object and render the mixed scene using existing computer-graphics rendering techniques [Ritschel et al., 2012]. Such techniques can be global by considering both direct and indirect illumination, or local when only direct illumination is considered.

Throughout the past decade, the depth sensing technology evolved and is nowadays part of several consumer smart-phones/tablets (Google Tango tablet, Intel RealSense sensors, Microsoft Kinect) and HMD devices (Hololens, Magic Leap). This breakthrough has moved geometric and photometric registrations closer providing new possible approaches to achieve realistic MR. In this thesis, we take advantage of existing 3D sensors and adequate existing geometric registration solutions to handle the 3D alignment of the real and digital worlds. **Hence, we focus our work on proposing novel approaches to the estimation of reflectance and illumination using an RGB-D camera.**

Photometric reconstruction can be achieved using laboratory measurement equipment [Loscos et al., 1999]. However, such equipment are not suitable for mixed reality systems because of their limited practicability and out-of-reach pricing. An alternative is

to consider modeling real-world reflectance and illumination using a sequence of RGB-D images of the real scene (RGB images along with Depth maps). The process of recovering these photometric properties from RGB-D images is often referred to as *inverse rendering*. The latter is a highly ill-posed problem with unknowns outnumbering input information provided by the sensor. To tackle this problem, a trade-off is often made between the following points:

- Additional devices: when experiencing MR, the user/camera is looking towards the augmented scene. In this configuration, both illumination and reflectance properties must be recovered from captured images. In order to handle this under-constrained problem, a solution consists in introducing a light probe to capture the illumination in the scene. Light probes correspond to a variety of devices that either look towards the illumination (e.g., fish-eye lenses) or reflect the illumination (e.g., chrome spheres). Such devices are not practical for MR systems that target large-public entertainment applications.
- Scene content: to constrain this *inverse rendering* problem, assumptions within scene properties can be considered. Regarding reflectance, assuming a uniform color surface simplifies the reflectance model. Hence, complex reflectances such as specular or glossy surfaces (shiny objects) and textured surfaces are often discarded. Illumination constraints mainly concern the number of light sources in the scene (reduced to a single one or provided by the user) or their properties. For instance, light sources in indoor scenes can be approximated by a distant distribution where only the orientation has to be recovered. This assumption is often invalid, especially in indoor scenes, where light sources are, unlike the sun, close to the real scene.
- Dynamic changes: in the real world, light sources can be switched on/off or moved. It is therefore necessary to be able to take these changes into account without apparent latency. When assuming static lighting, the user is forced to remain in a controlled environment. This type of configuration is not always suitable for mixed reality applications.
- Processing time: unlike video editing applications where the objective is to provide a visually coherent result without any *real-time* constraints, MR has to be achieved in real-time (near-instantaneous output). Specifically, photometric reconstruction approaches must not introduce any latency within the MR experience. The involved computation, along with the renderings, must be handled in a satisfyingly short period of time (referred to as near real-time or interactive frame rate).

The objective of this thesis is to develop, using a single RGB-D sensor, novel photometric registration algorithms for indoor real scenes. Proposed approaches must be user-friendly, compatible with MR consumer applications and run at interactive frame rate (the user must not notice any visual shift or inaccuracy between real and digital worlds while experiencing the MR scenario). Also, proposed methods must handle scenes with different reflectance properties (e.g., specular surfaces, challenging textures) and recover the properties of a non-distant illumination distribution (3D position, color, intensity). Finally, dynamic changes occurring in the real world must be accounted for.

1.3 Contributions Summary

The considered scenario within the scope of this thesis is the following: using an RGB-D camera, we browse an indoor real scene to acquire its geometry. Light probes or user interaction are not considered. Consequently, by analyzing the color images and acquired model of the scene, our proposed approaches recover surface reflectance properties and illumination characteristics (3D position, intensity, color). Our goal is to handle a large variety of indoor real scene. We therefore consider very few constraints with regard to the content of the scene. Specifically, we do not assume that the scene is composed of uniform-color 3D objects nor that the illumination is distant from the scene or reduced to a single light source. Finally, light sources can be dynamic and our approaches must take into account these changes when illuminating virtual objects. In this work, we are interested in delivering *functional* realism rather than physically-simulated one [Ferwerda, 2003]. Indeed, our goal is to produce a convincing and aesthetic blending between real and digital worlds.

To achieve photometric registration, this thesis focuses on analyzing RGB-D images provided by the sensor. We thus considered four main axes of research that are: **(1) Photometric registration using specular reflections**. **(2) Photometric registration using cast shadows**. **(3) Photometric registration using both specular reflections and cast shadows**. **(4) Detection of specular reflections and cast shadows of indoor real scenes using a deep learning approach**. These four axes of research led to four main contributions that are illustrated in figure 1.5 and are detailed in the following.

(1) Photometric registration using specular reflections.
Within our first contribution, we consider indoor real scenes where both geometry and illumination are static. As the sensor browses the scene, specular reflections can be observed through a sequence of RGB-D images (Figure 1.5). These visual cues are very informative about the scene's illumination and reflectance and have been long considered within photometric registration approaches. In this context, existing techniques often recover specularities as saturated regions in the image. Consequently, bright and white surfaces can be mistakenly considered. Moreover, light sources are often assumed to be distant and only their directions are recovered. However, within indoor real scenes, this assumption is not always valid. Our first contribution addresses these limitations. Specifically, we consider arbitrary real scenes composed of one or more objects with varying textures. We estimate both diffuse and specular reflectance properties using a robust spatio-temporal analysis of the acquired RGB-D sequence. Furthermore, we recover the 3D position of multiple light sources in the scene. Our algorithm has been evaluated on various indoor scenes and allows convincing MR results such as realistic virtual shadows as well as correct real specularity removal.

(2) Photometric registration using cast shadows.
In this contribution, the analysis is based on observed cast shadows in the scene (Figure 1.5). Shadows are omnipresent and result from the occlusion of light by existing

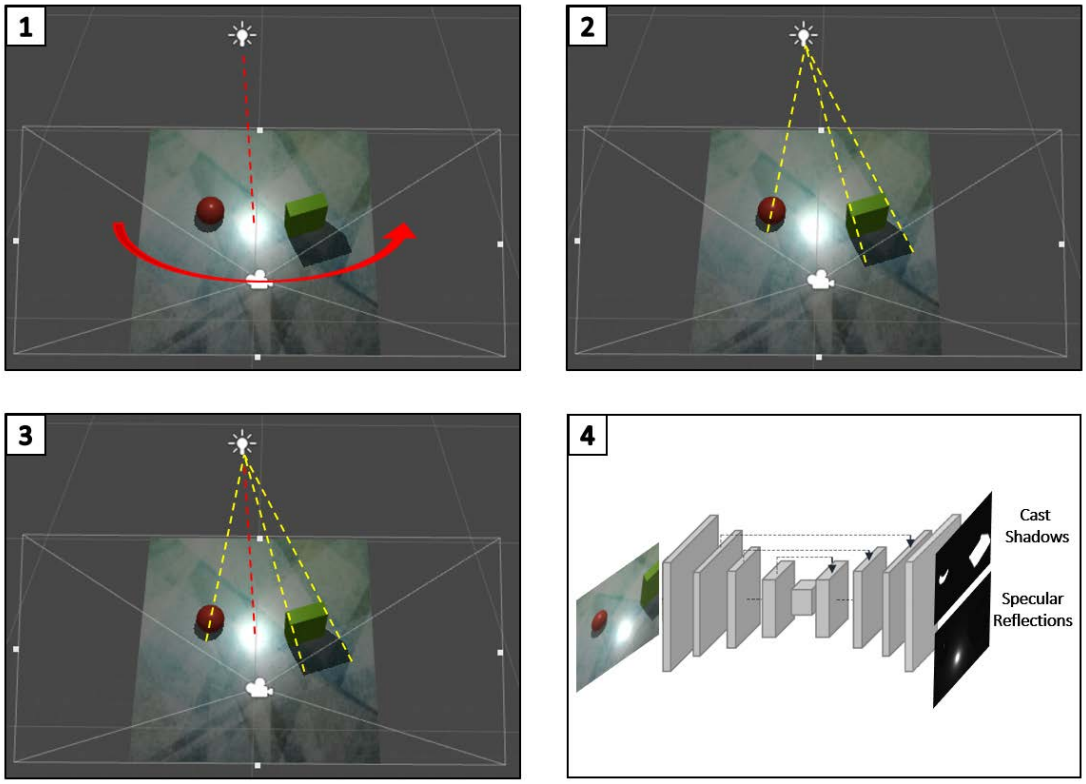


Figure 1.5 – The four contributions of this thesis are depicted in this figure. Contribution (1) corresponds to our photometric registration using specular reflections. Contribution (2) corresponds to our photometric registration using cast shadows. Contribution (3) aimed at estimating reflectance and illumination using both specular reflections and cast shadows. Contribution (4) aimed at detecting specular reflections and cast shadows using a deep learning framework.

geometry. They therefore represent interesting cues to reconstruct the photometric properties of the scene. When indoor scenes are considered, existing solutions often assume uniform-color surfaces to detect shadows. Presence of texture in this context is a challenging scenario. In fact, separating texture from illumination effects is often handled via approaches which require extensive user interaction (e.g., indication of shadows location) or do not satisfy mixed reality requirements (few minutes to detect shadows within a single image). In this contribution, we present a method which tackles these constraints. The proposed approach is twofold: we first separate texture and illumination by considering pairs of points with the same reflectance property but subject to different illumination conditions. Then, from recovered illumination, we estimate the 3D position and intensity of light sources within an iterative process. Our method handles dynamic illumination and runs at an interactive frame rate. Consequently, it is adapted to MR scenarios where the user can freely turn on/off and move the light sources.

(3) Photometric registration using both specular reflections and cast shadows.

In this contribution, we tackle the problem of illumination and reflectance estimation by jointly analysing specular reflections and cast shadows (Figure 1.5). The proposed approach takes advantage of information brought by both cues to handle a large variety of scenes. To illustrate, weak cast shadows are difficult to detect using only shadow-based approaches, however, when specular reflections are available, it is possible to effectively combine both cues to recover illumination. In this contribution, we propose a method which takes advantage of both cues to recover the position and color of multiple light sources. Our approach is capable of handling *any* textured surface and considers both static and dynamic light sources. Its effectiveness is demonstrated through a range of applications including real-time mixed reality scenarios where the rendering of synthetic objects is consistent with the real environment (e.g., correct real specularity removal, visually coherent shadows) and retexturing where the texture of the scene is altered whereas the incident lighting is preserved.

(4) Detection of specular reflections and cast shadows of indoor real scenes using a deep learning approach.
In the previously mentioned contributions, we have explored approaches to effectively detect and model specular reflections as well as cast shadows in order to achieve the photometric registration of real scenes. A last contribution of this thesis was to propose a deep-learning framework to jointly detect specularities and cast shadows in indoor real scenes. Furthermore, within data driven approaches, a key factor to generalization consists in having a dataset with a large variety of scenarios. With regard to our target task, datasets for specular reflection detection are not available and most shadow detection datasets consider outdoor scenes where the sun is the only light source. Hence, we have built a comprehensive and large dataset with the purpose of handling indoor real scene scenarios. Our framework was tested on both our dataset and available benchmarks and achieves good results in both indoor and outdoor scenes.

Publications

The complete publication list is provided below:

- **S. Jiddi**, P. Robert, E. Marchand. Using Specular Reflections and Cast Shadows to Recover Surface Reflectance and Illumination properties in Dynamic Indoor Scenes. In IEEE Transactions on Visualization and Computer Graphics, TVCG. (*Submitted*)
- **S. Jiddi**, P. Robert, E. Marchand. Estimation of position and intensity of dynamic light sources using cast shadows on textured real surfaces. In IEEE International Conference on Image Processing, ICIP'18, Athens, Greece, October 2018.
- **S. Jiddi**, P. Robert, E. Marchand. Photometric Registration using Specular Reflections and Application to Augmented Reality. In Asia Pacific Workshop on Mixed and Augmented Reality, APMAR'18, Taipe, Taiwan, April 2018.
- **S. Jiddi**, P. Robert, E. Marchand. Illumination Estimation using Cast Shadows for Realistic Augmented Reality Applications. In IEEE International Symposium

on Mixed and Augmented Reality(ISMAR-Adjunct), ISMAR'17, Nantes, France, October 2017.

- **S. Jiddi**, P. Robert, E. Marchand. Reflectance and Illumination Estimation for Realistic Augmentations of Real Scenes. In IEEE International Symposium on Mixed and Augmented Reality(ISMAR-Adjunct), ISMAR'16, Merida, Mexico, September 2016.

Demonstrations

The complete demonstrator list is provided below:

- **S. Jiddi**, P. Robert, E. Marchand, A. Laurent, M. Fradet, P. Jouet, C. Baillard. Probeless and Realistic Mixed Reality Application in Presence of Dynamic Light Sources. In IEEE International Symposium on Mixed and Augmented Reality, ISMAR'18, Munich, Germany, October 2018. Demo session: https://youtu.be/sENETegDHnQ
- **S. Jiddi**, P. Robert, E. Marchand, A. Laurent, M. Fradet, C. Baillard. Realistic Mixed Reality Scenarios under Dynamic Lighting and Moving Geometry. In Asia Pacific Workshop on Mixed and Augmented Reality, APMAR'18, Taipe, Taiwan, April 2018. *(Best Demo Award)*. Demo session: https://youtu.be/l6Phgm6C-D8

Patents

This work has been conducted within the MR technical area at Technicolor, where collaborations in several industrial projects have led to the following patents:

Published:

- P. Robert, **S. Jiddi**, M. Hudon. Estimation of specular light source and surface reflectance in a scene from a RGBD sequence, 2015, (EP3144893).
- P. Robert, **S. Jiddi**, M. Hudon. Reflectance parameter estimation in real scenes using an RGBD sequence, 2015, (EP3144893).

Filled:

- P. Robert, **S. Jiddi**, A. Laurent. Matching environment maps from various sources, 2016.
- **S. Jiddi**, P. Robert, L. Tao. Estimation of the 3D position and intensity of light sources using cast shadows, 2017.
- P. Robert, **S. Jiddi**, A. Laurent. Estimation of point light source 3D location and occlusion attenuation, 2017.
- P. Robert, **S. Jiddi**, L. Tao. Estimation of 3D lighting parameters from reference virtual viewpoints, 2018.

- P. Robert, **S. Jiddi**, G. Nieto. Intrinsic image decomposition in presence of textured surfaces for lighting estimation, 2018.
- G. Nieto, P. Robert, **S. Jiddi**. Differentiable Shadow Casting for Point Light Source Estimation, 2018.

1.4 Thesis Structure

This dissertation is divided into 6 technical chapters. The remaining chapters are structured as follows: In Chapter 2, the theoretical background related to the main components of mixed reality is laid out. In Chapter 3, we propose a photometric registration classification and provide an overview of relevant prior art. In Chapter 4, we present our specularity-based approach for photometric registration. In Chapter 5, we present our approach to estimate illumination properties using cast shadows on uniform and/or textured real surfaces. In Chapter 6, we present our generic method to recover both reflectance and illumination in dynamic real scenes with arbitrary material properties. In Chapter 7, we present a data driven approach to detect specular reflections and cast shadows using a deep learning framework. Finally, we provide a review of our research and conclude this thesis.

Background Knowledge

2

Contents

The overarching goal of Mixed Reality (MR) is to provide users with the illusion that virtual and real objects coexist indistinguishably in the same space. An effective illusion requires an accurate registration between both worlds. This registration must be geometrically and photometrically coherent. In this chapter, we present the main concepts and theoretical background related to the ecosystem of realistic MR.

2.1 Mixed Reality Framework

In order to deliver a realistic MR experience, virtual objects must be accurately aligned, in 3D, with the real world. Also, their rendering must be visually consistent with the real environment. This requires modeling the real world and using model properties within the virtual world. Figure 2.1 illustrates how these two worlds can be realistically mixed within a single image. Specifically, the following four components must be modeled:

- **Geometry**: the real world is composed of several 3D objects with different shapes. The position of these objects is expressed in the world frame $\mathcal{F}_w$. The virtual world contains a 3D model which represents the geometry of the real scene as well as virtual objects which augment the real scene. These 3D models can be represented by a set of 3D points whose positions are expressed in a virtual world frame $\mathcal{F}_{vw}$. In the following and without loss of generality, we assume that frames $\mathcal{F}_w$ and $\mathcal{F}_{vw}$ are aligned. Hence, both real and virtual scenes are expressed in the world frame $\mathcal{F}_w$. The creation of a 3D virtual model which matches the real-world environement is of interest within MR scenarios. For instance, when

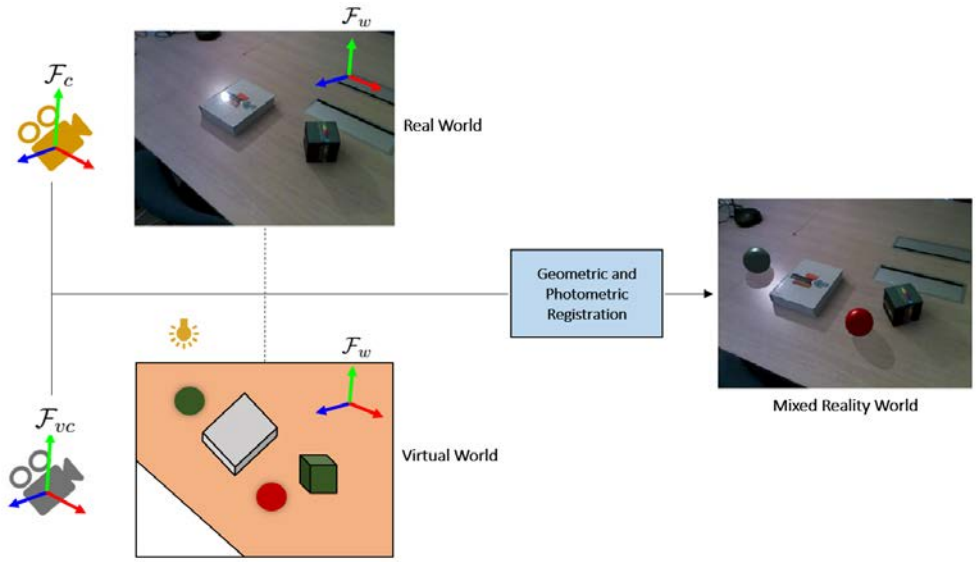


Figure 2.1 – In order to achieve realistic MR scenarios, the digital world must be aligned in 3D with the real world and the appearance of synthetic objects must be consistent with the real environment. These processes are respectively geometric and photometric registrations.

virtual objects are added to the model of the real scene, we are able to handle geometric occlusions between both worlds. The process of recovering scene geometry is referred to as *3D Surface Reconstruction*.

- **Camera**: on the real-world side, a camera captures an image of the real scene. The position of the real camera is expressed in the camera frame $\mathcal{F}_c$. Equally, the virtual scene is viewed through a virtual camera whose position is expressed in the virtual camera frame $\mathcal{F}_{vc}$. To create an image of the virtual world that is consistent with the real camera's current view, virtual and real cameras must be located at the same position, identically oriented and have the same intrinsic parameters (focal, field of view, etc.). Within this process, referred to as *camera pose estimation*, the unknowns are the real camera's position and orientation in the world frame $\mathcal{F}_w$ (the intrinsic parameters are often recovered within a calibration step). This is an important step as inaccurate camera poses result into inconsistencies within the MR scenario (e.g, misalignment errors). When both scene geometry and camera position are estimated, a compositing procedure provides a basic augmented image where real and virtual worlds are geometrically aligned.

- **Illumination**: in order to achieve a realistic compositing, virtual objects must be illuminated by a virtual lighting which mimics the real one. In the real-world, the interaction between existing light sources and scene is automatically captured within the camera's sensor which delivers a 2D color image of the scene. In the virtual world, light sources must retain the same characteristics as the real light sources (number, shape, color, position, etc.). Then, the interaction between virtual light sources, geometry and reflectance is described using a mathematical

reflection model.

- **Reflectance**: The amount of reflected light depends on the surface on which the incident light falls off. This property corresponds to surface reflectance. For instance, a rough surface such as wood reflects light differently compared to a shiny surface such as metal. Estimating real scene reflectance is an important step to achieve realistic mixed reality scenarios. To illustrate, when real specular reflections are occluded by a virtual object's shadow (virtual green sphere in figure 2.1), the reflectance must be accurately reconstructed in this modified region.

Geometric registration corresponds to geometric processes, including 3D surface reconstruction and camera pose estimation. The goal of this registration is to achieve an accurate geometric blending between virtual and real worlds. Algorithms and available tools with regard to geometric registration are presented in section 2.2. Photometric registration or reconstruction is the process of estimating the illumination and surface reflectance properties of an environment, given a geometric model of the scene and a set of photographs of its surfaces [Gibson et al., 2001]. Algorithms and available tools with regard to photometric registration are presented in section 2.3.

2.2 Geometric Registration

In this section, we review the main concepts and algorithms that concern 3D surface reconstruction and camera pose estimation.

2.2.1 Scene Surface Reconstruction

In computer vision, 3D surface reconstruction refers to the process of recovering 3D information of a given scene. This process is very complex due to the diversity of each of the involved parts. These parts concern the scene itself, its lighting, and the sensor that is used for data acquisition. 3D surface reconstruction has been extensively examined since Horn's introduction of Shape from Shading [Horn and Brooks, 1989]. Research in this area led to various approaches and techniques which can be categorized into passive and active methods.

Passive Methods

Passive 3D reconstruction approaches initially originated from the field of photogrammetry and, later on, from the field of computer vision. In contrast to photogrammetry, computer vision applications rely on fast, automatic techniques, sometimes at the expense of precision. Proposed passive 3D reconstruction techniques can be categorized into multiple view and single view approaches.

Within multiple view approaches, the scene is observed from two or more viewpoints. This is achieved by either a single moving camera at different times (structure from motion) or multiple cameras at the same time (stereo). From the collected images, the system aims at recovering the 3D structure of the scene. Figure 2.2 illustrates the

Figure 2.2 – First and second columns are views of a stereo pair. Third column is an outline of the operation of a simple stereo rig using stereo images to recover the 3D structure of the scene.

general concept of inferring the 3D structure of the scene using two viewpoints.

Provided that we can determine the correspondences between left and right image points ($\mathbf{x}$ and $\mathbf{x}'$) referring to the same 3D point, we can determine two directions along which this 3D point lies. The intersection of the these two rays corresponds to the 3D position of the scene point.

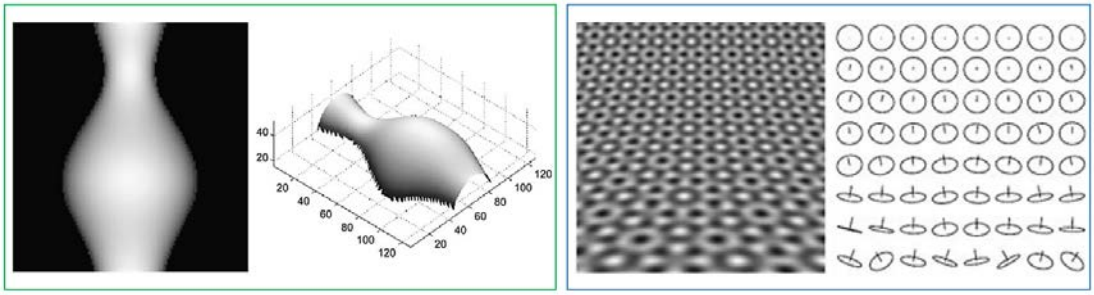


Figure 2.3 – (a) Example of synthetic shape from shading image (left) and corresponding shape from shading reconstruction (right) - Figure of [Horn and Brooks, 1989]. (b) Example of synthetic shape from a texture image (left) and corresponding surface normal estimate (right)- Figure of [Garding, 1992].

Within single view approaches, involved computations are more complex. Such methods are generally based on the analysis of 2D features (e.g., shading, texture, focus) to recover 3D information. For instance, shape from shading techniques [Horn and Brooks, 1989][Huang and Smith, 2009] use the shades in a grayscale image to infer the shape of the surfaces (Figure 2.3). Shape from texture [Garding, 1992] estimates the shape of the observed surface from the distortion of the texture created by the imaging process (Figure 2.3). In shape from focus [Nayar and Nakagawa, 1994], proposed algorithms estimate depth using two input images captured from the same viewpoint but with different camera depths of field. While 3D recovery from a single view is possible, such methods are often not practical in terms of either robustness or speed.

Active Methods

In contrast to passive reconstruction approaches, active 3D sensing measures depth by illuminating the scene with a controlled light source and measuring the backscattered light. Practically, 3D sensors directly provide the geometry of the scene and require minimal operator assistance. There are two types of such sensors: Structured-Light (SL) sensors and Time-Of-Flight (ToF) sensors.

SL sensors are based on a fairly easy principle to understand. In addition to the camera itself, a structured light system adds a light source to illuminate the scene being imaged with patterns. The regular patterns of this light are distorted by the surface of the object, and from this distortion the depth map of the object can be calculated. This core idea was integrated in several end-user sensors such as the Microsoft Kinect v1 and the Intel R200. The Kinect v1 sensor is composed of two cameras, a color RGB and a monochrome infrared (IR) camera, as well as an IR projector. The baseline between the IR projector and the IR camera is 7.5cm (Figure 2.4). The IR projector uses a known and fixed dot pattern to illuminate the scene. Simple triangulation techniques are later on used to compute the depth information between the projected pattern seen by the IR camera and the input pattern stored on the unit. In contrast to the Kinect v1, Intel R200 sensor is a very compact depth camera that can be mounted on laptops and mobile devices (Figure 2.4). It comes with a color camera and a depth camera system. This depth system is composed of two IR cameras and an IR projector.

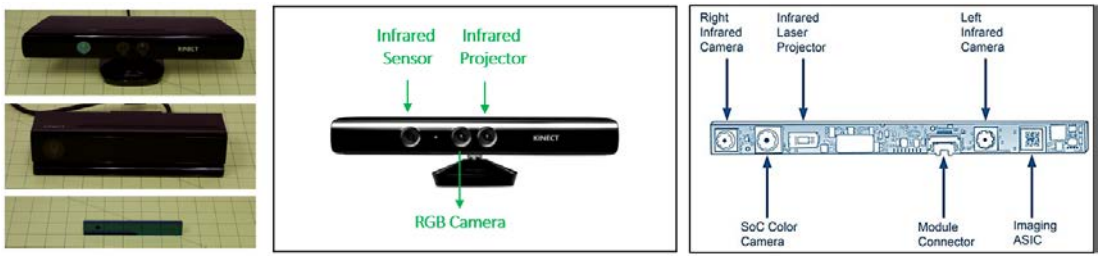


Figure 2.4 – Examples of 3D sensors: (left) Microsoft Kinect v1, Kinect v2 and R200 are represented row-wise (scaled) (center) Main components of the Kinect v1 (right) Main components of Intel R200 sensor.

Time-of-flight sensors measure depth by estimating the time delay between light emission and reflected light detection. In the last decade, this principle has found realization in various microelectronic devices resulting in new range-sensing devices. For instance, the second version of the Kinect sensor (v2) integrated this technology (Figure 2.4). The basic hardware used for Kinect v2 is very similar to the structured-light system (Kinect v1), using a light source and a camera. However, the difference consists in using the Continuous Wave (CW) Intensity Modulation approach. The general idea is to actively illuminate the scene under observation using near infrared (IR) intensity-modulated, periodic light. Due to the distance between the camera and the scene point, and the finite speed of light, a time shift is caused in the optical signal which is equivalent to a phase shift in the periodic signal. The time shift is then transformed into the sensor-object distance as the light has to travel the distance twice.

Representation of 3D information

There are different manners of representing the 3D structure of a given scene. In the following, we present three of the most used and known representations.

The simplest way of representing and storing the 3D coordinates of a scene is a depth map (Figure 2.5). It is a grey scale image generated by a 2D camera except that the depth information replaces the intensity information. This representation is considered to be convenient because retrieving both intensity and depth information can be simultaneously achieved by accessing the same pixel location within respectively the RGB color image and depth measurement image.

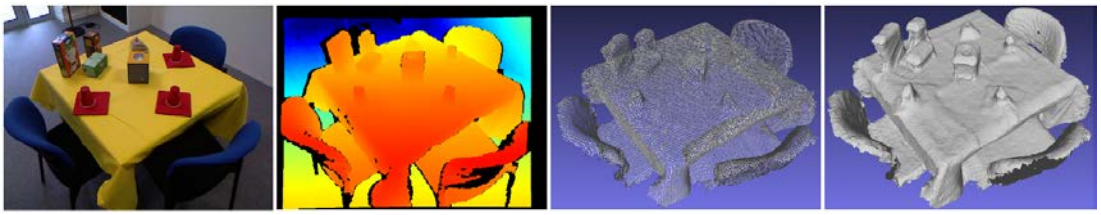


Figure 2.5 – From left to right, RGB image of the scene, its depth map with a specific color-scale (from red: near to blue: far - Black pixels correspond to missing depth due to occlusion), its point cloud and polygon-mesh representations.

In general, 3D active approaches produce a set of points lying on the surface of the scanned scene. The resulting set contains only 3D points and is referred to as a point cloud (Figure 2.5). This model representation can be obtained by merging information from depth maps or by sampling a voxel volume as well.

For mixed reality purposes, point clouds are often converted to polygon (generally triangle) mesh models (Figure 2.5). Techniques to achieve this conversion include Delaunay triangulation, alpha shapes and ball pivoting. This representation is commonly used in computer graphics rendering tasks because it contains a more self-contained representation of the scene (vertices linked by edges and forming shaded polygonal faces) and nowadays graphics boards are optimized for rendering such meshes.

2.2.2 Camera Pose Estimation

Camera pose estimation consists in computing the position and orientation of the camera with respect to the world frame, given a set of correspondences between 3D features (e.g., points, edges) and their projections in the image plane. Solving this ill-posed problem requires modeling the camera and determining the 2D-3D correspondences.

Camera Model

A camera is a device in which the 3D scene is projected down onto a 2D image. Specifically, it maps 3D world points whose coordinates are expressed in standard metric units into the pixel coordinates in the image plane. It is convenient to think of this mapping as a cascade of three successive stages (Figure 2.6): (i) a 6 degree-of-freedom

(DoF) transformation ${}^{c}\mathbf{T}_{w}$ which maps points expressed in the world frame $\mathcal{F}_w$ to the same points expressed in the camera frame $\mathcal{F}_c$. (ii) a projection operation from the 3D world to the 2D image plane. (iii) a mapping from metric image coordinates to pixels coordinates. In the following, we will discuss each of these transformations and mappings.

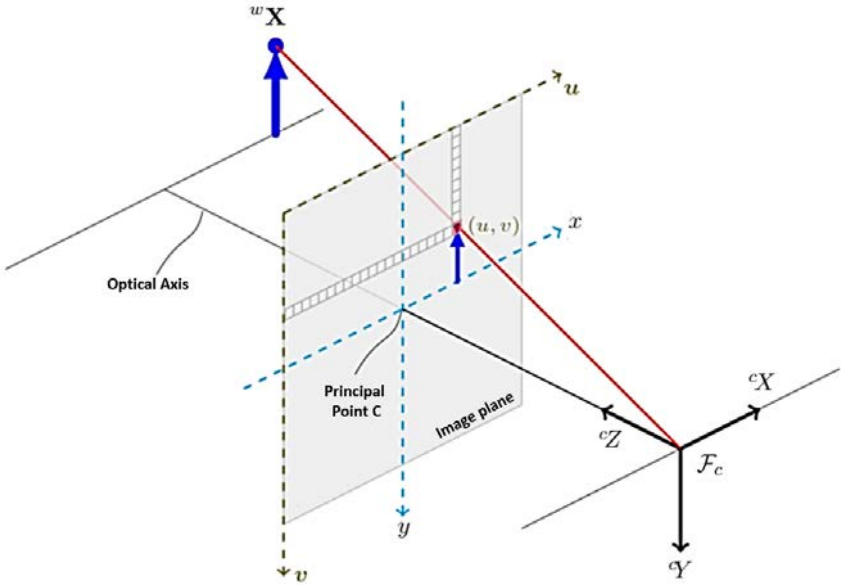


Figure 2.6 – A cascade of three transformations must be applied in order to convert a 3D world point into a pixel in the image.

Camera Model: From World to Camera Coordinates

3D surface reconstruction approaches provide a set of 3D points which represent scene surfaces. Using Euclidean geometry, we can express the position of a given 3D point in the 3D space within a basis of 3 orthonormal unitary vectors $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. The three coordinates ${}^{w}\bar{\mathbf{X}} = (\ {}^{w}X,\ {}^{w}Y,\ {}^{w}Z)^{\top}$ are likely defined first in the world frame $\mathcal{F}_w$ as follows:

$$ {}^{w}\bar{\mathbf{X}} = {}^{w}X\mathbf{i} + {}^{w}Y\mathbf{j} + {}^{w}Z\mathbf{k} \tag{2.1} $$

In order to define the coordinates of the point in the camera frame $\mathcal{F}_c$, we must know the rigid transformation which models the change in position and orientation between both frames $\mathcal{F}_w$ and $\mathcal{F}_c$. The change in position is defined by a 3D translation ${}^{c}\mathbf{t}_{w}$ which transforms the origin of the world frame $\mathcal{F}_w$ into the center of the camera frame $\mathcal{F}_c$. The change in orientation is defined by a 3D rotation ${}^{c}\mathbf{R}_{w}$ which defines the transformation from the axes of the world frame to the axes of the camera frame. Consequently, the coordinates of the 3D point in the camera frame are given by:

$$ {}^{c}\bar{\mathbf{X}} = {}^{c}\mathbf{R}_{w}\,{}^{w}\bar{\mathbf{X}} + {}^{c}\mathbf{t}_{w} \tag{2.2} $$

In the following, we will use the homogeneous coordinates to define the 3D position of a point in a given frame as they allow transformations to be represented as linear mappings. Hence, the Cartesian coordinates ${}^{w}\bar{\mathbf{X}} = (\ {}^{w}X,\ {}^{w}Y,\ {}^{w}Z)^{\top}$ in the Euclidean

space can be defined in a protective space by ${}^{w}\mathbf{X} = ({}^{w}\bar{\mathbf{X}}, 1)^{\top}$. Using the homogeneous coordinate formulation, equation 2.2 can be rewritten as:

$$ {}^{c}\mathbf{X} = {}^{c}\mathbf{T}_{w}\,{}^{w}\mathbf{X} \quad \text{with :} \quad {}^{c}\mathbf{T}_{w} = \begin{pmatrix} {}^{c}\mathbf{R}_{w} & {}^{c}\mathbf{t}_{w} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{2.3} $$

where the homogeneous matrix ${}^{c}\mathbf{T}_{w}$ represents the transformation from the world frame $\mathcal{F}_w$ to the camera frame $\mathcal{F}_c$. ${}^{c}\mathbf{R}_{w}$ is a 3×3 matrix and ${}^{c}\mathbf{t}_{w}$ is a 3×1 vector.

Camera Model: Image Plane Projection

The most commonly used projection in computer vision is the 3D perspective projection. It has its roots in photography where a device, called camera obscura, was used to image the 3D world. The pinhole camera model relies on perspective projection to describe the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane. The image is formed on the plane $Z = f$ where f is the distance from the center $\mathbf{C}$ of the camera to its focal plane. The 2D coordinates $\bar{\mathbf{x}}_m = (x_m, y_m)^{\top}$ in the image are given by the Thales theorem:

$$ \begin{pmatrix} x_m \\ y_m \end{pmatrix} = \frac{f}{{}^{c}Z} \begin{pmatrix} {}^{c}X \\ {}^{c}Y \end{pmatrix} \tag{2.4} $$

where ${}^{c}\bar{\mathbf{X}} = ({}^{c}X,\ {}^{c}Y,\ {}^{c}Z)$ are the coordinates of the point in the camera frame $\mathcal{F}_c$. Using the homogeneous coordinates, we rewrite the projection operation in a linear form:

$$ \mathbf{x}_m = \mathbf{A}\,{}^{c}\mathbf{X} \quad \text{with :} \quad \mathbf{A} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.5} $$

where $\mathbf{x}_m = (\bar{\mathbf{x}}_m, 1)^{\top}$ are the 2D homogeneous coordinates corresponding to the location of the 3D point on the projection image plane.

Camera Model: Pixel Space

An image is basically a grid of pixels. It is defined by its width w, its height h and its origin located at the corner of the sensor. To convert the 2D coordinates of a point's projection from meters to pixels, we need to apply the following transformation which takes into account the coordinates (u_0, v_0) of the principal point (corresponds to $(0, 0)$ in the 2D meter space) and the pixel size (l_x, l_y) on the sensor:

$$ \begin{cases} u = u_0 + \frac{1}{l_x} x \\ v = v_0 + \frac{1}{l_y} y \end{cases} \tag{2.6} $$

Finally, to convert a 3D point from the camera frame $\mathcal{F}_c$ to its 2D pixel coordinates, the following transformation is applied:

$$\begin{aligned} \mathbf{x} = \mathbf{K}\Pi^c\mathbf{X} \quad \text{with :} \quad \Pi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ \text{and :} \quad \mathbf{K} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{2.7}$$

where p_x (respectively p_y) is the ratio between the focal length f and the pixel width (respectively height). $\mathbf{K}$ contains all the parameters peculiar to the camera, and is called the intrinsic matrix. To sum up, the projection in the image plane of a given 3D point defined in the world frame $\mathcal{F}_w$ is a concatenation of the following three mappings:

$$\mathbf{x} = \mathbf{K}\Pi^c\mathbf{T}_w{}^w\mathbf{X} \tag{2.8}$$

Typical cameras have a lens distortion, which disrupts the assumed linear projective model (equation 2.4). Thus a camera may not be accurately represented by the pinhole camera model that we have described, particularly if a low-cost lens or a wide field-of-view(short focal length) lens such as a fisheye lens is employed. The effect is non-linear and, if significant, it must be corrected so that the camera can again be modeled as a linear device. The process of finding the camera's intrinsic parameters $\mathbf{K}$ is referred to as *camera calibration*. It is generally performed using a set of images where some known 3D points ${}^w\mathbf{X}$ are projected at known positions $\mathbf{x}$ (identified via detection). This defines a system of equations from which the parameters (u_0, v_0, p_x, p_y) are recovered. A number of publicly available camera calibration packages (including the non-linear case) is available on the web, such as the Caltech camera calibration toolbox for MATLAB and in the OpenCV computer vision library.

2.2.2.1 Approaches Overview for Pose Estimation

Given a calibrated camera (known intrinsic matrix $\mathbf{K}$) and a 3D model of the scene, the camera pose estimation consists in recovering the full transformation ${}^c\mathbf{T}_w$ which maps 2D image coordinates $\mathbf{x}$ (or their corresponding 3D points ${}^c\mathbf{X}$) to 3D world points ${}^w\mathbf{X}$. In the following, we will review various approaches allowing to solve this problem.

3D-3D Registration

When 3D coordinates of observed points are available in both camera $\mathcal{F}_c$ and world $\mathcal{F}_w$ frames, the registration can be done directly in the 3D space, also referred to as 3D-3D registration. Denoting $\mathbf{q}$ a minimal representation of ${}^c\mathbf{T}_w$ (e.g., $\mathbf{q} = ({}^c\mathbf{t}_w, \theta\mathbf{u})$, where θ and $\mathbf{u}$ are the angle and the axis of the rotation ${}^c\mathbf{R}_w$) , the problem can be reformulated as follows:

$$\hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \sum_{i=1}^{\mathrm{N}} ({}^c\mathbf{X}_i - {}^c\mathbf{T}_w{}^w\mathbf{X}_i)^2 \tag{2.9}$$

Solving equation 2.9 can be achieved using an iterative minimization algorithm such as Gauss-Newton or Levenberg-Marquart methods. When the 3D correspondences are unknown, the Iterative Closest Point (ICP) technique [Besl and McKay, 1992] can be considered to solve this problem.

2D-3D Registration

Within 2D-3D registration, the problem of camera pose estimation consists in recovering the transformation ${}^{c}\mathbf{T}_{w}$ which maps a set of N correspondences between 2D image coordinates $\mathbf{x}$ and 3D world points ${}^{w}\mathbf{X}$. Proposed methods can be categorized into marker-based (or keypoint-based) approaches and marker-less approaches.

Keypoint-based approaches rely on a two-step solution. First an estimation of the unknown depth of each keypoint in the camera frame is achieved (3D model is usually expressed in the world frame). Once the N points coordinates are known in the camera frame, the second step consists in estimating the rigid transformation ${}^{c}\mathbf{T}_{w}$ that maps the coordinates expressed in the camera frame to the coordinates expressed in the world frame (3D-3D registration). In addition to these two-step solutions, there exist direct (or one-step) approaches such as the Direct Linear Transform (DLT) [Hartley and Zisserman, 2003] which recover camera pose by solving a linear system built by considering N correspondences within equation 2.8. These solutions are not accurate because this problem is intrinsically non-linear. Consequently, a more accurate solution consists in minimizing the norm of the re-projection error as follows:

$$\hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \sum_{i=1}^{\mathrm{N}} d(\mathbf{x}_i, \mathbf{K}\Pi\, {}^{c}\mathbf{T}_{w}\, {}^{w}\mathbf{X}_i) \tag{2.10}$$

where $d(\mathbf{x}_1, \mathbf{x}_2)$ is the Euclidean distance between two points $\mathbf{x}_1$ and $\mathbf{x}_2$. The solution to this problem can be achieved using a non-linear minimization algorithm such Gauss-Newton or Levenberg-Marquart.

On the other hand, markerless approaches do not require any marker or keypoints matching process. In fact, they mainly rely on the following key idea: instead of considering distance between the image coordinates of two keypoints , the distance between a contour point in the image and the projected 3D line $\mathbf{L}(\mathbf{q})$, using the 3D model and camera pose $\mathbf{q}$, is considered (Figure 2.7). Recovering camera pose is achieved by considering the following minimization:

$$\hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \sum_{i=1}^{\mathrm{N}} d^{*}(\mathbf{L}(\mathbf{q}), \mathbf{x}_i) \tag{2.11}$$

where $d^{*}(\mathbf{L}(\mathbf{q}), \mathbf{x}_i)$ is the squared distance between the point $\mathbf{x}_i$ and the projection of the contour $\mathbf{L}$ of the model for the pose $\mathbf{q}$. This core idea has been extensively used to propose various markerless camera pose estimators [Choi and Christensen, 2012], [Comport et al., 2006], [Drummond and Cipolla, 2002], [Lowe, 1991].

Simultaneous Localization And Mapping (SLAM)

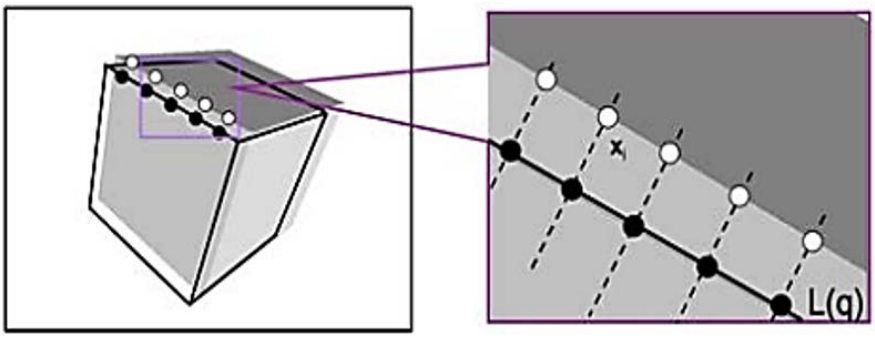


Figure 2.7 – Core idea of markerless approaches where geometric contours are considered to recover the camera pose - Figure from [Marchand et al., 2016].

As previously described, 3D-3D and 2D-3D registration approaches require a 3D model of the observed points to recover camera pose. When 3D data is not available, methods which do not require any 3D knowledge about the observed scene can be considered. The idea consists in estimating, from a sequence of images, the scene structure and the camera position at the same time. Denoting $[\mathbf{q}]_t = (\mathbf{q}_1 ... \mathbf{q}_t)$ a sequence of t camera positions and $[{}^w\mathbf{X}]_N = ({}^w\mathbf{X}_1 ... {}^w\mathbf{X}_N)$ a set of N points, the problem can be formulated as follows:

$$([\hat{\mathbf{q}}]_t, [{}^w\hat{\mathbf{X}}]_N) = \underset{([\mathbf{q}]_t, [{}^w\mathbf{X}]_N)}{\arg\min} \sum_{j=1}^{\mathrm{t}} \sum_{i=1}^{\mathrm{N}} d(\mathbf{x}_i, \mathbf{K}\Pi^j \mathbf{T}_w \, {}^w\mathbf{X}_i) \tag{2.12}$$

This problem originally known as the structure from motion issue was handled off-line due to the high computational complexity of the solution. Research in this area has led to more efficient and fast SLAM solutions [Triggs et al., 1999][Klein and Murray, 2007] [Mouragnon et al., 2006].

Recently, several techniques tackled the same problem and additionnaly used an RGB-D sensor to acquire partial 3D knowledge about the scene [Newcombe et al., 2011b][Newcombe et al., 2011a]. The latter are more adequate for MR specifications (Figure 2.8-a) as they can run in real-time and require light-weighted hand-held devices instead of previously considered complex setups (e.g., laser range sensors, rotary encoders, inertial sensors, cameras).

2D-2D Registration

All previously described geometric registrations rely on some knowledge of the 3D structure of the observed scene. An alternative is to infer the camera pose using image processing techniques. The appearance-based approaches, also known as template-based approaches, only rely on captured 2D images of the scene. Specifically, these techniques use a 2D model (reference image or reference template) to estimate the camera motion between the current and reference images at the pixel intensity level (Figure 2.8-b). In the following, we consider that the 2D template model is represented by a region $w(\mathbf{x})$ in a reference image I_0. The goal is to find, within the current image I, the template's new location $w(\mathbf{x}, \mathbf{h})$ where $\mathbf{h}$ are the parameters of the motion model

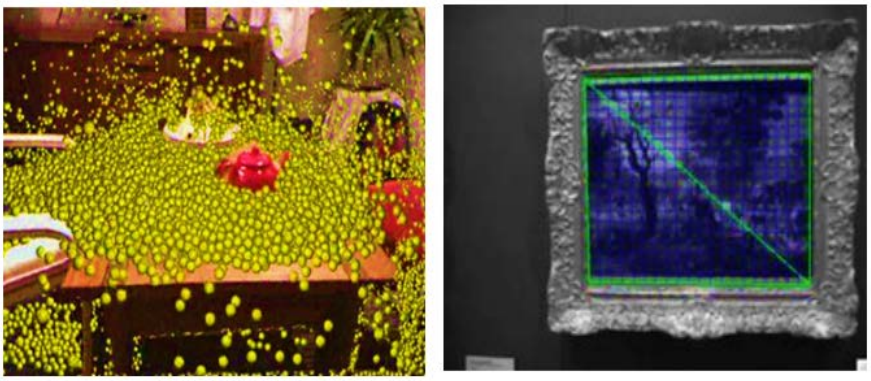


Figure 2.8 – (a) Results of KinectFusion [Newcombe et al., 2011a], a fast SLAM approach, where green particles interact with the recovered geometry of the scene. (b) 2D-2D registration example where the painting (framed by virtual green lines) represents the considered template - Figure of [Tillon et al., 2010].

(usually described using the homography). This problem can be formulated as follows:

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \sum_{x \in w} \mathrm{f}(I_0(w(\mathbf{x})), I(w(\mathbf{x}, \mathbf{h})) \tag{2.13}$$

where f is the dissimilarity function. A basic choice of f corresponds to considering the sum of squared differences (SSD) within pixels intensities [Lucas and Kanade, 1981]. More sophisticated template trackers have been proposed in order to efficiently handle image blur [Park et al., 2012], illumination and occlusions [Irani and Anandan, 1998][Dame and Marchand, 2010].

2.2.2.2 2D-2D Correspondences

Within several camera pose estimation methods, we need to establish correspondences between 2D points in images. This is usually achieved by considering a set of salient points or keypoints. The procedure is threefold: the basic idea is to first detect keypoints. Then, an invariant feature representation (descriptor) for image data around the detected keypoint is built. Finally, a step of features matching is performed.

In general, within a captured image, two types of image features can be extracted, namely global features and local features. Global features (e.g., color and texture) aim to describe an image as a whole and can be interpreted as a particular property of the image involving all pixels. On the other hand, local features aim to detect keypoints or interest regions in an image and describe them. As real time applications have to handle large amounts of data and/or run on mobile devices with limited computational capabilities, there is a growing need for local descriptors that are fast to compute, fast to match, memory efficient, and yet exhibiting good accuracy. Historically, Harris detector [Harris and Stephens, 1988] is a commonly used local feature detector that computes the cornerness score of each pixel from gradients of an image patch. The cornerness score is then classified into flat, edge and corner according to the intensity

structure of the patch. Various alternatives to this detector have been proposed to handle cornerness detection differently [Smith and Brady, 1997], lower the processing time [Rosten et al., 2010] and deal with scale-invariance within images [Lindeberg, 1998][Lowe, 2004][Bay et al., 2006][Alcantarilla et al., 2012].

Once a keypoint is detected, the next step consists in computing a feature vector that fully describes the keypoint along with its local neighbors. These vectors can be classified into two main categories: histogram of oriented gradients approaches and intensity approaches. Histograms of oriented gradients are computed within small patches in the image and are then concatenated to represent the whole image. Thus, shape and intensity information are preseved within this image representation. This process has been extensively used in common descriptors such as SIFT [Lowe, 2004], SURF[Bay et al., 2006] and CARD[Ambai and Yoshida, 2011]. Intensity comparisons based approach consists in computing and storing comparisons between pairwise pixels intensities. For instance, this approach has been considered within BRIEF [Chaumette and Hutchinson, 2006], ORB [Rublee et al., 2011] and BRISK [Leutenegger et al., 2011].

The last step consists in matching the detected keypoints using the computed descriptors. Within this process, a nearest neighbor searching approach is considered in order to find the closest descriptor in the reference image that matches the current descriptor.

2.3 Photometric Registration

The previous section describes the main geometric registration components (3D surface reconstruction and camera pose estimation). These processes allow a 3D alignment between the real world with the digital world. In order to *realistically* blend both worlds, one must handle the photometric registration problem as well. This requires modeling the interaction between geometry, light and surface reflectance in both the physical and virtual worlds. In this section, we give an overview of the imaging pipeline that starts with the acquisition of a real world scene or with the rendering of an abstract model using computer graphics techniques and results in a 2D image of the considered scene.

2.3.1 Terminology

In the following, we present several physical and perceptual quantities important for digital imaging.

Radiometry

Radiometry is the science concerned with light measurement. In this context, light is represented by a radiant energy Q_e which is measured in Joules (J). Since light propagates through different media (e.g., space, air, water), it is necessary to model the way it propagates through these environments within time and space. Integrating

radiant energy over time is referred to as radiant flux or radiant power P_e:

$$P_e = \frac{dQ_e}{dt} \tag{2.14}$$

It is measured in Joules per second (J/s) or Watts (W) and represents therefore a measure of energy per unit of time. Integrating radiant flux over space, per unit area dA, is referred to as radiant flux density. Radiant flux density is also known as irradiance E_e if we are considering the flux arriving from all possible directions at a point on a surface (Figure 2.9 - **a**) and as radiant exitance M_e for the flux leaving a point on a surface in all possible directions (Figure 2.9 - **b**):

$$\begin{cases} E_e = \frac{dP_e}{dA} \\ \\ M_e = \frac{dP_e}{dA} \end{cases} \tag{2.15}$$

Both irradiance and radiant exitance are measured in Watts per square meter (W/m^2). They are thus measures of energy per unit of time and per unit of area.

Figure 2.9 – (a) Irradiance: radiant flux density considering flux arriving from all possible directions upon unit area dA. (b) Radiant exitance: radiant flux density considering flux leaving the unit area dA. (c) Radiant intensity: light power emitted per unit solid angle $d\omega$. (d) Radiance: incident light upon a unit area dA from a unit direction represented by the solid angle $d\omega$.

If we consider a point light source (Figure 2.9 - **c**), the light emitted into a particular direction is called radiant intensity I_e:

$$I_e = \frac{dP_e}{d\omega} \tag{2.16}$$

and is measured in Watts per steradian (W/sr). A steradian is a measure of solid angle corresponding to area on the unit sphere. The flux leaving or arriving at a point in a particular direction is known as radiance L_e:

$$L_e = \frac{d^2P_e}{dAcos\theta d\omega} \tag{2.17}$$

and is measured in Watts per square meter per steradian (Figure 2.9 - **d**) and represents a measure of energy per unit of time as well as per unit of area and per unit of direction.

Photometry

The human eye is sensitive to wavelengths λ varying between 380 and 830 nm. Within this range, it is not equally sensitive to all wavelengths and the sensitivity is not the

same for all individuals. Nevertheless, these variations are small enough to approximate the spectral sensitivity of any human observer with a single curve, known as "$V(\lambda)$ curve" (Figure 2.10). Since, we are interested in the way a human observer perceives light, its spectral composition may be weighted according to $V(\lambda)$. The science of measuring light weighted with regard to $V(\lambda)$ is called photometry.

All previously presented radiometric quantities have their photometric counterparts. By spectrally weighting radiometric quantities with $V(\lambda)$, they are converted into photometric quantities. For instance, luminous flux or luminous power P_v is the photometrically weighted radiant flux P_e and it is measured in lumens (lm). One of the most

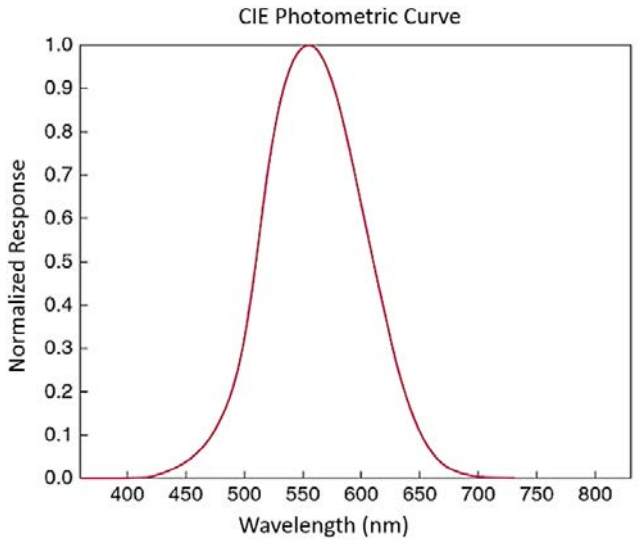


Figure 2.10 – CIE photoptic luminous efficiency curve, also known as the $V(\lambda)$, which represents the sensitivity of any human observer with regard to visible light.

important light derivatives is luminance. It corresponds to photometrically weighted radiance and constitutes an approximate measure of how bright a surface appears. Spectrally weighting radiance corresponds to multiplying each spectral component with the corresponding value given by the weight function $V(\lambda)$ and then integrating over all visible wavelengths.

Bidirectional Reflection Distribution Function (BRDF)

As previously stated, radiance (or luminance) corresponds to the flow of light traveling between two surfaces. When light, represented by a set of light rays, hits an opaque surface, it can be either absorbed and converted into thermal energy, or reflected into some direction. The amount of light reflected as well as the direction in which it is reflected depends on a particular surface property named reflectance. To illustrate, matte surfaces reflect light almost evently in all directions, whereas glossy and shiny surfaces reflect light in a preferred direction. Mirrors are the opposite of matte surfaces and emit light into almost a single direction (Figure 2.11).

For the purpose of image formation, the exact distribution of light reflected off sur-

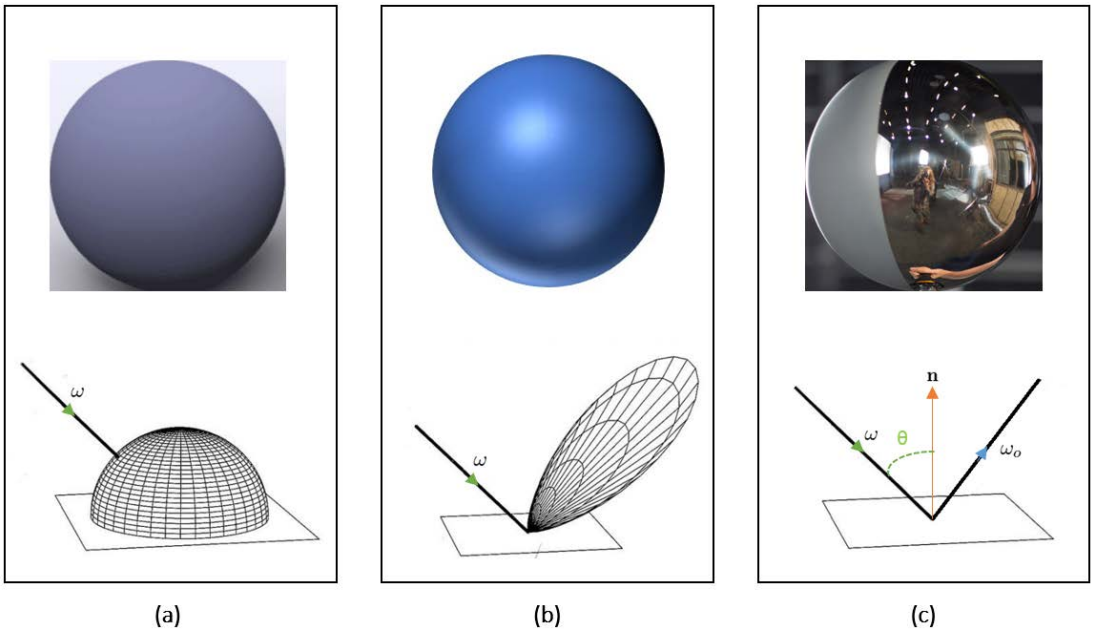


Figure 2.11 – (a) Matte surfaces, referred to as Lambertian or diffuse surfaces, reflect light equally in all directions. (b) Glossy surfaces reflect light in a preferred direction, generally comprised within a lobe. (c) Mirror surfaces reflect light in a single direction.

faces is modeled with a bidirectional reflection distribution function $f_r(\mathbf{X}, \omega, \omega_o)$ which depends on the incident light direction ω and outgoing light direction ω_o at a surface point $\mathbf{X}$:

$$f_r(\mathbf{X}, \omega, \omega_o) = \frac{dL_o(\mathbf{X}, \omega_o)}{L(\mathbf{X}, \omega) cos\theta d\omega} \tag{2.18}$$

where L is the incident radiance and L_o is the outgoing reflected radiance. A physically plausible BRDF must maintain three important properties:

1. The BRDF is a positive function:

$$f_r(\mathbf{X}, \omega, \omega_o) \geq 0 \tag{2.19}$$

2. The BRDF must follow the Helmholtz reciprocity principle: if the incident and reflected light directions are reversed, the BRDF must stay the same:

$$f_r(\mathbf{X}, \omega, \omega_o) = f_r(\mathbf{X}, \omega_o, \omega) \tag{2.20}$$

3. The BRDF must uphold the law of conservation of energy. Therefore the outgoing radiance must be less than or equal to the incoming radiance:

$$\int_\Omega f_r(\mathbf{X}, \omega, \omega_o) cos\theta d\omega \leq 1.0 \tag{2.21}$$

where Ω comprises all incoming/incident light directions and θ is the angle between the incident light direction ω and the point's normal direction $\mathbf{n}$. When the BRDF is integrated over the entire space of incident directions, we obtain the total reflectance of the surface point $\mathbf{X}$.

Rendering Equation

The *rendering equation* was first introduced by [Kajiya, 1986]. It is an analytic formulation which fully describes the interaction between light, geometry and surface reflectance. Using equation 2.18, the flow of light throughout an environment can be rewritten as:

$$dL_o(\mathbf{X}, \omega_r) = f_r(\mathbf{X}, \omega, \omega_o) L(\mathbf{X}, \omega) cos\theta d\omega \tag{2.22}$$

where dL_o is the outgoing reflected radiance with regard to the incoming radiance L of one light ray. The total reflected radiance at a scene point $\mathbf{X}$, in the outgoing direction ω_o , is referred to as surface radiance R_s and can be described by the rendering equation here after:

$$R_s(\mathbf{X}, \omega_o) = L_e(\mathbf{X}, \omega_o) + \int_\Omega f_r(\mathbf{X}, \omega, \omega_o) L(\mathbf{X}, \omega) cos\theta d\omega \tag{2.23}$$

where Le is the emitted radiance. In fact, some scene surfaces (e.g., light sources) can reflect and emit light at the same time. The term $cos\theta$ is equal to the dot product $(\mathbf{n} \cdot -\omega)$ where $\mathbf{n}$ is the normal vector of point $\mathbf{X}$ and ω is the incident/incoming light direction.

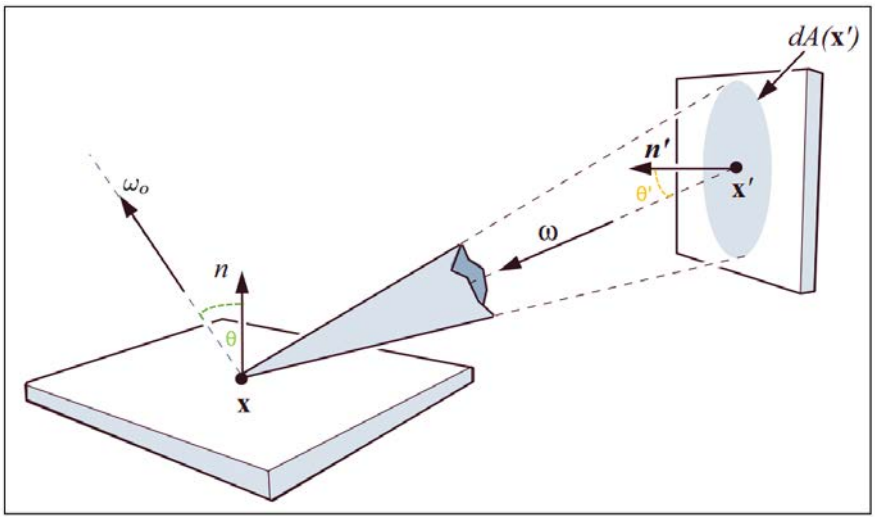


Figure 2.12 – Geometry of the Rendering Equation: light rays originating from the unit surface dA' and traveling in the direction ω hit the scene point $\mathbf{X}$. These light rays are then reflected in direction ω_o.

Equation 2.23 can be expressed using geometric relationships between emitting and receiving surfaces (Figure 2.12):

$$R_s(\mathbf{X}, \omega_o) = L_e(\mathbf{X}, \omega_o) + \int_\Omega g(\mathbf{X}, \mathbf{X}') f_r(\mathbf{X}, \omega, \omega_o) L(\mathbf{X}, \omega) \frac{cos\theta cos\theta' dA}{\|\mathbf{X} - \mathbf{X}'\|^2} \tag{2.24}$$

where $\|\mathbf{X} - \mathbf{X}'\|^2$ is the distance from point $\mathbf{X}$ to point $\mathbf{X}'$ and $g(\mathbf{X}, \mathbf{X}')$ is the occlusion term (equal to 0 if $\mathbf{X}$ is visible to $\mathbf{X}'$ and 0 otherwise). g is used to account for the fact that some surfaces might be blocked with regard to light rays.

2.3.2 Virtual Image Formation

One of the computer graphics goals is the production of realistic synthetic images from digital object models (Figure 2.13). The generation of such realistic images relies on the numerical computation of approximations to the rendering equation 2.24. In fact, solving this equation analytically is impossible in most 3D scenes [Dutre et al., 2006]. One way to approximate the solution of this problem is by using Monte Carlo

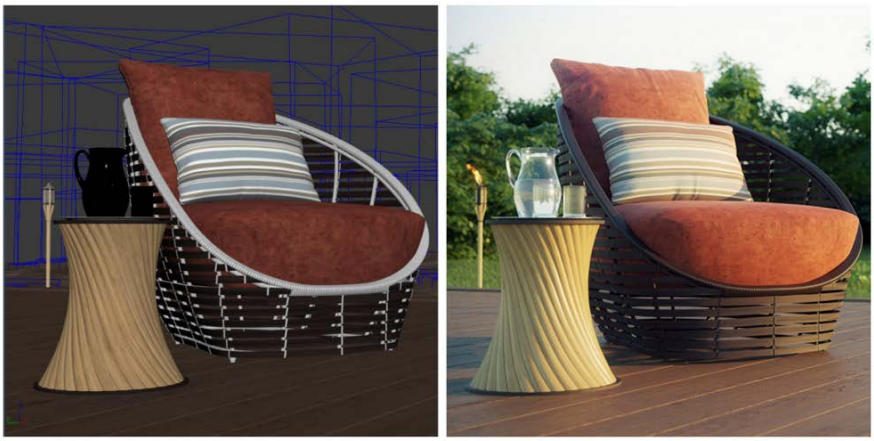


Figure 2.13 – Modeled virtual scene in terms of geometry and reflectance (left) combined with illumination and its photorealistic created image using computer graphics rendering techniques - Figure from [Vorba and Karlik, 2012]

integration where N random samples of light ray paths are generated according to a probability density function. Because evaluating these samples can be computationally expensive, high-quality images can therefore take hours, or even days to compute. An alternative consists in modeling light, reflectance and geometry interactions within simplified *reflection models* that respect the time-constraints for real-time applications such as MR.

Light Models

In the physical world, light sources are 3D objects which have the property of emitting light. Hence, they retain the same characteristics as any common object in the scene (position, orientation, size, shape). When light rays originating from a light source illuminate an object, this object is considered to be lit by a *direct illumination* or *direct lighting*. On the other hand, when the emitted light falls on a surface **A** which reflects a proportional amount of light towards surface **B**, surface **B** is then lit by an *indirect illumination* (Figure 2.14).

In the following, we present some of the most known and used light source models (Figure 2.15):

- **Ambient light** does not have any identifiable source position or orientation. It is often used as a secondary light source to mimic the effect of indirect lighting, where light is scattered off surfaces. When using only an ambient light source to illuminate a scene, the latter appears to be flat.

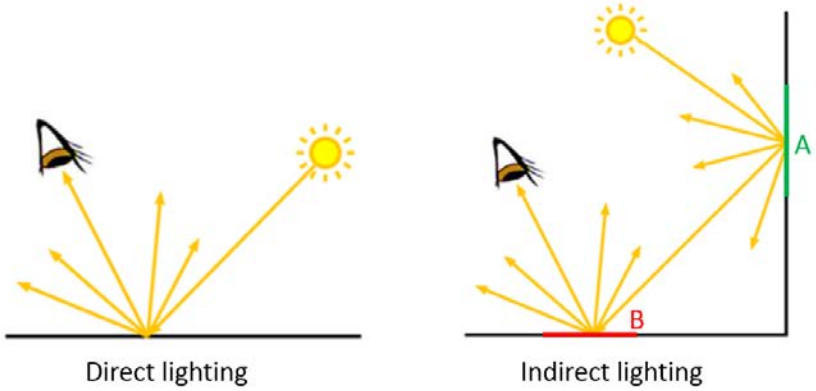


Figure 2.14 – Direct illumination (left): the human observer (or camera) sees the light reflected by the objects directly lit by light sources. indirect Illumination (right): the light, emitted by the source, is reflected first at surface A, then surface B reflects the incident light towards the observer.

- **Directional light**, also referred to as distant or infinite light, simulates a light source which is located far away from the lit scene (e.g., the sun). It is considered to be so far from the scene that its rays reach the surface in a parallel form. It does not have any identifiable source position and only retains a defined orientation.

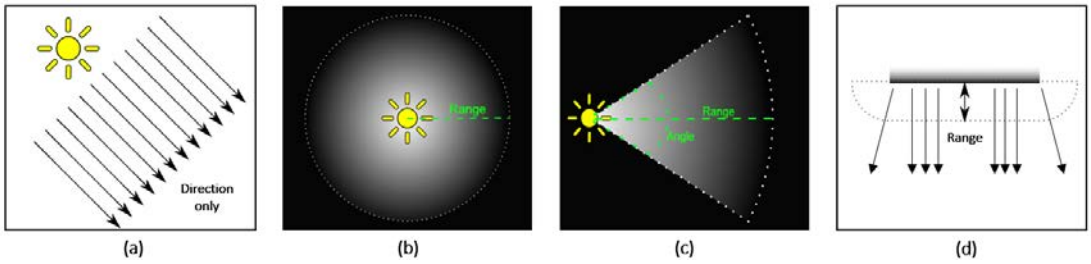


Figure 2.15 – Examples of light source models where (a) is a directional light, (b) is a point light, (c) is a spot light and (d) corresponds to a rectangular area light.

- **Point light**, also referred to as omni-directional light, simulates rays shining out from a single point in space in all directions. It can mimic the effect given by an omni-directional local light source such as light bulbs and candles. It is located at a point in space and sends light out in all directions equally. The intensity diminishes with distance from the light, reaching zero at a specified range.

- **Spot light** simulates light radiating from a single point in space and has a cone of influence in a specific direction. It can be controlled conveniently to aim at a specific target. Like a point light, a spot light has a specified location and range over which the light falls off.

- **Area light** has a definable size. It simulates a realistic soft lighting distribution and realistic shadows that vary from hard to soft. It is defined by a rectangle in space and emits in all directions uniformly across its surface area.

Another interesting way of representing illumination in the scene is an environment map, also referred to as reflection map. It is an efficient way for approximating the appearance of a reflective surface by means of a pre-computed texture image of the *distant* environment surrounding the rendered object. Several ways of storing an environment

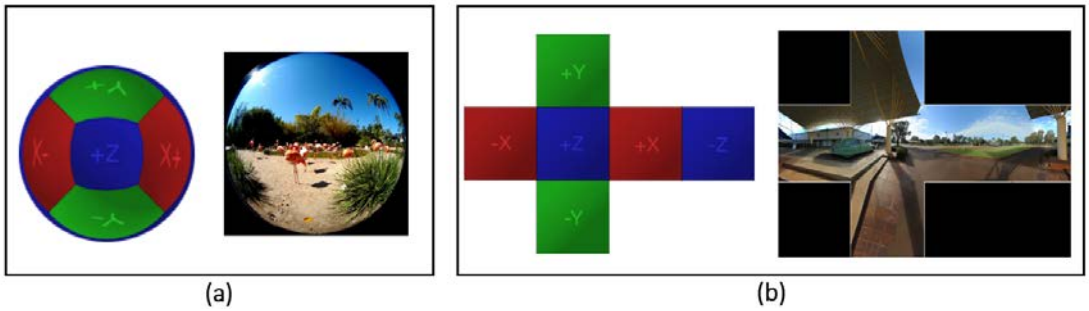


Figure 2.16 – Examples of environment map textures with sphere mapping in (a) and cube mapping in (b) representing the surrounding environment of the captured scene.

map can be employed: the first technique is sphere mapping, in which a single texture contains the image of the surroundings as reflected on a mirror ball. It has been almost entirely surpassed by cube mapping, in which the environment is projected onto the six faces of a cube and stored as six square textures or unfolded into six square regions of a single texture (Figure 2.16).

BRDF Models

Various BRDF models have been proposed in order to cover different surface reflective properties that one might encounter in the real world. Some of these are data-driven, some are analytic. In the following, we will give an overview of the analytic models which can be categorized into physical and empirical models. Figure 2.17 shows the vector system involved in these BRDF models description for a point located at $\mathbf{X}$ in the 3D space: ω is the incident light source direction, $\mathbf{n}$ is the normal vector of the point and $\mathbf{r}$ is the perfect reflection vector with regard to ω. Also, ω_o is the outgoing direction and, it generally corresponds to the observation viewpoint $\mathbf{v}$ (in this case $\omega_o = \mathbf{v}$).

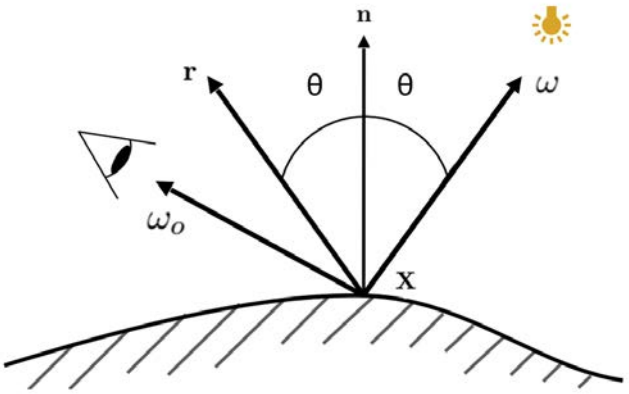


Figure 2.17 – Considered vectors within the BRDF model description.

Physically-based models try to accurately simulate light scattering by using physics laws. They usually lead to complex formulations and high computational effort. The simplest ones are the ideal specular and ideal diffuse reflections. In case of ideal specular reflection, light coming from a given direction is reflected into a single direction following the *law of reflection*. The BRDF in this case is a delta dirac distribution δ, giving always zero, except when $\mathbf{r}$ and ω_o are aligned:

$$f_r(\mathbf{X}, \omega, \omega_o) = \mathbf{k_s}\delta(\mathbf{r}, \omega_o) \tag{2.25}$$

where $\mathbf{k_s}$ is the specular reflectance at point $\mathbf{X}$. A diffuse surface has a BRDF that has the same value for all incident and outgoing directions. This substantially reduces the computations and thus it is commonly used to model diffuse surfaces as it is physically plausible, even though there are no pure diffuse materials in the real world. This BRDF is expressed as:

$$f_r(\mathbf{X}, \omega, \omega_o) = \frac{\mathbf{k_d}}{\pi} \tag{2.26}$$

where $\mathbf{k_d}$ is the diffuse reflectance at point $\mathbf{X}$. One of the most complete physical reflection models is the Torrance-Sparrow BRDF. The roughness is modeled using microscopic concavities in V-form of equal length called microfacets. Their orientation is random and their distribution is controlled by parameters, so it is possible to simulate different degrees of roughness. The complete BRDF function is :

$$f_r(\mathbf{X}, \omega, \omega_o) = \frac{\mathbf{k_d}}{\pi} + DFG\frac{\mathbf{k_s}}{4\pi(\mathbf{n} \cdot \omega)} \tag{2.27}$$

D is the microfacets distribution, F is the Fresnel factor which gives the fraction of light that is reflected from the entire surface and G is the geometric attenuation factor representing the ratio of light that is not occluded by the surface due to geometric occlusions. This microfacet model was the basis for many other works who offered variations on the calculation of the functions D, F and G [Maxwell et al., 1973][Cook and Torrance, 1981].

For empirical models, the main aim is to provide a simple formulation specifically designed to mimic a kind of reflection. Consequently, we get a fast computational model adjustable by parameters, but without considering the physics behind it. A widely used empirical model is Phong model [Phong, 1975]. It obeys neither energy conservation nor reciprocity, but its simplicity has made it one of the most used in Computer Graphics. In [Phong, 1975], the way a surface point reflects light is described as follows:

$$f_r(\mathbf{X}, \omega, \omega_o) = \frac{\mathbf{k_d}}{\pi} + \mathbf{k_s}(\mathbf{r} \cdot \omega_o)^\alpha \tag{2.28}$$

where the parameter $\alpha \in [0, \infty[$ characterizes the shape of the specular highlight (from dull to more glossy surface) and is often referred to as shininess. Within this model, $\mathbf{k_d}$, $\mathbf{k_s}$ and α are parameters which can be chosen in order to simulate different types of surface materials (Figure 2.18).

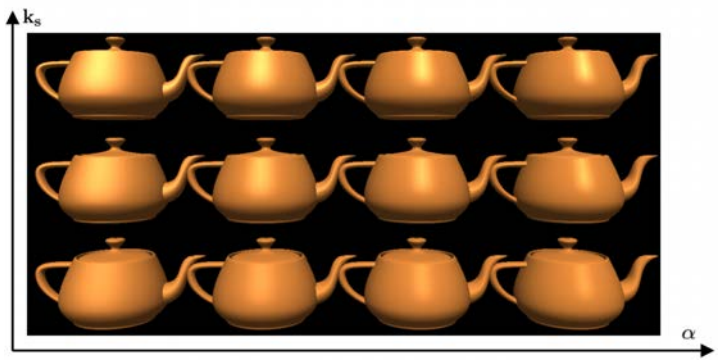


Figure 2.18 – Results of a rendered synthetic object using Phong reflection model with varying specular reflectance $\mathbf{k_s}$ and shininess α.

Rendering Approaches

Existing approaches to the production of synthetic images from 3D modeled scenes can be categorized into geometry-based (GB) and image-based (IB) techniques.

In geometry based approaches, the illumination of a scene is simulated by applying a shading model. These models can be local, such as Gouraud shading [Gouraud, 1971], which is a very simple technique that linearly interpolates color intensities calculated at the vertices of a rendered polygon across the interior of the polygon (Figure 2.20-a). Also, Phong [Phong, 1975] introduced a more accurate model that is able to simulate specular reflections. Specifically, in [Phong, 1975], the way a point p in the scene reflects light is described as a linear combination of three reflection components:

$$\mathbf{I}^p = \mathbf{I}_a^p + \mathbf{I}_d^p + \mathbf{I}_s^p \tag{2.29}$$

where $\mathbf{I}^p$ is the color of p and, $\mathbf{I}_a^p$, $\mathbf{I}_d^p$ and $\mathbf{I}_s^p$ are respectively ambient, diffuse and specular reflection components of point p (Figure 2.19). The ambient reflection is a simple way of modeling indirect reflections. When using only ambient lighting, all surfaces are equally illuminated. Diffuse reflection is the property that defines an ideal matte surface, also called Lambertian surface. Its apparent brightness to an observer is the same regardless of his angle of view (view-independent). Specular reflection is the mirror-like reflection of light from a surface. This component is view-dependent since such reflections are only observed when the viewpoint $\mathbf{v}$ and perfect reflection $\mathbf{r}$ vectors are roughly aligned.

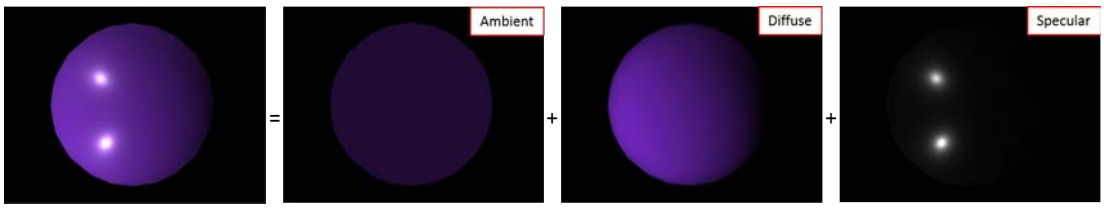


Figure 2.19 – The color of a point p described as a linear combination of ambient, diffuse and specular reflection components using Phong model [Phong, 1975]

Using [Phong, 1975], each reflection component in equation 2.29 is further described as follows:

$$\mathbf{I}^p = \mathbf{k}_d^p \mathbf{L}_a + \mathbf{k}_d^p \sum_{i=1}^{M} (\mathbf{n}^p \cdot \omega_i^p) \mathbf{L}_i \mathrm{O}_i^p + \mathbf{k}_s^p \sum_{i=1}^{M} (\mathbf{r}_i^p \cdot \mathbf{v}^p)^{\alpha_p} \mathbf{L}_i \mathrm{O}_i^p \tag{2.30}$$

where $\mathbf{L}_a$, $\mathbf{L}_i$ are respectively the color vectors of ambient and light source i. $\mathbf{k}_d^p$ and $\mathbf{k}_s^p$ are respectively the diffuse and specular reflectances of point p, $\mathbf{n}^p$ is its normal vector, $\mathbf{v}^p$ is its viewpoint vector, and α_p is its shininess parameter (roughness of the surface). $\mathbf{r}_i^p$ is the ideal reflection vector at point p with regard to light source i and ω_i^p is the direction of the light source i from point p. M is the number of light sources present in the scene. O_i^p is a binary visibility term that is equal to 1 if light i is visible from the 3D point p and equal to 0 if occluded.

Previously described models ([Gouraud, 1971] and [Phong, 1975]) are local in the sense that they do not model global illumination effects such as indirect reflections. To achieve more sophisticated renderings, there is a second class of illumination models that can be applied to polygonal scenes, which is referred to as global illumination. Unlike local approaches, these methods are able to simulate the inter-reflections between surfaces. For instance, diffuse inter-reflections (Figure 2.20-b) can be simulated by the radiosity method [Greenberg et al., 1986][Pattanaik and Bouatouch, 1994], and specular reflections can be handled by recursive ray-tracing techniques [Schmitt et al., 1988]. Nonetheless, these techniques are computationally too complex to be used for real time image synthesis on common MR devices.

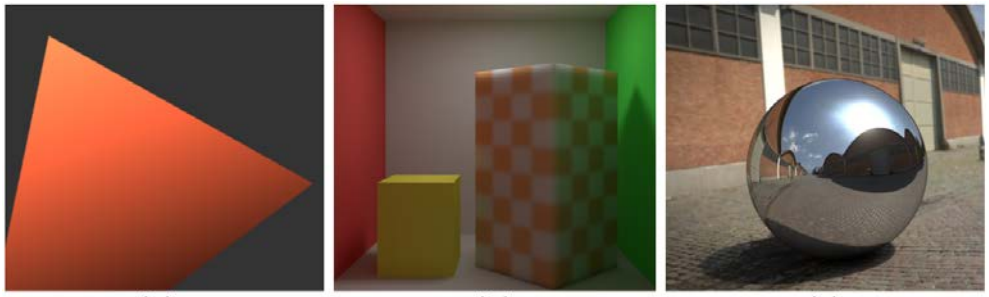


Figure 2.20 – (a) Rendered triangle using Gouraud shading [Gouraud, 1971] where the color intensities at the triangle's vertices are linearly interpolated. (b) Examples of the Cornell box scene rendered using radiosity techniques which allow diffuse inter-reflections (e.g., green color bleeding on the side of the patched box) - Figure of [Sheng et al., 2014]. (c) Example of a virtual sphere rendered using an image-based technique (rendering using an environment map).

Image based approaches (IB) involve capturing an omnidirectional representation of real-world light information as an image, typically using a specialized camera (Figure 2.20-c). This image is then projected onto a sphere or cube analogously to environment mapping. This map is finally used to simulate the lighting for the objects in the scene. This technique often produces results that are similar to those generated by raytracing, but is less computationally expensive since the radiance value of the reflection comes from calculating the angles of incidence and reflection, followed by a texture lookup,

rather than followed by tracing a ray against the scene geometry and computing the radiance of the ray.

2.3.3 Real Image Formation

Light that hits a point on a surface from a particular direction is at the heart of image formation. When a picture is taken, the shutter (piece of the camera which allows light through the lens) is open for a small amount of time. During that time, light originating from the visible scene (e.g., light sources, reflective surfaces) is focused through a lens and reaches the camera's image sensor, where the actual scene image is formed. The image sensor is partitioned into small pixels (photosites) where each pixel records light received over a small area. The recorded light corresponds to scene radiance which is first recovered as voltages and then converted ,within the image processing unit, into pixel values. In fact, scene radiance becomes pixel values through several linear and nonlinear transformations which can be modeled using the camera response function (CRF). This function is the aggregate mapping from sensor exposure (luminance) to pixel values. In most imaging systems, it usually follows an S-shaped curve (Figure 2.21), which tends to saturate both the highest and the lowest luminance values.

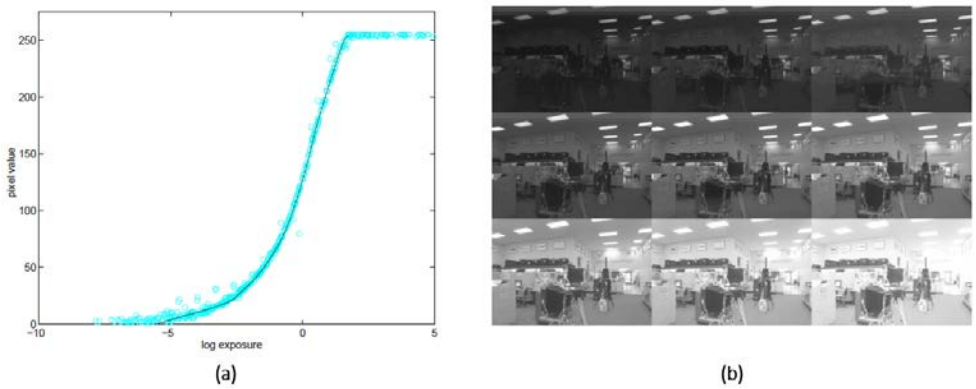


Figure 2.21 – (a) Example of a recovered camera response function (CRF) which maps exposure values to pixels values. (b) Nine photographs of an indoor scene acquired with varying shutter speeds in order to recover the camera response function of the sensor. Figures of [Debevec and Malik, 1997]

Knowing the camera response function is of interest for photometric registration algorithms. In fact, because of limited dynamic range in common cameras, one has to choose the range of radiance values which are of interest and determine the exposure time suitably. For instance, scenes with glossy materials and artificial light sources, often have extreme differences in radiance values that are impossible to capture without either under-exposing or saturating the sensor. Hence, one ends up either loosing details in shadowed regions or saturating specular reflections. However, when the goal is to estimate surface materials, having access to an accurate radiance information of such surfaces is an important step toward estimating scene reflectance propeties. Thus, by recovering the camera response function of the sensor, the exposure can be first manually or automatically adapted to the current dynamic range of the scene, and recorded

pixel values can be later converted to radiance maps which are independent of the current camera settings (shutter speed, aperture, ISO, etc.). Several approaches have been proposed to recover the CRF of a given sensor [Debevec and Malik, 1997][Grossberg and Nayar, 2003][Aimone and Mann, 2007], the core idea consists in a pre-calibration step where overlapped images with different exposure images are acquired (Figure 2.21).

2.4 Conclusion

In this chapter, we presented the main theoretical and practical components involved in mixed reality frameworks. Specifically, two main aspects need to be addressed, namely geometric registration and photometric registration.

The goal of geometric registration is to align in 3D both real and digital worlds. To achieve this goal, the geometry of the scene as well as the camera's model (intrinsic and extrinsic parameters) need to be, as much as possible, precisely recovered. Several existing solutions have been presented with regard to both tasks. An accurate registration results in a geometric compositing of real and digital worlds where occlusions and collisions can be correctly handled.

As far as photometric registration is concerned, the aim is to realistically blend real and digital worlds. To achieve this task, a trade off is made with regard to the choice of the reflection model used to model both real and digital scenes. On one hand, sophisticated reflection models are capable of delivering outstanding renderings of synthetic scenes. Nonetheless, since such models usually involve many parameters, they are not adequate when the estimation of illumination and reflectance is achieved using low quality images provided by an end-user sensor.

In this thesis, our goal is to propose photometric registration approaches using a common RGB-D camera. The 3D model and color images of the scene, acquired using a calibrated sensor (intrinsic parameters and camera response function), represent our input data to estimate both reflectance and illumination of real scene surfaces.

State-Of-the-Art of Photometric Registration

3

Contents

A mixed reality scenario is convincingly real when it is impossible to separate the virtual elements from the real elements in the resulting mixed environment. In order to achieve such realism, real and virtual worlds must be photometrically registered. This mainly requires the estimation of surface reflectance (e.g., diffuse, specular) and illumination characteristics (e.g., position, color).

Extensive work has been carried out within the photometric registration task. A comprehensive classification of these methods is proposed by Jacobs et al. [Jacobs and Loscos, 2006]. The existing approaches are grouped into three different classes: the first class corresponds to techniques where the 3D model of the real scene is unknown and only a single image is available. The second class comprises techniques where both the 3D model and a single image of the scene are known. The third class corresponds to approaches where the 3D model is known in addition to a set of images of the scene.

In this thesis, our goal is to achieve realistic mixed reality scenarios. As in most cases, MR is experienced through a camera's stream, we generally dispose of a sequence of color images. In the following, we present state-of-the-art methods, grouped with regard to the required input devices. Specifically, we present methods which respectively use an RGB camera (the 3D model is not necessary or reduced to basic geometry such as a plane), an RGB camera along with one or more light probes and, an RGB-D camera (the 3D model is either reconstructed using depth maps or a single depth map is sufficient for the processing).

3.1 Approaches using an RGB Camera

Early work considered the problem of separating diffuse and specular reflection components within a single color image using the dichromatic model [Shafer, 1992]. Within

this model, the color $\mathbf{I}^p$ of a pixel p is described as follows:

$$\mathbf{I}^p = \alpha^p \mathbf{I}_d^p + \beta^p \mathbf{I}_s^p \tag{3.1}$$

where $\mathbf{I}_d^p$, $\mathbf{I}_s^p$ are respectively the diffuse and specular reflection components. α^p and β^p are respectively factors (proportions) of these two reflections at pixel p. Within equation 3.1, provided that the illumination color is known (e.g., by imaging a white object surface), the proportions of diffuse and specular reflection components, α^p and β^p, can be easily computed by solving the dichromatic equation using least squares. For robustness reasons, Tan et al. [Tan and Ikeuchi, 2003] introduced the concept of Specular Free images (SF) to solve equation 3.1. In fact, by producing an image which is free from specularities, the pixels retaining only the diffuse reflection are considered separately from pixels exhibiting both reflections. Shen et al. [Shen and Cai, 2009] used similar separation mechanism and proposed a new SF image, referred to as Modified Specular Free (MSF), which proved to be more robust in presence of image noise. The diffuse and specular candidates are identified therefore according to the difference between the MSF and original images. Experimental results showed promising results with regard to separating both diffuse and specular reflections (Figure 3.1). Nonetheless, such methods deliver good results for special setups which do not satisfy MR requirements (e.g., a single isolated object, locally small-sized specular reflections). Furthermore, there is no estimation of the illumination within these approaches (e.g., number, position) as the main target application is specularity removal in input images.

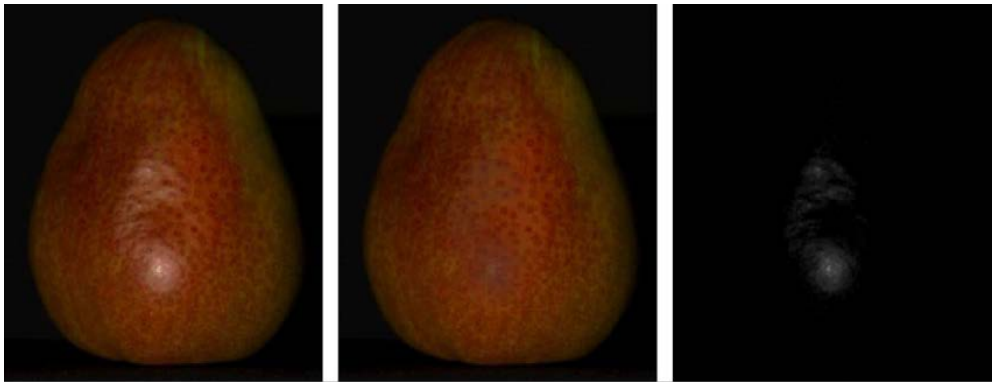


Figure 3.1 – Decomposition of a color image (a) into its diffuse reflection component (b) and its specular reflection component (c). Figures from [Shen and Cai, 2009]

Several works considered, instead of a single image, a set of color images of real scenes. Machita et al. [Mashita et al., 2013] presented an in-situ lighting and reflectance estimation method which uses images of the scene taken from multiple viewpoints. The estimation is achieved in a two-pass procedure: initial values of reflectance and illumination parameters are roughly recovered then, a non-linear optimization which considers the difference between real and synthesized images is carried. The convergence of such systems highly depends on the initial values. Consequently, by recovering light sources direction from high intensity image areas (e.g., saturated pixels), the initialization is prone to errors (e.g., bright and white surfaces can be confused with specular

reflections). For parameters whose initial values are difficult to estimate, values were assigned heuristically: the light distance is defined based on the height of the ceiling, while the components of the ambient, diffuse and specular reflections as well as the shininess coefficient are arbitrary chosen as the center of each range.

Jachnik et al. [Jachnik et al., 2012] presented an algorithm which is able to capture surface light-field from a single hand-held RGB camera by moving it around a specular planar object (e.g., shiny book). The captured surface light-field for each point p is then split into its diffuse and specular components (Figure 3.2-a). Moreover, the recovered specular reflection is used to estimate an environment map representing illumination in the scene (Figure 3.2-b). The core idea of retrieving the diffuse and specular reflection

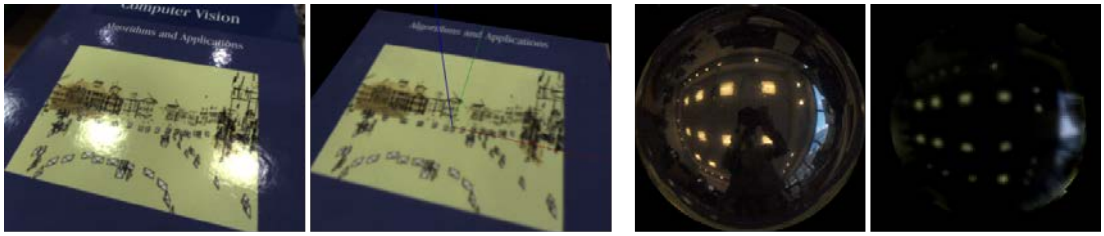


Figure 3.2 – (a) Input color image (left) and recovered diffuse reflection component (right). (b) Captured (left) and estimated (right) environment maps. Figures from [Jachnik et al., 2012].

components using color variations resulting from a moving camera shows robustness in comparison to recovering specular reflections as saturated regions. Convincing planar augmentations such as generating shadows for virtual objects and removing real specularities are achieved (Figure 3.3, video). Though this method provides good MR renderings, it considers only a planar and small surface. Furthermore, the illumination is recovered using an environment map which implies the assumption of distant light sources.

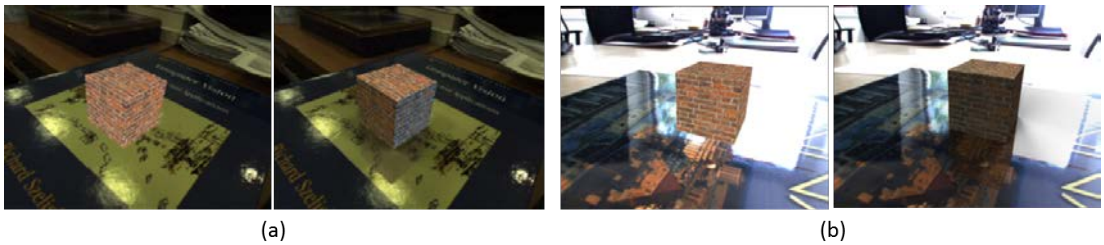


Figure 3.3 – Augmented scenes (a) and (b) without (left) and with (right) using recovered reflectance and illumination: the most important aspect is the oclusion of real specularities by the virtual cube. Figures from [Jachnik et al., 2012].

Recently, several data driven approaches have been proposed to estimate reflectance and illumination from a single image. Deep lambertian networks [Tang et al., 2012] apply deep belief networks to the joint estimation of a surface's reflectance and the direction of a single point light source. They rely on Gaussian Restricted Boltzmann Machines

to model the prior of surface albedos. In [Georgoulis et al., 2018], the Lambertian constraint is further relaxed and several specular materials under general illumination are handled. The approach uses a novel deep learning architecture which achieves sparse data interpolation and infers high dynamic range data from low dynamic inputs in order to recover accurate specular parameters.

3.2 Approaches using an RGB Camera and Light probes

A light probe is an omni-directional high-dynamic range image. Because light probes are 'supposed' to capture light sources intensities (radiance) from all directions, they are useful for providing measurements of the incident illumination. Consequently, they can be used to provide interesting and realistic lighting environments and backgrounds for rendered graphics.

One method of obtaining a light probe is to produce a high-dynamic range image of a chrome sphere (Figure 3.4), usually carefully positioned at the center of the target real scene. Early work was proposed by Debevec et al. [Debevec, 1998] where they photographed a chrome sphere under three exposure settings in order to recover its high-dynamic range image using [Debevec and Malik, 1997].

Figure 3.4 – Three photographs of a mirrored ball (chrome sphere) used to recover a high dynamic range of a light probe image. Figures from [Debevec, 1998].

In [Debevec, 1998], the light probe is used to render the scene which is partitioned into three components. The first is the distant scene, which is the visible part of the background environment, too distant to be perceptibly affected by the synthetic object. The second is the local scene, which is the part of the environment which will be significantly affected by the presence of the virtual objects. The third component is the synthetic objects. Using such scene partition along with a novel global illumination method referred to as *differential rendering*, the approach produces perceptually convincing results such as realistic shadows and inter-reflections (Figure 3.5).

Another method of obtaining a light probe consists in using a wide-angle lens camera (e.g., fish-eye lens). Knecht et al. [Knecht et al., 2012] proposed a method that reconstructs the surrounding environment using this type of light probe (Figure 3.6-a). The proposed approach uses a Kinect sensor to acquire scene's geometry and a fisheye

(a) (b) (c)

Figure 3.5 – Results of Debevec et al. [Debevec, 1998] where (a) is the input color image, (b) is a capture from the scene calibration step where the light probe (chrome sphere) is placed at the center of the scene and (c) is the achieved rendering of mixed reality. Figures from [Debevec, 1998].

camera to capture the incident illumination. Diffuse reflectance is recovered per cluster (and not per pixel) as the mean color of pixels belonging to the same cluster after a color-segmentation of the scene. Furthermore, specular reflectance is recovered as the maximum color intensity within detected highlight regions using [Ortiz and Torres, 2006]. The method runs in real-time and achieves convincing renderings such as inter-reflections between real and virtual objects (Figures 3.6-b). Nonetheless, it only handles scenes where each object holds a single color. Most importantly, specular reflections can be erroneously clustered when the highlight removal step does not succeed and subsequently, are recovered as the diffuse component within these regions (Figure 3.6-c).

(a) (b) (c)

Figure 3.6 – (a) Example of a camera with fish-eye lenses used in [Knecht et al., 2012] (Figure from this link). (b) Mixed reality scenario using the approach of [Knecht et al., 2012]: one can notice the yellow inter-reflections between the virtual statue and the real yellow bucket. (c) Critical scenario with regard to the method in (b) where the specularity is recovered as the difffuse reflection component. Figure from [Knecht et al., 2012].

Several works using one or more light probes have been proposed. Recently, Rohmer et al. [Rohmer et al., 2014] proposed a novel distributed illumination approach for MR where scene analysis is achieved using a stationary PC and MR rendering is experienced through a Surface Pro Tablet. The method uses four HDR cameras equipped with fish-eye lenses and placed in the scene, such that all regions are visible in at least one camera image. The proposed framework allows for an interactive illumination of virtual objects with a consistent appearance (e.g., realistic shadows, color bleeding)

under temporally varying real illumination conditions as shown in this video. The downside of this approach is mainly related to the complexity of the setup and, the manual and offline estimation of geometry and diffuse reflectance using common Digital Content Creation (DCC) tools.

3.3 Approaches using an RGB-D Camera

Generally speaking, proposed photometric registration approaches using an RGB-D camera handle more complex scenes geometry-wise. The fact that depth maps are provided in real-time within the RGB-D stream allows for more generic 3D reconstruction of real scenes. Also, such approaches usually take advantage of observed cues such as shading, shadows and specularities to estimate the reflectance and illumination of scene surfaces.

Specularity-based Methods

Early work using specular cues was proposed by Nishino et al. [Nishino et al., 2001] where they separate diffuse and specular reflection components using a sparse image set and a geometric model. Since the real scene (a single object) and the light sources are assumed to be fixed and only the camera is moving, only the viewing direction changes through the image sequence. This means that only the specular reflection component varies from image to image for each point on the object surface, while the diffuse reflection component is view-independent and constant. The curve in figure 3.7-b shows how the intensity value of a particular surface point varies while the camera moves around the object depicted in 3.7-a.

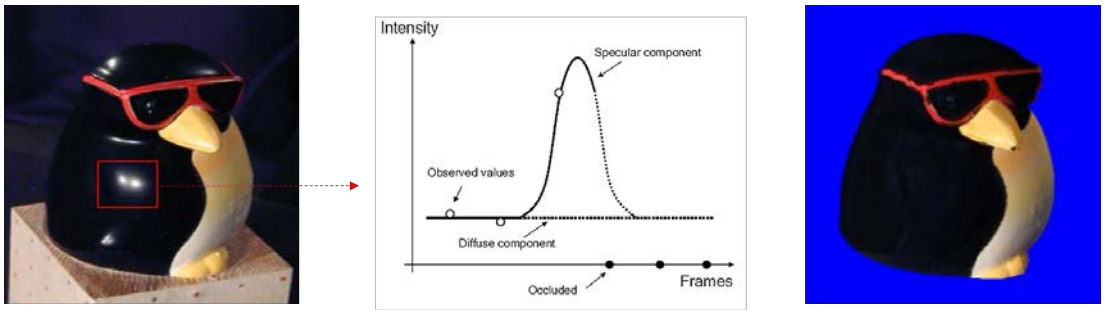


Figure 3.7 – (a) Input color image of a specular object from a sparse set of images. (b) Theoretical curve demonstrating the evolution of a point's color intensity, under static lighting, when the camera moves. (c) Estimated diffuse reflection component with regard to (a). Figures from [Nishino et al., 2001].

As the diffuse reflection is theoretically constant throughout the image sequence, it is recovered as the minimum of the color intensity curve (Figure 3.7-(b,c)). Furthermore, residual images, generated by subtracting the diffuse reflection component from each original input image, are considered as an initial estimation of the specular reflection component. Finally, using recovered specular components, illumination is estimated

as a hemisphere over the object and both reflectance and lighting are refined using a non-linear optimization which minimizes the difference between input and rendered images. Although the method assumes that light sources are distant from the object, it provides a dense representation of the reflectance properties (per pixel estimates) and illumination conditions of the object.

Plopski et al. [Plopski et al., 2014] proposed a method based on the analysis of specularities as well. In [Plopski et al., 2014], the input image is first decomposed into its diffuse and specular reflection components using [Shen and Cai, 2009]. Then, the highlight regions from the recovered specular component are used to estimate the direction of light sources in the scene. Finally, a refinement of reflectance and illumination properties is achieved by minimizing the error between input and rendered images. In this work, both scene analysis and mixed reality application run in real-time using the Kinect sensor.

Shadow-based Methods

Early work using shadows to recover illumination in the scene was proposed by Sato et al. [Sato et al., 1999][Sato et al., 2003]. The authors proposed a method that estimates the illumination distribution using cast shadows by an object of known geometry. The illumination distribution is first approximated by discrete sampling of a hemisphere where virtual point lights are equally positioned. Then by considering pixels luminance within shadowed regions and, under the assumption of known reflectance, the light sources intensities are recovered using least squares. The main drawback of this method consists in requiring extensive user intervention. In fact, in order to identiy the shadowed regions within the image, two captures of the scene are required: with and without occluding objects (Figure 3.8-(a,b)). Using a fine sampling of the hemisphere, the approach reconstructs convincing synthetic shadows (Figure 3.8-c).

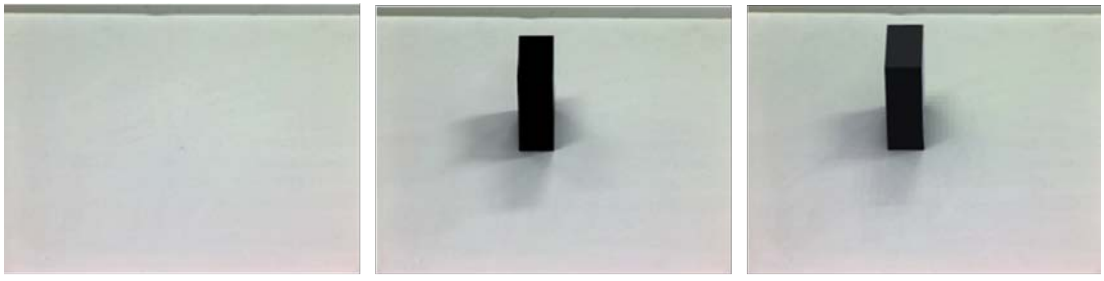


Figure 3.8 – Captures of the scene without (a) and with (b) occluding objects, used to detect shadows. (c) Reconstructed virtual shadows using a fine sampling of the illumination hemisphere. Figures from [Sato et al., 2003].

Extensive work has been carried in order to automatically detect shadows and integrate these cues within a photometric registration task. Arief et al. [Arief et al., 2012] estimate the position of a single strong light source in a controlled environment. By considering an object with simple and known geometry (e.g., cube), they analyze the

shadows which it casts on a single-color surface. The algorithm first detect the contours of the cast shadow. Then, using cornerness features, they recover the lines which relate shadow corners to their corresponding 3D points. The intersection of these lines corresponds to the 3D position of the light source (Figure 3.9).

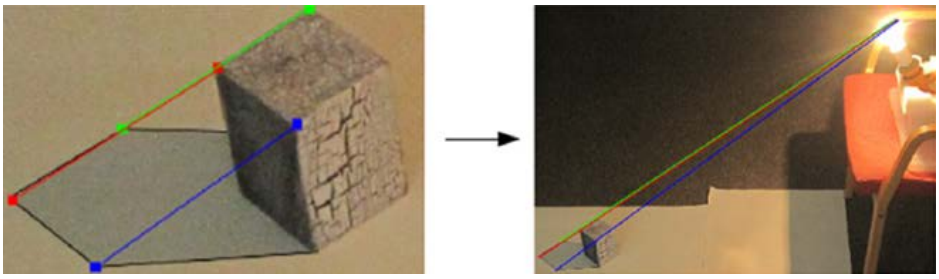


Figure 3.9 – Detection of the corners of the cast shadow and estimation of the 3D position of the light source in the scene. Figure from [Arief et al., 2012].

Panagopoulos et al. [Panagopoulos et al., 2009][Panagopoulos et al., 2011] proposed an approach which detects shadows in less constraining environments. In fact, their approach handles textured surfaces (Figure 3.10) and further recovers multiple light sources directions. The innovation is in the formulation of a Markov Random Field (MRF) model, where the energy to minimize in order to detect shadows contains several terms favoring consistency between neighboring pixels and, pixels corresponding to points with the same material property but subject to different illumination conditions. The proposed approach only handles Lambertian surfaces and takes 3 to 5 minutes to process a single image.

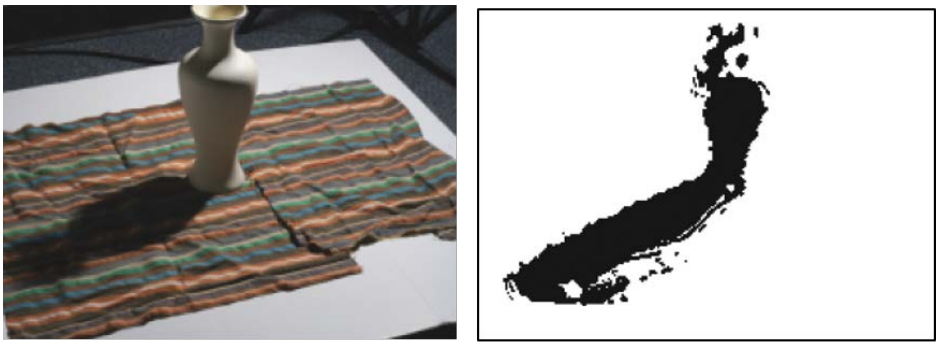


Figure 3.10 – (a) Input color image of the scene. (b) Detected shadows for the image in (a). Figures from [Panagopoulos et al., 2009].

Intrinsic Image Decomposition

Several methods are based on intrinsic image decomposition techniques. Their objective is to separate an image into its reflectance and illumination components. Within such approaches, the color $\mathbf{I}^p$ of a point p is described as follows:

$$\mathbf{I}^p = \mathbf{R}^p \mathbf{L}^p \tag{3.2}$$

where $\mathbf{R}^p$ and $\mathbf{L}^p$ are respectively the unknown reflectance and illumination components at pixel p. The term $\mathbf{R}$ contains the intrinsic color/texture of depicted surfaces while the illumination component encodes the incident illumination in the scene (e.g., shading, shadows and specular reflections). With regard to solving equation 3.2 for all scene points, Land and McCann proposed in 1971 the Retinex theory [Land and McCann, 1971] assuming that reflectance is characterized by sharp edges while illumination varies slowly.

Inspired by [Land and McCann, 1971], several approaches have been proposed since then to improve the intrinsic decomposition. For instance, often within intrinsic image decomposition methods, surface scenes are assumed to be Lambertian. Hence, the illumination term $\mathbf{L}$ is often referred to as shading because it only comprises shading and shadowing. Lee et al. [Lee et al., 2012] presented a technique to solve the intrinsic decomposition problem using an RGB-D sequence in presence of specular effects. This technique proposed two new types of constraints derived from available viewpoints and reconstructed geometry of the scene. While the former provides shading constraints based on surface orientation similarity, the latter imposes temporal constraints that enforce consistency within the intrinsic color of a surface point. This method uses both local and non-local, shading and temporal constraints, and yields interesting results (Figure 3.11).

Figure 3.11 – (a) Input color image of the scene. (b) and (c) are respectively recovered reflectance and illumination components with regard to the image in (a). Figures from [Lee et al., 2012].

Neverova et al. [Neverova et al., 2012] presented an approach which consists in decomposing an original input into its reflectance (diffuse and specular) and illumination components. First, they used [Shen and Cai, 2009] to decompose the input image into its diffuse and specular components. Then, within the recovered diffuse reflection component, they applied a Retinex based decomposition to estimate the shading and albedo images. The initially obtained components represent the inputs of an optimization process aiming at finding the 3D position of the light sources. Though this method gives good estimation results on low quality images, a weak point is the use of the dichromatic decomposition of Shen et al. [Shen and Cai, 2009] to separate the diffuse and specular components. In fact, this approach is not very robust as it can not handle large specular effect (in this case, the diffuse component is not well estimated).

Using Geometry as a Light Probe

Using a single RGB-D camera, Gruber et al. [Gruber et al., 2012][Gruber et al., 2014] developed an approach for real-time global illumination based on the reconstructed 3D model of the real scene. The method uses scene geometry as a light probe in combination with Spherical Harmonics (SH) to model the illumination in the scene. The proposed estimation procedure is further strengthened using visibility and normals information and, delivers convincing mixed reality results (Figure 3.12) using differential rendering [Debevec and Malik, 1997]. Such methods often assume Lambertian surfaces and SH-based recovered illumination exhibits a soft rendering of shadows (low-frequency) as shown within the grey virtual sphere in figure 3.12.

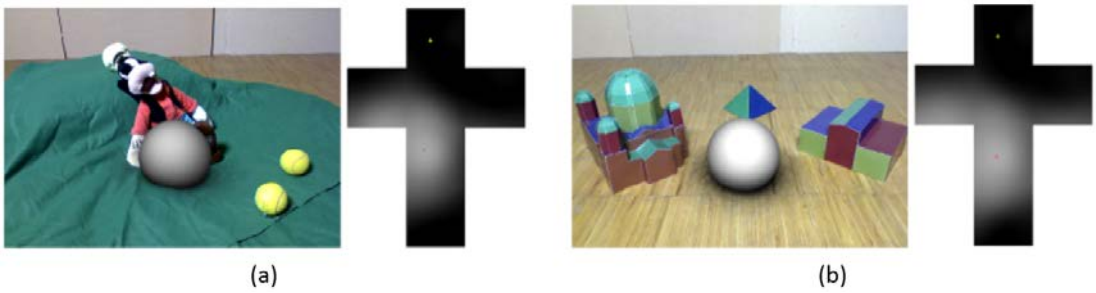


Figure 3.12 – Augmented real scenes (a) and (b) with a virtual grey sphere (left) and their recovered illumination (right). Figures from [Gruber et al., 2012].

Recently, Mandl et al. [Mandl et al., 2017] proposed learned light probes as an alternative to the probeless method of Gruber et al. [Gruber et al., 2012]. The core idea is, similarily to [Gruber et al., 2012], to use geometry as a learned light probe since it provides unobtrusive user experience in comparison with active light probes (e.g., chrome sphere, fish-eye lens). In this work, the incident lighting is estimated with a pre-trained convolutional neural network, which analyzes the appearance of a known object in the scene. Such methods require an object with a rich distribution of surface normals (ideally a sphere) in order to recover accurate illumination of the scene.

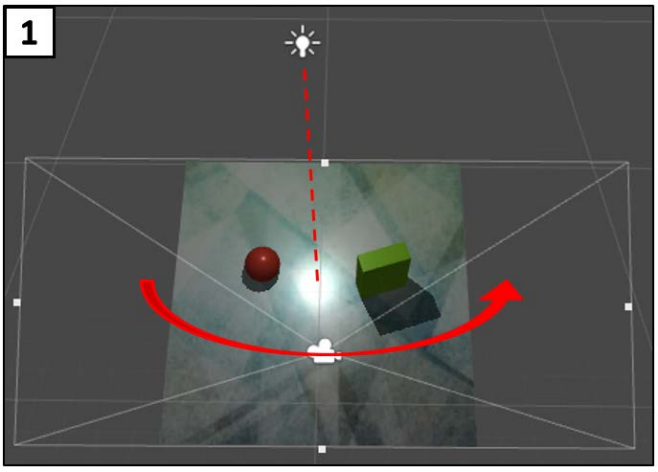
3.4 Conclusion

In this chapter, we presented several photometric registration approaches grouped with regard to their required input devices. Although the proposed approaches which use active light probes (e.g., chrome sphere, fish-eye lenses) deliver convincing results within MR scenarios (e.g., inter-reflections), the use of such additional devices is cumbersome for an end-user. Consequently, an RGB or RGB-D camera is usually more convenient for scene-analysis, especially that nowadays phones are commonly integrating depth sensing within their technology.

Furthermore, we favor RGB-D cameras since they provide online recovery of 3D information which allows to handle more complex real scene geometry-wise. Within this category of approaches, our goal is to estimate both diffuse and specular reflectance

properties for scene surfaces where texture/color can spatially vary. Also, we aim at estimating the 3D position (and not only the direction) and color of multiple light sources.

4 Photometric Registration using Specular Reflections



Contents

The word specular is derived from the latin word speculum, which means mirror. In the real world, many materials including leaves, plastic, and chrome exhibit specular reflections. When light hits such surfaces, it is reflected at a definite angle which obeys the Law of Reflection. Visually, the surface appears to be brighter and details/texture can be partially or completely obscured. Modeling such visual cues is evidently necessary to model the specular reflectance of real surfaces. Equally, it is key to probelessly recovering the illumination in the scene. In fact, many photometric registration approaches focus on deriving scene reflectance and illumination by detecting [Knecht et al., 2012] or predicting [Morgand et al., 2017] specularities within captured images. Nonetheless, existing solutions often address scenes with simple geometry (e.g., an isolated single object [Nishino et al., 2001], a planar surface [Morgand et al., 2018]) and/or simple textures (e.g., per-object constant color [Boom et al., 2013][Knecht et al., 2012]). Furthermore, the illumination is usually recovered as a distant lighting [Jachnik et al., 2012] or reduced to a single light source [Boom et al., 2013].

In this thesis, we are interested in relaxing these constraints. Specifically, our goal is to estimate both diffuse and specular reflectance properties of complex real scenes. The reflectance can spatially vary from one object to another and/or within the same object. Moreover, we aim at recovering the 3D position of existing light sources without using any light probe or external assistance. We only consider as input the RGB-D data provided by an RGB-D camera. To summarize, the main contributions of this chapter are:

- Estimation of specular reflections in complex real scenes using spatio-temporal data analysis.
- Recovery of spatially varying diffuse reflectance for 3D scene points.
- Estimation of the 3D position of light sources using only specular reflections observed in the scene.
- Photometry-based classification of all scene points (shadowed areas, diffuse and glossy surfaces).

In the remainder of this chapter, we first present the results of other approaches when considering our captured scenes. In particular, we highlight failure cases in accurately detecting specular reflections and estimating diffuse reflectance. Then, we present our approach to handle these challenges. Finally, results are discussed and our reflectance and illumination estimates are used to show realistic MR scenarios such as real specular reflections removed by the insertion of a virtual object and visually coherent virtual shadows.

4.1 Problem Description

Specular reflections refer to bright pixels with unsaturated colors which can occur in captured images. As previously stated, such reflections represent interesting cues from which we can derive the reflectance and illumination of a real scene. Detecting and localizing these cues is a great challenge in the computer vision field because, unlike

other materials, the appearance of a specular surface changes as function of its surface property, the lighting environment (e.g., light sources, other shiny surrounding surfaces) as well as the position of the observer. The specularity detection problem has been extensively studied, especially in the context of considering only a single image as input [Alsaleh et al., 2015][Ganz et al., 2012][Ortiz and Torres, 2006]. Within this configuration, different color spaces have been used. For instance, the Hue-Saturation-Value (HSV) color space is very informative in terms of detection of specular reflections. One of the characteristics of these reflections is that colors are unsaturated and the value (V) component is saturated. In [Ortiz and Torres, 2006], specular reflections are detected at pixels where the color has high value (V) but low saturation (S). In a first pass, the highlight detection result is written into a binary mask with a 'one' where the value and saturation criteria are met and a 'zero' otherwise. Then a morphological operation is performed in order to deal with noisy detections resulting from the thresholding step. In figure 4.1-(a,b), we can see that the specular reflections have been properly detected.

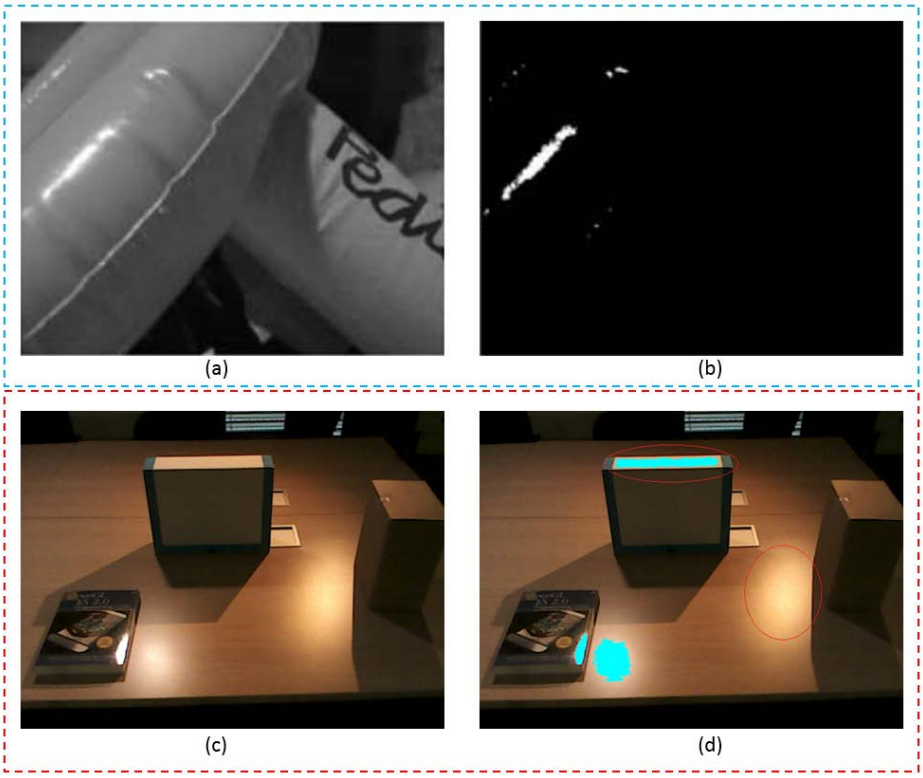


Figure 4.1 – (a) Input image. (b) Detected specular reflections for the input image (a). (a) and (b) are figures from [Ortiz and Torres, 2006]. (c) Our input captured image. (d) Results of [Ortiz and Torres, 2006] for our image (c): cyan pixels correspond to detected specular reflections. Red circles underline failure cases of this method such as weak specular effects and white surfaces.

Nonetheless, such algorithms often misinterpret a bright white-color surface with an actual specular reflection (Figure 4.1-(c,d): the top surface of the white/blue box). Additionally, such approaches can not handle weak specular reflections unless the thresholds are fine-tuned for each image independently (Figure 4.1-(c,d): only the specular

reflection located near the specular book is detected). Obviously, this process is not adequate for MR scenarios where the end-user does not have the knowledge of such parameters. Beside the problem of accurately detecting specular effects, another challenge rises when considering the reconstruction of the texture/details within these regions. In fact, various approaches consider the problem of recovering both diffuse and specular components within a single image [Shen and Cai, 2009][Tan and Ikeuchi, 2003][Mallick et al., 2006]. For instance, Shen et al. [Shen and Cai, 2009] proposed a method to separate diffuse and specular reflections in a color image based on the analysis of chromaticity at the pixel level (Figure 4.2-(a,b,c)). The proposed approach does not require any image segmentation or local interactions between neighboring pixels.

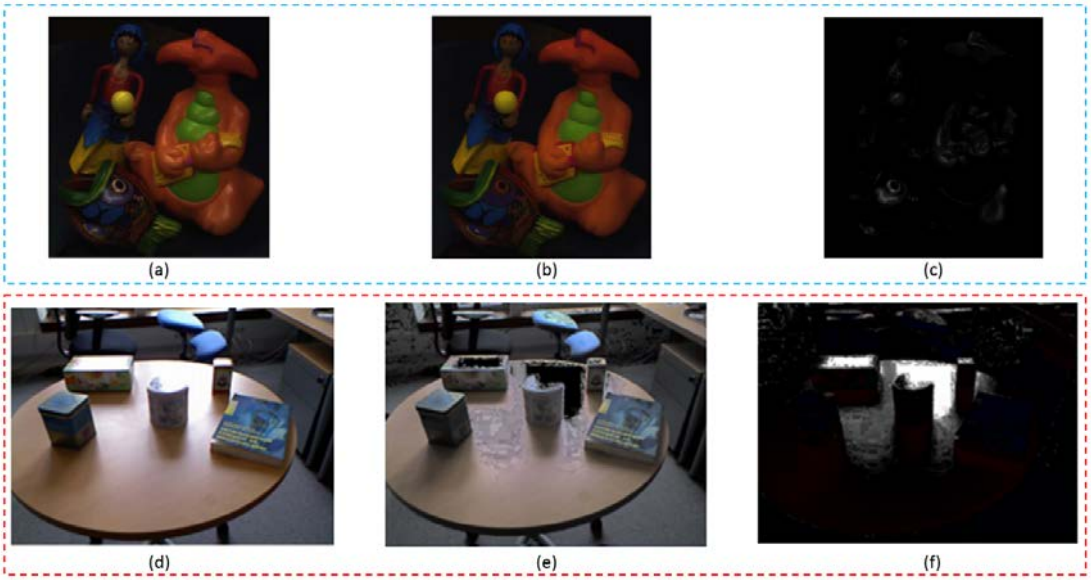


Figure 4.2 – (a) Input image. (b) Estimated diffuse component using [Shen and Cai, 2009]. (c) Estimated specular component using the same method. (a), (b) and (c) are figures from [Shen and Cai, 2009]. (d) Our captured input image. (e) and (f) are respectively the estimated diffuse and specular components using Shen et al.

Although Shen et al. [Shen and Cai, 2009] succeed in detecting the specular reflections within our captured images (Figure 4.2-(d,f)), the method fails at recovering the diffuse component (Figure 4.2-(d,e)). In fact, such algorithms usually rely on the closely surrounding pixels to recover the diffuse component. Hence, in presence of significantly large regions exhibiting specular reflections, the algorithm fails at recovering scene reflectance.

Our first contribution consists in robustly handling both specularities detection and diffuse reflectance reconstruction. In order to achieve this goal, we take advantage of the fact that within MR scenarios, the scene is observed through a sequence of images captured from various viewpoints and not only a single image. Let us consider a point p in the scene (e.g., a point on the shiny book) observed from two different viewpoints (Figure 4.3). The geometry and the lighting are assumed to be static.

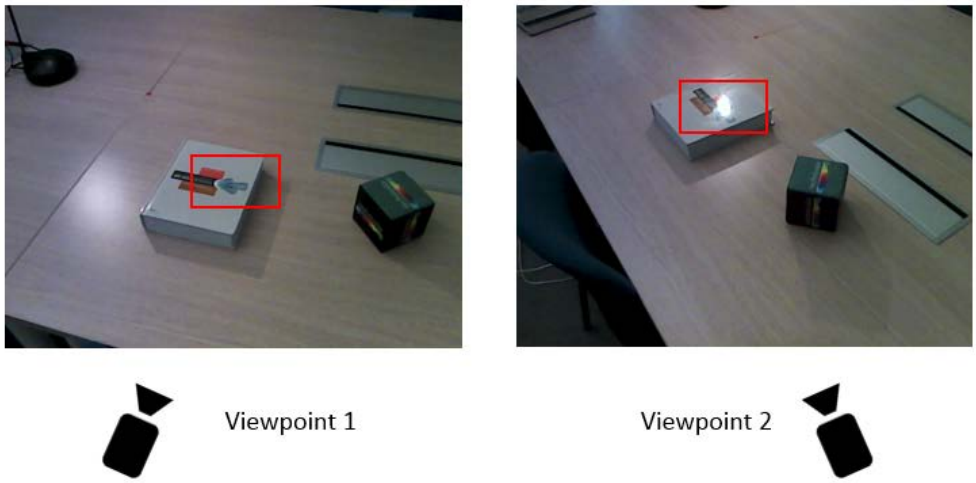


Figure 4.3 – The color intensity of a point p located on the shiny book's surface varies when seen from two different viewpoints due to its specular reflection component.

The color $\mathbf{I}^p$ of a point p in the scene can be described by Phong reflection model [Phong, 1975] as follows:

$$\mathbf{I}^p = \mathbf{I}_d^p + \mathbf{I}_s^p \tag{4.1}$$

where $\mathbf{I}_d^p$ and $\mathbf{I}_s^p$ are respectively the diffuse and specular components. Throughout the sequence, both reflection components can be described, at each frame t, using Phong model [Phong, 1975]:

$$\mathbf{I}^p(t) = \sum_{i=1}^{M} \mathbf{k}_d^p \mathbf{L}_i (\mathbf{n}^p \cdot \omega_i^p) + \sum_{i=1}^{M} \mathbf{k}_s^p \mathbf{L}_i (\mathbf{r}_i^p \cdot \mathbf{v}_t^p)^{\alpha_p} \tag{4.2}$$

Since we assume that the geometry and the light sources are fixed and only the camera is moving, only the viewing direction $\mathbf{v}_t^p$ changes through the image sequence. This means that only the specular reflection component varies from image to image for each point on the scene surface, while the diffuse reflection component is view-independent and remains the same. Consequently, by retrieving the evolution of scene points luminance through the sequence, we obtain a luminance profile (LP), shaped as a lobe, as depicted in figure 4.4. The lobe's peak value occurs when the viewpoint vector $\mathbf{v}^p$ and the perfect reflection vector $\mathbf{r}_i^p$ are (roughly) aligned.

Luminance profiles contain contain valuable information about the reflectance and illumination within the scene. In this contribution, we therefore consider such profiles to robustly detect specularities and reconstruct diffuse reflectance. Specifically, from recovered luminance profiles, we estimate diffuse and specular reflection properties as well as the 3D position of light sources.

4.2 Our Proposed Approach

Our method is an offline photometric analysis of 3D real scenes (Figure 4.5). We use a calibrated RGB-D sensor to capture the scene under various viewing angles in order

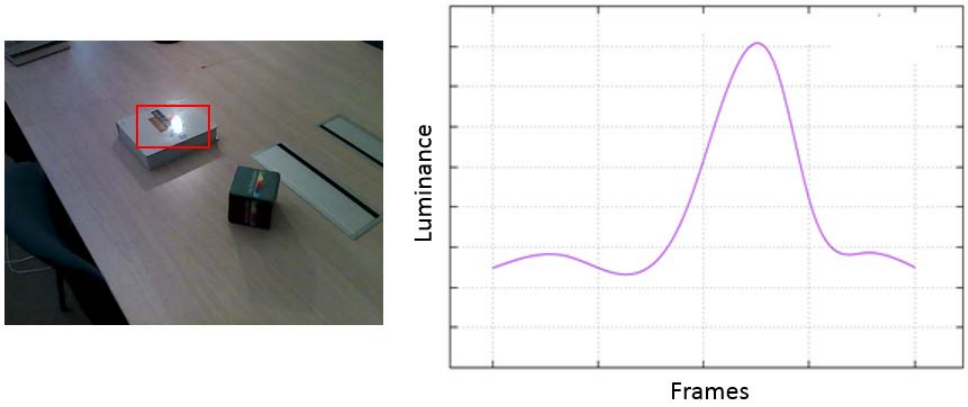


Figure 4.4 – Luminance profile (LP) of a point p located on the shiny book in the scene: the peak of the curve corresponds to an observed specularity when the angle between the viewpoint $\mathbf{v}^p$ and the perfect reflection $\mathbf{r}_i^p$ vectors is minimum.

to bring out specular and diffuse reflections with respect to each scene point. For this purpose, we track the 3D points of the scene along the sequence and retrieve their respective luminance profiles. Practically, using camera positions, we register the entire sequence with regard to a reference frame. We propose a simple and efficient statistical method to classify recovered luminance profiles and estimate both, view-independent (diffuse) reflection and view-dependent (specular) reflection components. Furthermore, we estimate the 3D position of light sources responsible for specular effects and provide a photometry-based classification of the scene's 3D points to distinguish between various surfaces (shadowed areas, Lambertian and/or specular surfaces).

Our algorithm deals with a variety of scenes where the texture spatially varies and several objects with different shapes can be present. We use our estimates to demonstrate specular occlusions between real and virtual objects (disappearance of real specularity due to a cast virtual shadow) as well as realistic virtual shadows. In terms of assumptions, scene geometry and illumination are supposed to be static. Only the sensor moves. The 3D model of the scene is represented by depth maps, no necessary reconstruction (mesh) is needed within our algorithm pipeline. The light sources are supposed to be sparse and are modeled as white point lights. Therefore, only one light source locally creates a specular effect at a time.

4.2.1 Sequence Registration

Using the Kinect v1 sensor

In a first phase, we used the Microsoft Kinect v1 sensor to capture the scene. This sensor provides both color and depth streams (Figure 4.6) at 30Hz with a 640×480 pixel resolution. Our first objective is to extract the evolution of luminance for all observed scene points. This requires tracking all pixels over the sequence. Practically, this is achieved by registering the images with regard to a reference one using camera poses and depth maps. The aim of image registration is to geometrically align two

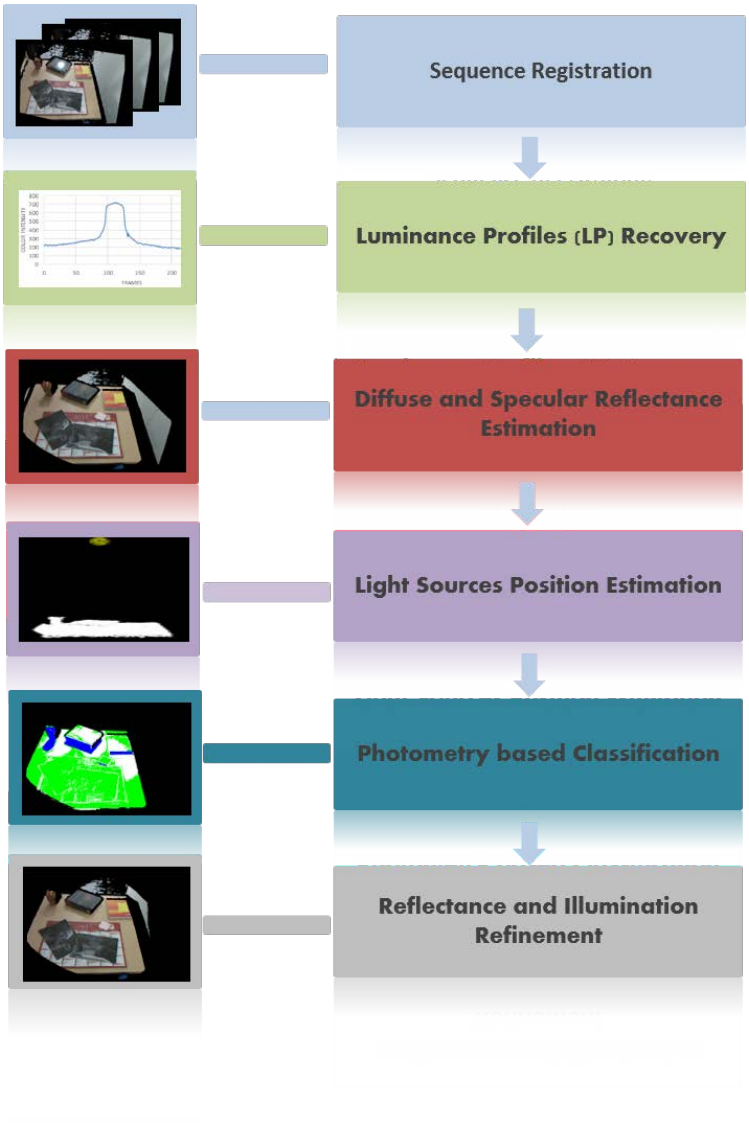


Figure 4.5 – Outline of our proposed photometric registration approach.

images acquired at different times and from different camera viewpoints. In order to obtain accurate results, we implemented a three-pass registration procedure: first, we use a robust 2D displacement estimator (marker based), then recover the camera's 3D position and finally use it to register all color images.

The first step consists in estimating the displacement of a 2D model, inserted in the scene, throughout the entire acquired sequence. The 2D model here, is an image with known dimensions. Hence, we use a template tracking method, also referred to as Differential Image Alignment [Baker and Matthews, 2004]. The basic principle is -assuming that two consecutive images are slightly different- to estimate the displacement $\mathbf{h}$ of the reference template $\mathbf{I}_r$ in a sequence of images. The problem can be written as follows:

$$\widehat{\mathbf{h}} = \arg\max_{\mathbf{h}} f(\mathbf{I}_r, \omega(\mathbf{I}_c, \mathbf{h})) \tag{4.3}$$

Figure 4.6 – RGB-D capture of the scene using the Kinect v1 sensor: the left image shows the input depth map. The right image shows the input color image.

where $\widehat{\mathbf{h}}$ is the current displacement which we aim to estimate in order to maximize the similarity function between the reference template $\mathbf{I}_r$ and the warped current template $\mathbf{I}_c$. Various similarity functions exist such as the Sum of Squared Difference (SSD) or the Normalized Crossed Correlation (NCC). Because of its accurate, robust and real-time demonstrated results, we chose the Mutual Information (MI) as a metric for image alignment [Dame and Marchand, 2012]. MI is the quantity of information shared between two signals. It uses the entropy of both, the reference and current warped templates. In [Dame and Marchand, 2012], they proposed a novel optimization process for the MI cost function and used an inverse compositional approach. Subsequently, equation 4.3 is then described as follows:

$$\widehat{\mathbf{h}} = \arg\max_{\mathbf{h}} \mathrm{MI}(\mathbf{I}_r(x), \mathbf{I}_c(\omega(x, \mathbf{h}))) \tag{4.4}$$

where x is a point that belongs to the region of interest which is here the reference template, and $\omega(x, \mathbf{h})$ is the location of x in the warped current region. In our case, the displacement parameters correspond to an homography transformation. The results (Figure 4.7) show accurate and robust displacement estimation in presence of important illumination changes.

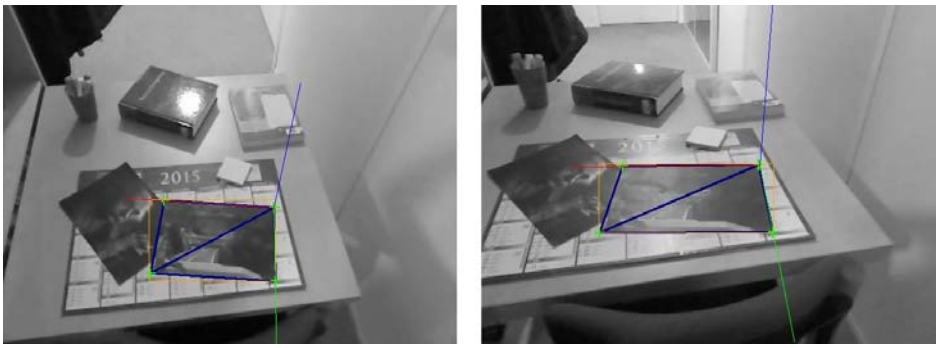


Figure 4.7 – MI-based tracking. The world frame related to our 2D model is projected using the recovered 3D camera pose from the MI-based tracking. The right image shows the robustness of our tracking with regard to strong illumination changes.

The second pass consists in recovering the 3D camera pose. As we have previously pointed out, the 2D tracked model is an image, where the coordinates of its four corners are known in a chosen reference frame. Therefore, using the estimated homography

parameters, we are able to update the corners 2D coordinates for every input image of the sequence:

$$\mathbf{x}_i^{\mathrm{k}} = {}^{\mathrm{k}}\mathbf{H}_0 \mathbf{x}_i^0 \tag{4.5}$$

where $\mathbf{x}_i^0$ is the ith corner's 2D coordinates in the reference image, $\mathbf{x}_i^{\mathrm{k}}$ is the updated ith corner in image k and $i = 1..4$. Using the intrinsic parameters of the sensor, we estimate the camera pose, within a non-linear Gauss-Newton minimization process:

$$(\widehat{{}^c\mathbf{R}_w}, {}^c\widehat{\mathbf{t}}_w) = \underset{({}^c\mathbf{R}_w, {}^c\mathbf{t}_w)}{\arg\min} \sum_{j=1}^{4} \mathrm{d}(\mathbf{x}_j, \mathbf{K}\Pi {}^c\mathbf{T}_w {}^w\mathbf{X}_j)^2 \tag{4.6}$$

where d(.) is the distance between the homogeneous 2D coordinates $\mathbf{x}_j$ in the camera frame and 3D coordinates ${}^w\mathbf{X}_j$ in the reference frame, $\mathbf{K}$ is the camera's intrinsic parameters matrix, Π is the perspective projection matrix and ${}^c\mathbf{T}_w$ is the transformation that fully defines the reference frame in the camera frame. The estimates are both, the rotation matrix $\widehat{{}^c\mathbf{R}_w}$ and the translation vector ${}^c\widehat{\mathbf{t}}_w$ which define the camera's position with regard to the reference frame.

The aim of the third pass is to align all the sequence images with a chosen reference image. Using the recovered 3D camera positions, we compute the transformation between each image and the reference one as follows:

$${}^{c_0}\mathbf{T}_{c_i} = {}^{c_0}\mathbf{T}_w {}^w\mathbf{T}_{c_i} = {}^{c_0}\mathbf{T}_w ({}^{c_i}\mathbf{T}_w)^{-1} \tag{4.7}$$

Subsequently, we use the 3D model of the scene at each frame, retrieved from the color and depth sequences, and re-project it on the reference image using the combined 3D transformation ${}^{c_0}\mathbf{T}_{c_i}$:

$$\mathbf{x}^{c_0} = \mathbf{K}\Pi {}^{c_0}\mathbf{T}_{c_i} {}^{c_i}\mathbf{X} \tag{4.8}$$

where $\mathbf{x}^{c_0}$ is a pixel, observed at image i, and re-projected using its model ${}^{c_i}\mathbf{X}$, on the reference image using ${}^{c_0}\mathbf{T}_{c_i}$. Figure 4.8 shows extracted images from the registered sequence: the color of some pixels was set to 0 because their corresponding 3D points were either geometrically occluded or simply out of the field of view in the current image.

Using the R200 sensor

By 2016, Intel released a new RGB-D sensor, the R200, which is more adequate for MR applications. In fact, the Kinect requires a separate power cable which is not convenient for mobile applications. On the other hand, the R200 only needs a USB 3.0 port which is very common in nowadays tablets. Additionally, the R200 sensor comes with an SDK which offers various features such as real-time 3D reconstruction, 3D camera pose estimation and provides a 3D mesh of the reconstructed scene model.

Using the R200, we developed a three times faster GPU-based implementation of equation 4.8 where the 3D model of the scene as well as the camera poses are provided by the sensor. Furthermore, the use of the 3D model instead of a single depth map delivers better results within the registered sequence (Figure 4.9) since the geometry is available for most scene surfaces.

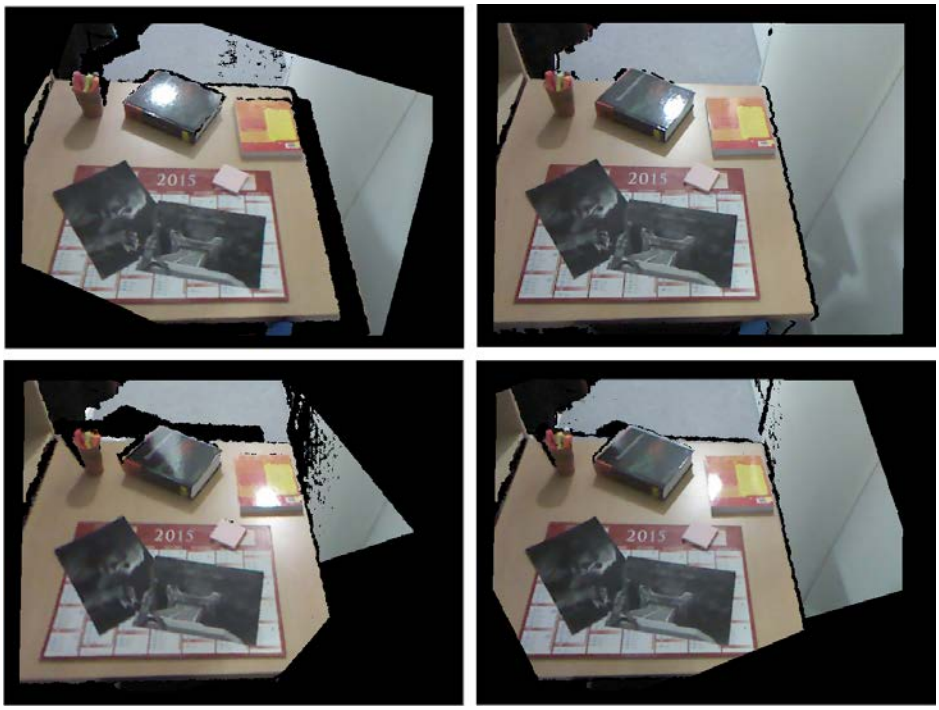


Figure 4.8 – Registered images using RGB-D images provided by the Kinect v1 sensor. The black pixels are 3D points which are either occluded or out of the field of view in the current image.

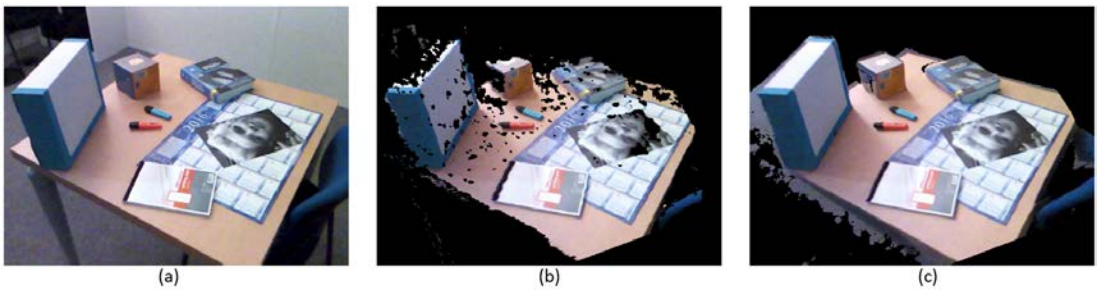


Figure 4.9 – (a) Reference color image. (b) Registered image from another viewpoint using a depth map. (c) Registered image from the same viewpoint as (b) using the 3D model provided by the R200.

4.2.2 Luminance Profiles (LP)

In the following, we will equally talk about the pixels of the reference image and the 3D scene points they correspond to. Considering all the sequence frames, we are able to track and estimate the luminance variations of a given pixel. These variations constitute a spatio-temporal Luminance Profile (LP). Practically, our set of registered images provides for each 3D point/pixel observed in the reference frame the evolution of its luminance along the video, described by $\mathbf{I}^p(t)$, where t is the index of the image in the sequence and p is a given pixel of fixed 2D coordinates along the sequence.

A particular curve is retrieved for each pixel p and corresponds to a linear combi-

nation of the three RGB color channels. Various and differently shaped profile curves can be obtained (Figure 4.10). For instance, if the values of the profile significantly vary, the 3D point is bound to belong to a specular surface. On the other hand, we have observed three main cases where a 3D point p holds an invariant profile: **(1)** p is Lambertian and never exhibits specular effects; **(2)** the point is not subject to specular effects because the observer's trajectory never meets the ideal specular reflection direction or simply because it is geometrically occluded by another scene object; **(3)** a specular effect exists all along the acquired sequence due for example to a large-surface light source.

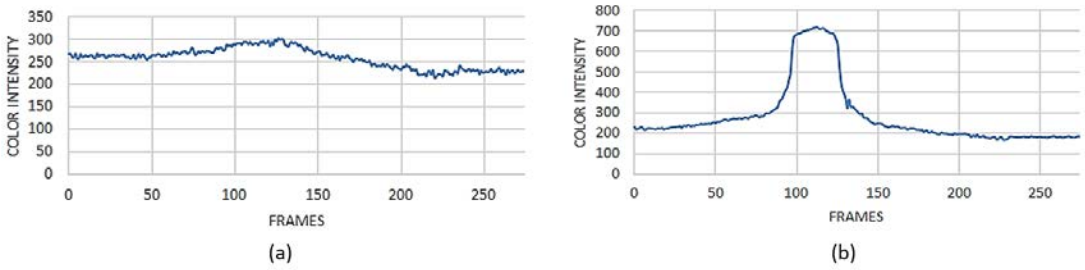


Figure 4.10 – Luminance Profiles: (a) corresponds to a point on the wood table which retains a roughly constant profile because the observer's trajectory never met the ideal specular reflection direction. (b) demonstrates strong luminance variations for a 3D point located on the specular black book in figure 4.8.

The amount of information brought by these luminance profiles is essential to our reflectance and illumination estimation process. Thus, in order to make the best use of it, we propose to separate these profiles into two main categories: Constant Luminane Profiles (CLP) and Variable Luminance Profiles (VLP). The former represents 3D points with weak luminance variations. The latter represents 3D points which hold both, diffuse and specular reflectance components. In order to correctly apply this LP-classification, we propose a simple and efficient statistical analysis.

First, we consider the profiles whose length is above half the total of frames (some points may be visible only on a part of the sequence). If a pixel's LP satisfies this first condition, we apply a gaussian filter in order to smooth the curve. For all the selected profiles, we compute the minimum m_p and maximum M_p luminance value, the mean MN_p and median MD_p values of all stored variations, and the standard deviation of the distribution SD_p. Based on our data analysis and observations, we propose a simple separation criterion to distinguish variable profiles from constant ones:

$$\begin{cases} |\mathrm{MD}_{\mathbf{p}} - \mathrm{MN}_{\mathbf{p}}| \geq \xi_1 \text{ or } |\mathrm{SD}_{\mathbf{p}}| \geq \xi_2 & , \mathrm{p} \in \mathrm{VLP} \\ \text{else} & , \mathrm{p} \in \mathrm{CLP} \end{cases} \tag{4.9}$$

Using these criteria, we quantify the amount of luminance variations and dispersion throughout the entire sequence. Thus, when a profile holds significant variations, the difference between the mean and median values is expected to be significant as well. Furthermore, a VLP corresponds to a curve that demonstrates dispersion with regard to the mean value.

As far as the VLP points are concerned, we are mainly interested in estimating their specular reflectance parameters which we will further use to recover the 3D position of the scene light sources. For CLP-points, the specular component is set to 0 provided that their constancy comes from their Lambertian property or constant geometry occlusion. In fact, as the light sources are sparse and the camera's trajectory is supposed to significantly cover the scene, we do not observe the case of a specular reflection all along the sequence. The results of our LP-based classification (Figure 4.11) matches our initial observations.

Figure 4.11 – (a): color image of the scene. (b): Luminance Profile based classification: the black pixels represent the discarded points (mostly occluded within the sequence), the grey ones correspond to constant luminance profiles and the white ones hold variable luminance profiles.

In figure 4.11, the 3D points which are classified as VLP (white color pixels), have actually exhibited specular effects during our scene capture. One can notice some noisy VLP classifications mainly due to registration errors. On the other hand, points which are constantly occluded by an object with regard to a light source or that simply were not observed in the ideal specular reflection direction appear in the CLP group (grey color pixels).

4.2.3 Diffuse and Specular Reflectance Estimation

Recovered luminance profiles retrain valuable information about the reflectance of the scene. Our goal is to take advantage of the profile's variations to robustly estimate both diffuse and specular reflection components. As previously mentioned, the color $\mathbf{I}^p$ of a point p, along the sequence, can be described by Phong reflection model [Phong, 1975] as follows:

$$\mathbf{I}^p(t) = \mathbf{I}^p_d + \mathbf{I}^p_s(t) \tag{4.10}$$

where $\mathbf{I}^p_d$ and $\mathbf{I}^p_s$ are respectively the diffuse and specular components at frame t. Since scene illumination and geometry are supposed to be static, the luminance variations present in the LP can only originate from the specular component which is view-dependent. In [Jachnik et al., 2012] and [Wood et al., 2000], the diffuse component

$\mathbf{I}_d^p$ is recovered as the median of the observed profile values. Although this value is robust in presence of shadows or bad registration errors, it gives an over-estimation of the view-independent component. Hence, we have chosen as in [Nishino et al., 2001], to estimate the diffuse component, for points belonging to either VLP or CLP group, as the minimum observed value in the LP since it should be closer to the correct diffuse component value (Figure 4.12).

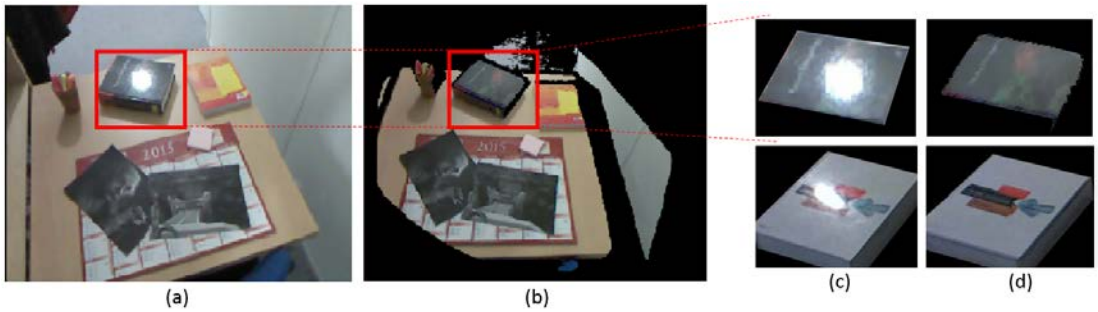


Figure 4.12 – (a) Captured reference image. (b) Recovered diffuse component. (c) Close captures of specular surfaces in considered indoor scenes. (d) Recovered diffuse component for captures in (c). Our approach correctly reconstructs the diffuse component within the regions with specular reflections.

The specular component is set to 0 for points with a constant LP since they do not exhibit any specular effects. On the other hand, for points holding a variable LP, it is retrieved for each frame t, as the difference between the diffuse component estimate $\mathbf{I}_d^p$ and the observed color intensity $\mathbf{I}^p(t)$ (Figure 4.13-(a,b)). Furthermore, the retrieved specular component can be described using Phong model as follows:

$$\mathbf{I}_s^p(t) = \sum_{i=1}^{M} \mathbf{k}_s^p \mathbf{L}_i (\mathbf{r}_i^p \cdot \mathbf{v}_t^p)^{\alpha_p} \tag{4.11}$$

where $\mathbf{k}_s^p$ is the specular reflectance of point p, $\mathbf{v}_t^p$ is its viewpoint vector at frame t, α_p is its shininess parameter, L_i is the intensity of the point light source i, $\mathbf{r}_i^p$ is its ideal reflection vector, and M is the number of light sources present in the scene (Figure 4.13-c). The unknown parameters are the combination of both, the specular reflectance and the intensity of the light source ($\mathbf{k}_s^p \mathrm{L}_i$), the reflection vector $\mathbf{r}_i^p$, the shininess coefficient α_p and the number of light sources in the scene i.

If the pixel's luminance at the peak of the LP is not saturated, the product ($\mathbf{k}_s^p \mathbf{L}_i$) can be recovered as the lobe's peak luminance value. We refer to the index of the frame where the maximum is reached as t_m. In fact, when the specular effect occurs, the viewpoint vector and the reflection vector are roughly aligned (their scalar product is roughly equal to 1). Subsequently, for every 3D point of the scene that holds a variable profile, we have:

$$\begin{cases} \mathbf{k}_s^p \mathbf{L}_i = \mathbf{I}_s^p(t_m) \\ \mathbf{r}_i^p = \mathbf{v}_{t_m}^p \end{cases} \tag{4.12}$$

Estimated specular components can contain errors due to registration misalignments. Hence, we apply a morphological erosion to the recovered specular component image

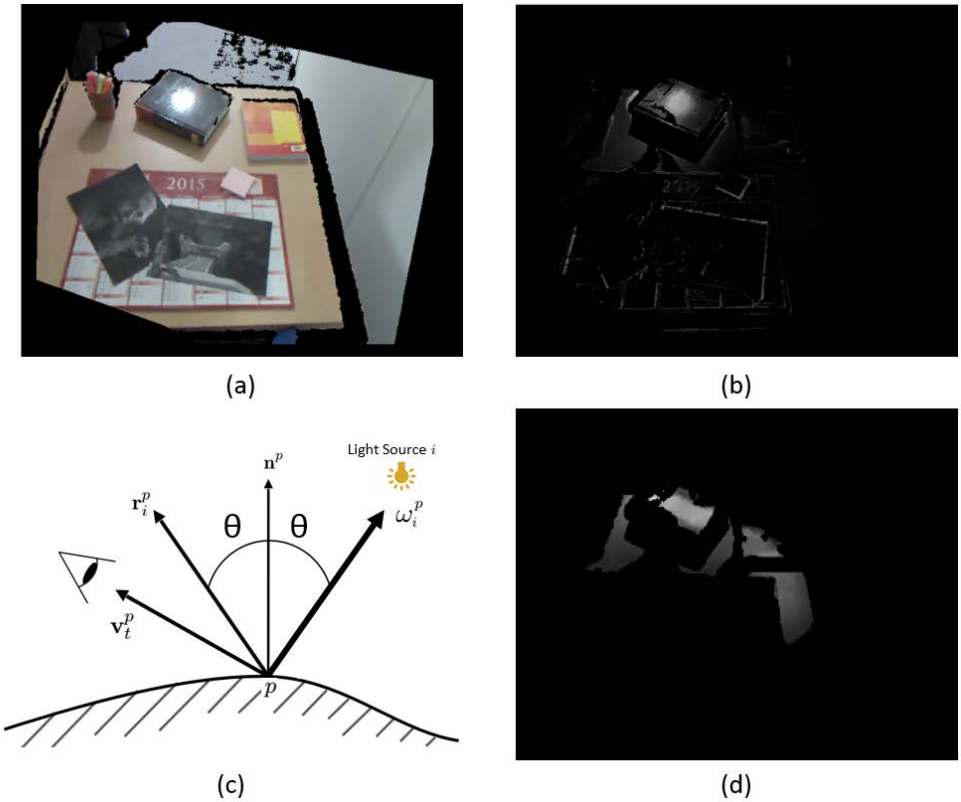


Figure 4.13 – (a) A registered captured image t of the scene. (b) Recovered specular component for the frame t. (c) Description of the 3D vectors used in the Phong reflection model. (d) Initial estimate of $(\mathbf{k}_s^p \mathbf{L}_i)$.

$\mathbf{I}_s^p(t_m)$ and recover the specular intensities $\mathbf{k}_s^p \mathbf{L}_i$ and reflection vectors $\mathbf{r}_i^p$ (Figure 4.13-(a,d)). Since we have no guarantee that the luminance lobe has reached its maximum possible value, we will be refining these estimates further in this work (Section 4.2.6.2).

4.2.4 Light Sources 3D Position Estimation

In this section, our goal is to estimate the 3D position of the light sources represented by point lights. To begin with, we compute for each VLP-point p the light direction vector ω_i^p using the estimated reflection vector $\mathbf{r}_i^p$:

$$\omega_i^p = 2.(\mathbf{r}_i^p \cdot \mathbf{n}^p) \cdot \mathbf{n}^p - \mathbf{r}_i^p \tag{4.13}$$

where $\mathbf{n}^p$ is the normal vector of point p estimated using [PCL, 2013]. Light rays originating from specular reflections that fall within the same specular area in the scene are clustered using Euclidean distance. Consequently, small clusters are processed as outliers since they generally result from registration errors or inaccurate normals. Then, a mean light direction vector is computed for each significant cluster (Figure 4.14).

Finally, the problem of finding the position of light sources is similar to computing the intersection points of a set of 3D lines using least squares.

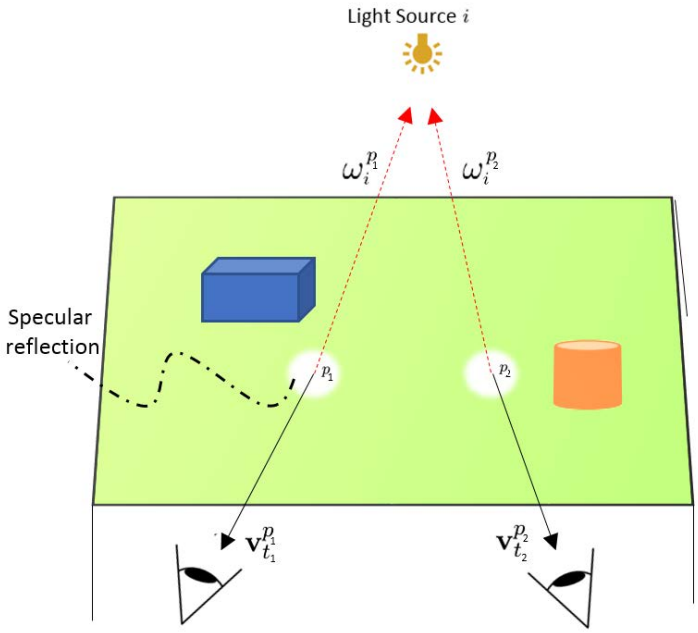


Figure 4.14 – Our approach to estimating the real light sources 3D position represented by point lights. The intersection of recovered light direction vectors ω_i^p corresponds to the 3D position of the point light source i.

4.2.5 Photometry-based Classification of the Scene

In section 4.2.2, we proposed a first classification within the luminance profiles which separates them into VLP and CLP groups. Within the CLP class, we can encounter points with different reflectance properties. Hence, they need to be processed differently in order to obtain accurate diffuse reflectance estimates. To illustrate, we will consider in the following the scenario of a single light source ($i = 1$), Phong model [Phong, 1975] is rewritten as:

$$\mathbf{I}^p(t) = \mathbf{I}_d^p + O_i^p \mathbf{k}_s^p \mathbf{L}_i (\mathbf{r}_i^p \cdot \mathbf{v}_t^p)^{\alpha_p} \tag{4.14}$$

where O_i^p is the occlusion parameter, equal to 1 if light source i is visible from point p and 0 otherwise. This parameter is used to take account of visibility within the scene with regard to the light source i. According to equation 4.14, a profile's constancy is observed when the specular component is equal to 0. This can be due to four possible reasons: the 3D point is not visible from the light source i because an object occludes it ($O_i^p = 0$). The 3D point belongs to a Lambertian surface ($\mathbf{k}_s^p = 0$). The trajectory of the camera with respect to point p and to the light source i is such that the reflection vector $\mathbf{r}_i^p$ and the viewpoint vector $\mathbf{v}_t^p$ have significantly different directions ($\mathbf{r}_i^p \cdot \mathbf{v}_t^p$ is always equal or close to 0). Last, the 3D point has a constant specular effect all along the sequence, however as our light sources are sparse and the camera trajectory is significantly varying, this case is not met.

Based on these observations, we categorize the points with a constant luminance profile into three main classes: Diffuse Points (DP) which are 3D points lit by all light sources and showing no specular reflections ($\mathbf{k}_s^p = 0$). Occluded Points (OP) are 3D points

which are constantly occluded with respect to the estimated light source i ($O_i^p = 0$). Non-Occluded Points (NOP) such that ($\mathbf{r}_i^p \cdot \mathbf{v}_t^p$) is equal or close to 0 all along the sequence. In the following, our objective is to fully classify CLP-points with regard to the previously defined subgroups (DP, OP and NOP). The results of our 3D classification are shown in (Figure 4.15).

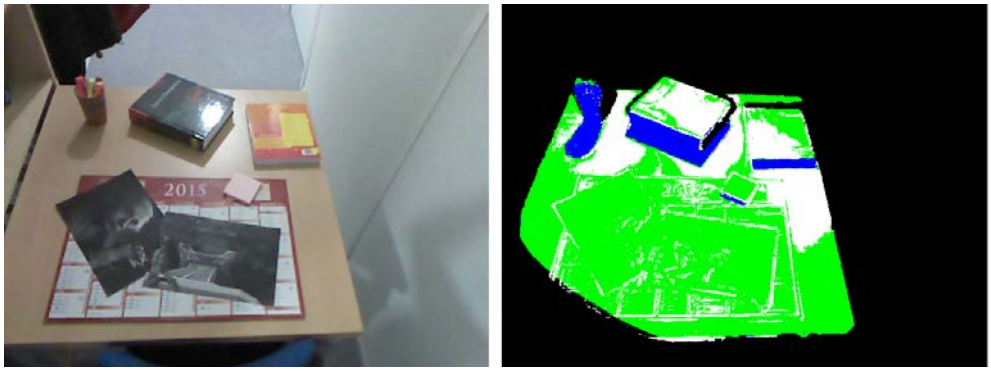


Figure 4.15 – (a) Reference image of the scene. (b) Photometry-based classification results: white color pixels corresponds to 3D points with variable profiles, green color pixels are 3D points which belong to the NOP subgroup (the reflection vector $\mathbf{r}_i^p$ and the viewpoint vector $\mathbf{v}_t^p$ have significantly different directions) and blue color pixels are occluded points with regard to light sources (OP).

Our classification is a two-pass procedure. The first pass consists in computing a visibility map with regard to the identified point lights. We perform the *shadow mapping* technique, a standard computer graphics algorithm. We refer to the shadow map as O_i^p where:

$$\begin{cases} O_i^p = 1, & \text{p is a visible point} \\ O_i^p = 0, & \text{p is an occluded point} \end{cases} \tag{4.15}$$

If point p is detected as an occluded point with regard to light source i, it is then classified as a OP point. Otherwise, it can belong to either NOP or DP subgroups. The second pass consists in separating these two sub-classes by detecting 3D points which might be subject to a specular effect but still conserve a constant profile, e.g. diffuse surfaces. To begin with, using the estimated reflection vector $\mathbf{r}_i^p$ and the viewpoint vector $\mathbf{v}_t^p$, we retrieve a novel profile $\epsilon_i^p(t)$ such as:

$$\epsilon_i^p(t) = \mathbf{r}_i^p \cdot \mathbf{v}_t^p \tag{4.16}$$

We use a method similar to the one described in section 4.2.2, with different thresholds, to distinguish variable and constant profiles. If $\epsilon_i^p(t)$ is significantly variable, then we conclude that $\mathbf{k}_s^p = 0$ and that point p belongs then to the DP subgroup. If $\epsilon_i^p(t)$ is a constant intensity profile, the point belongs to the NOP subgroup.

4.2.6 Reflectance and Illumination Refinement

4.2.6.1 Diffuse Component Refinement

In section 4.2.3, we have recovered the diffuse component of all observed points within the reference frame. The accuracy of this component estimate is of paramount importance when considering MR and relighting scenarios. Within these scenarios, the diffuse reflectance $\mathbf{k}_d^p$ represents the intrinsic color (texture) of the surface which must be independent from the illumination present when the scene was initially captured. However, in the previous section 4.2.5, we encountered several points which are occluded with regard to the current light sources in the scene (OP). Hence, the previously recovered diffuse component does not include the contribution of occluded light sources for points belonging to the OP group. Our objective is to correctly estimate the diffuse component for such points. The proposed approach is based on the comparison between points with the same diffuse reflectance but subject to different lighting conditions: visible (points belonging to NOP and DP) and occluded (points belonging to OP) points. Using [Phong, 1975], the color of these points differs as follows:

$$\begin{cases} \mathbf{I}^p = \mathbf{I}_d^p = \mathbf{I}_{d,O}^p + \mathbf{I}_{d,V}^p & \text{, if p} \in \text{(NOP or DP)} \\ \mathbf{I}^p = \mathbf{I}_d^p = \mathbf{I}_{d,O}^p & \text{, if p} \in \text{OP} \end{cases} \tag{4.17}$$

where $\mathbf{I}_{d,V}^p$ corresponds to the diffuse component due to the light sources that are visible from NOP and DP points only, and $\mathbf{I}_{d,O}^p$ corresponds to the diffuse component due to the light sources that are visible from all points. To begin with, we suppose -for now- that we are able to identify two points p_1 and p_2 with the same unknown diffuse parameter ($\mathbf{k}_d^p = \mathbf{k}_d^{p_1} = \mathbf{k}_d^{p_2}$). Point p_1 is supposed to belong to the visible subgroups (NOP,DP) whereas p_2 belongs to the OP subgroup. Since p_1 and p_2 are assumed to have the same diffuse parameter, we can write:

$$\begin{cases} \mathbf{I}_d^{p_1} = \mathbf{k}_d^p l_O^p + \mathbf{k}_d^p l_V^{p_1} \\ \mathbf{I}_d^{p_2} = \mathbf{k}_d^p l_O^p \end{cases} \tag{4.18}$$

where $l_V^{p_1}$ refers to the intensity of the light sources occluded from OP points and l_O^p refers to the intensity of the light sources visible from both types of points. Hence, both $\mathbf{k}_d^p$ and l_O^p can be estimated for the selected points:

$$\begin{cases} \mathbf{k}_d^p = \frac{\mathbf{I}_d^{p_1} - \mathbf{I}_d^{p_2}}{l_V^{p_1}} \\ l_O^p = \frac{\mathbf{I}_d^{p_2}}{\mathbf{k}_d^p} \end{cases} \tag{4.19}$$

Except close to shadow edges, lighting is locally constant and there is interest to consider a group of points as far as they are identified as having the same diffuse parameter $\mathbf{k}_d^p$:

$$\widehat{\mathbf{k}}_d^p = \frac{\overline{\mathbf{I}}_{d,V}^p - \overline{\mathbf{I}}_{d,O}^p}{\overline{l}_V^p} \tag{4.20}$$

where $\overline{\mathbf{I}}_{d,V}^p$ and $\overline{l}_V^p$ correspond respectively to average diffuse component and visible light sources intensities computed over all points belonging to the visible subgroups,

and $\overline{\mathbf{I}}^p_{d,O}$ is the average diffuse component intensity computed over all points belonging to the OP group (e.g. constantly occluded points). l^p_O is then estimated as:

$$\widehat{l^p_O} = \frac{\sum_p \overline{\mathbf{I}}^p_{d,OP}}{\sum_p \widehat{\mathbf{k}}^p_d} \tag{4.21}$$

Using the estimated $\widehat{\mathbf{k}}^p_d$, we are able to recover the contribution of the occluded light sources with regard to OP points (Figure 4.16).

Figure 4.16 – Refined diffuse component. The left image is the diffuse component estimate. The right image demonstrates the results of our diffuse recovery for occluded points. Because the area light source was approximated by a point light source, we can observe that the edges of the soft shadows were not correctly recovered.

The main challenge in estimating the parameter $\mathbf{k}^p_d$ remains in identifying visible and occluded surfaces corresponding to points with the same unknown diffuse reflectance. To achieve this goal, we propose to group points with regard to several strong similarity metrics: chromaticity values, color intensities, normal vectors and 3D locations. Chromaticity values are computed using the Modified Specular Free (MSF) image that is more robust than Specular Free images [Shen and Cai, 2009].

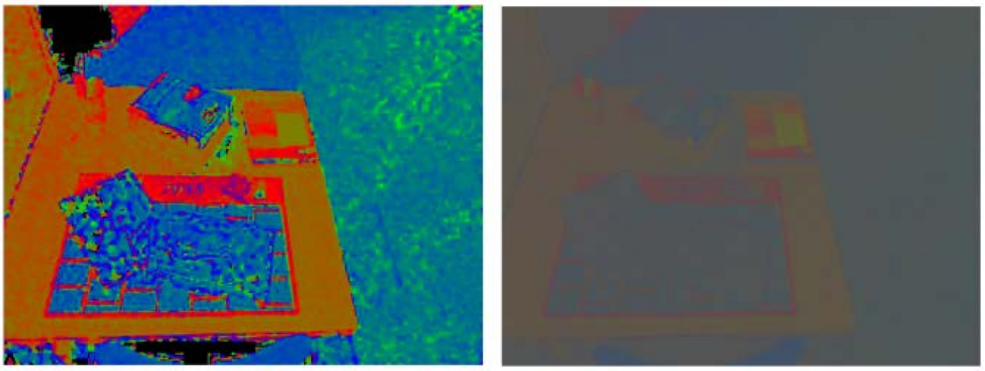


Figure 4.17 – Chromaticity images. The left image represents chromaticity values computed using a Specular Free (SF) image whereas the right image shows chromaticity values using a Modified Specular Free (MSF) image.

In order to compute the average diffuse values $\overline{\mathbf{I}}^p_{d,V}$ and $\overline{\mathbf{I}}^p_{d,O}$, we propose a feature-weighted filter defined as follows:

$$\overline{\mathbf{I}}^p_d = \frac{\sum_q (\omega_{p,q} \mathbf{I}^q_d)}{\sum_q \omega_{p,q}} \tag{4.22}$$

$\overline{\mathbf{I}}^p_{d,V}$ and $\overline{\mathbf{I}}^p_{d,O}$ use a set of points q located respectively in visible and occluded areas. These areas are known thanks to our photometry-based classification results. The weights $\omega_{p,q}$ consider all the previously mentioned similarity features:

$$\omega_{p,q} = \exp^{-\sum_f cost(f)} \tag{4.23}$$

where index f refers to a feature and $cost(f)$ refers to the norm of dissimilarity between features attached to points p and q.

4.2.6.2 Specular Component and Light Sources Position Refinement

Specular reflectance parameters have been previously recovered only for 3D points roughly viewed along the observed reflection direction. Our goal is to estimate dense reflectance maps. To achieve this, we initially assume uniform specular reflectance for each 3D object in the scene. A first step consists then in clustering the 3D mesh of the scene using Euclidean distance between vertices and normals smoothness constraint [PCL, 2013]. Then, provided that each cluster contains at least one 3D point with its specular reflectance estimate $\mathbf{k}_s \mathrm{L}_i$, we spread its value to all the cluster points.

We now address the possibility that an object/cluster may not exhibit a unique specular reflectance. First, we render specular reflections using previously recovered $\mathbf{k}_s \mathrm{L}_i$ values and light sources position. Rendered specular maps are correlated with observed specular maps. If correlation fails, we proceeed to a color-based segmentation using the k-means algorithm and set final $\mathbf{k}_s \mathrm{L}_i$ values to each color segment as follows: observed points in the direction of light sources that do not exhibit specular effects are considered to be Lambertian ($\mathbf{k}_s = 0$), points that are left keep the recovered $\mathbf{k}_s$ value.

Finally, we estimate the shininess parameter α_p, and refine $\mathbf{k}_s \mathrm{L}_i$ alternatively with the refinement of light sources position. The optimization is achieved using the Levenberg Marquardt algorithm with the following cost functions:

$$\begin{cases} F_j = \sum_i [\mathbf{I}^i - (\mathbf{I}^i_d + \sum_m \mathbf{k}_s \mathbf{L}_m (\mathbf{r}^i_m \cdot \mathbf{v}^i_t)^{\alpha_i})]^2 \\ G = \sum_u [\mathbf{I}^u - (\mathbf{I}^u_d + \sum_m \mathbf{k}_s \mathbf{L}_m (\mathbf{r}^u_m \cdot \mathbf{v}^u_t)^{\alpha_u})]^2 \end{cases} \tag{4.24}$$

where i and m respectively iterate over pixels that belong to cluster j and over recovered light sources. u iterates over all the reference image pixels. I^i and $\mathbf{I}^u$ correspond to observed pixels color. In F_j, the diffuse component $\mathbf{I}^i_d$ is fixed and only $\mathbf{k}_s \mathbf{L}_m$ and α_i can be varied by the solver. In G, all parameters are fixed and only the specular reflection vector $\mathbf{r}^u_m$ is updated.

4.3 Experimental Results

We photometrically calibrate our sensor as in [Robertson et al., 1999] by taking three images of a color checker with patches of known reflectance at three different shutter speeds. Hence, a calibrated camera with fixed aperture, shutter speed and color gain browses the scene. The proposed approach takes an average time of 2.4 minutes to process a sequence of 400 images.

Reflectance Evaluation

Our approach was tested on various indoor real scenes. In figure 4.18, we present the photometric registration results for three different scenes (S1, S2 and S3). Our algorithm succeeds in correctly recovering the diffuse component in regions initially exhibiting specular effects (Figure 4.18-b). Furthermore, our approach recovers accurate specular maps (Figure 4.18-d). In fact, we are able to infer the difference within the specular reflectance in various indoor real surfaces (e.g., in the second column of figure 4.18-d, the shiny books retrain a higher specular reflectance (brighter intensity) in comparison with the wood table). In figure 4.19, we use recovered reflectance and illumination to virtually relight the specular book. One can notice that the rendered specularity is well estimated as it is visually coherent with the real one.

Illumination Evaluation

Using a fish-eye lens, we capture the environment map and qualitatively compare it to the recovered lighting distribution (Figure 4.20) for three scenes (S1, S2 and S3 with respectively one, two and three light sources). Our algorithm estimates correct illumination distribution as it recovers the correct number of light sources responsible for specular reflections. Furthermore, we use salient control points and evaluate the accuracy of recovered positions. Table 4.1 shows a comparison between measured (Measured D) and recovered distances to light sources (Estimated D). Our algorithm is tested on various indoor scenes under various lighting (e.g single and multiple spot lights and/or led lights) and recovers light sources positions with an average error of 16cm for a mean distance of 1.95m to the light source.

Scenes	Measured D (cm)			Estimated D (cm)		
S1	72.2	-	-	86.1	-	-
S2	94.7	105.6	-	88.9	127.1	-
S3	314.7	293.6	306	297.1	261.6	283.2

Table 4.1 – Comparison between measured and estimated distances to light sources for scenes (S1, S2 and S3) under different lighting conditions.

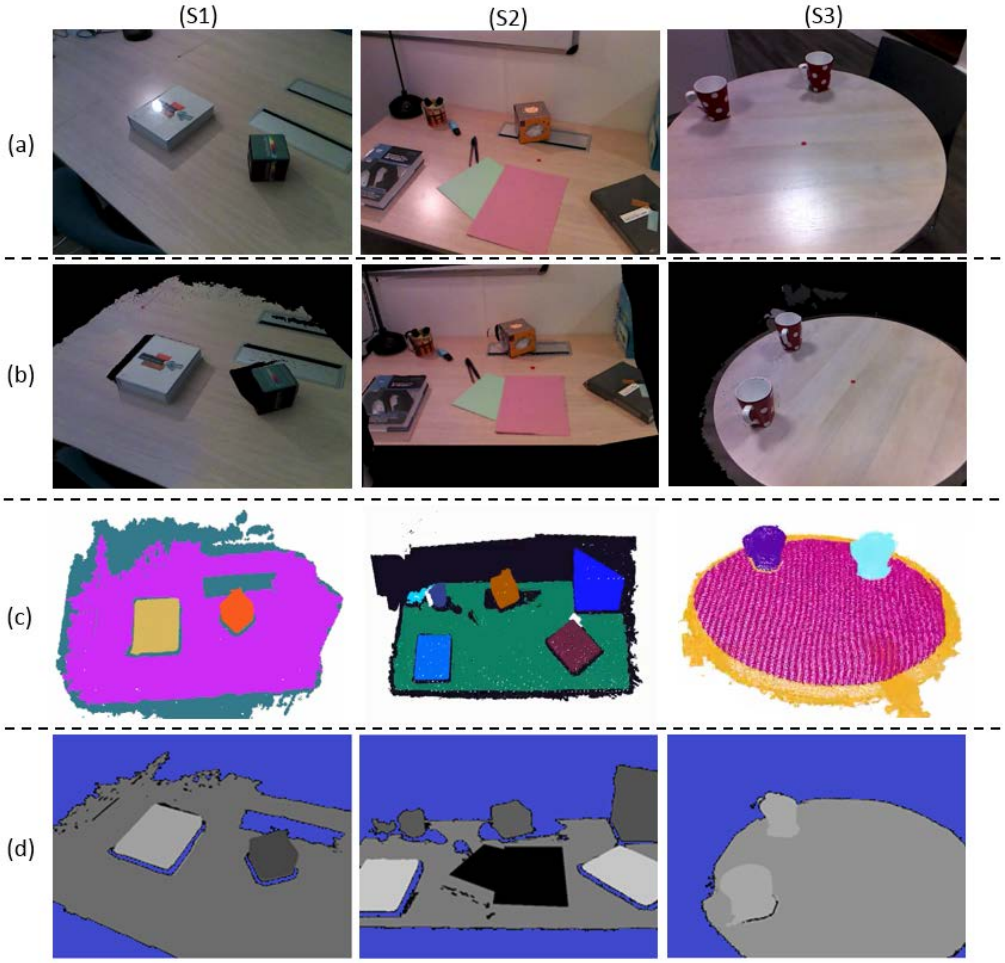


Figure 4.18 – (a) Camera views of three indoor scenes (S1, S2 and S3). (b) Recovered diffuse maps for surfaces with various textures and reflective properties. (c) 3D mesh clustering. (d) Estimated specular reflectance parameter $\mathbf{k}_s^j$ for each cluster j. Blue pixels correspond to 3D points frequently occluded during scene browsing. Brighter $\mathbf{k}_s^j$ values correspond to more specular surfaces.

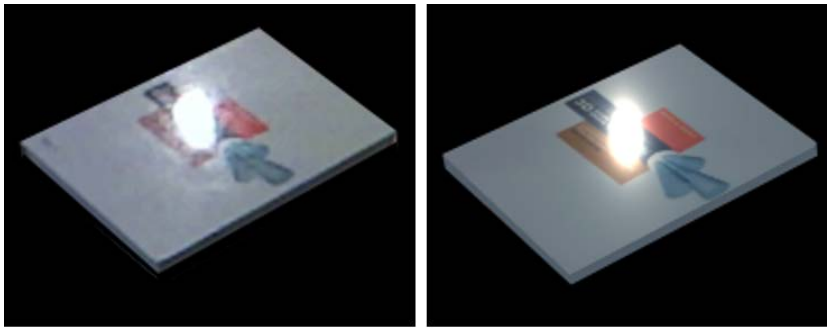


Figure 4.19 – Comparison between input image (a) and virtually rendered image using estimated reflectance and illumination (b).

Mixed Reality

Using recovered reflectance and illumiation, we realistically blend virtual objects within the real scene. This is achieved in a two-step procedure: first, using the estimated

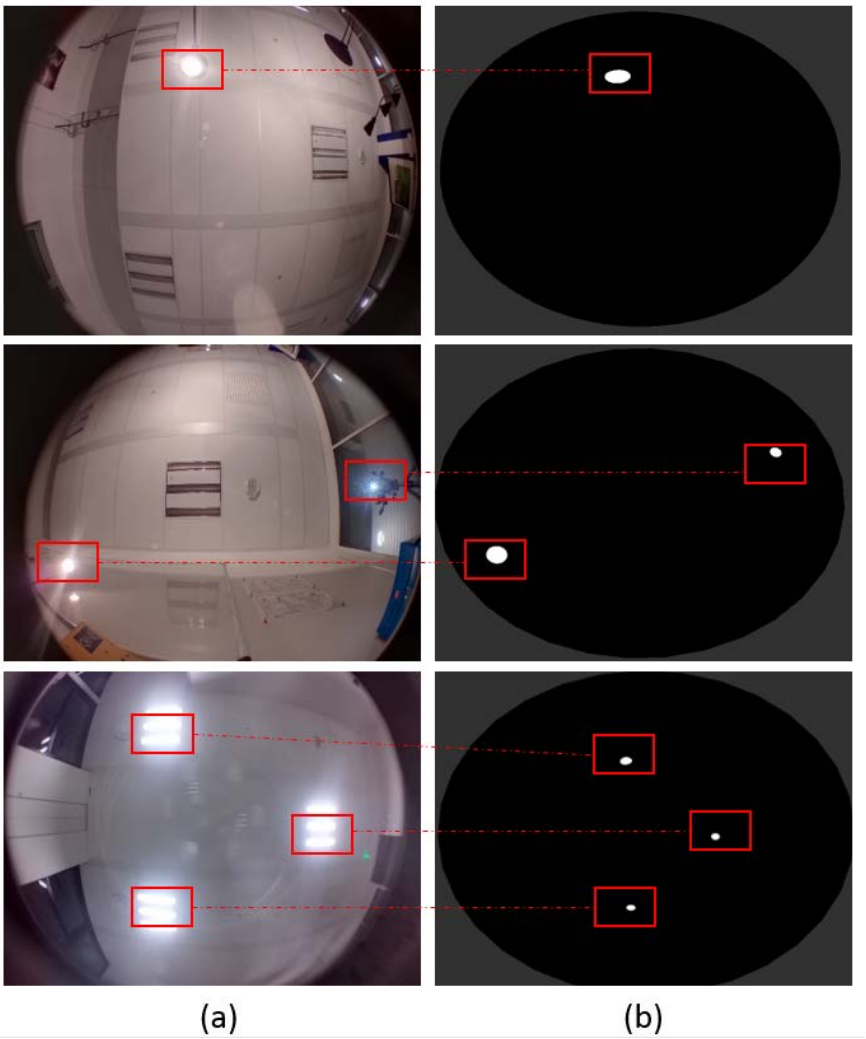


Figure 4.20 – (a) Captured environment maps using a fish-eye lens. (b) Recovered lighting respectively for S1 (row-1), S2 (row-2) and S3 (row-3) with respectively one, two and three *main* light sources.

average diffuse components in both visible and occluded surfaces (section 4.2.6.1), we compute an attenuation coefficient β as the ratio of both recovered values:

$$\beta = \frac{\bar{\mathbf{I}}^p_{d,O}}{\bar{\mathbf{I}}^p_{d,O} + \bar{\mathbf{I}}^p_{d,V}} \tag{4.25}$$

The second step consists in handling the rendering of virtual shadows using our photometry-classification (section 4.2.5). If the virtual object occludes an OP point, no shadow is rendered. On the other hand, if the virtual object occludes a point p belonging to another subgroup, we render the virtual shadow by multiplying its diffuse component $\mathbf{I}^p_d$ by the attenuation factor β. Figure 4.21 shows a variety of augmented scenes where virtual shadows and specularity removal are correctly rendered. The most important aspects of realism are the synthetic shadows (same attenuation as the spatially-close real shadows) and the removal of real specular reflections by virtual objects to observe

the recovered diffuse component. Further results and rendering comparisons are shown in this video.

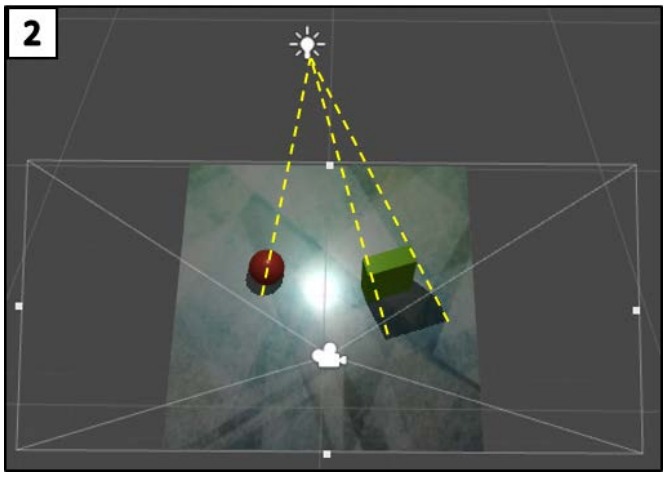
Figure 4.21 – Augmented scenes, with different reflectance and illumination conditions. We demonstrate correctly rendered virtual objects as they occlude real specular reflections (note the presence of the recovered diffuse component in the occluded region) and cast realistic shadows on real surfaces.

4.4 Conclusions and Future Research Directions

In this chapter, we presented a method to recover diffuse and specular reflection components for indoor real scenes from an RGB-D sequence. Moreover, we estimated the 3D position of light sources responsible for specular effects within the scene. Our photometric estimates were then used to correctly insert virtual objects in the real scene and deliver realistic MR scenarios. The proposed approach handles indoor real scenes with one or more objects and does not make any assumption with regard to the texture of the scene (e.g., we do not assume a per-object constant color). Furthermore, our illumination estimation is not reduced to a single point light nor to a distant lighting representation.

As demonstrated within this chapter, visual cues observed through acquired image sequences can be efficiently used to recover reflectance and illumination properties. We are therefore interested in considering even further cues. Also, the proposed approach does not run in real-time and does not handle dynamic light sources. Nonetheless, such requirements must be fulfilled within MR scenarios.

5 Photometric Registration using Cast Shadows



Contents

Our goal is to achieve a realistic blending between virtual objects and real-world scenes for MR applications. A key step consists in recovering the photometric properties of the real scene, namely reflectance and illumination. In this work, we are specifically interested in probeless photometric registration approaches which rely on the use of a single sensor to capture the scene. Such methods take advantage of the information brought by the sensor's RGB or RGB-D stream to infer reflectance and/or illumination. In the previous chapter 4, we proposed a method which, based on observed specular reflections, estimates both diffuse and specular components of scene surfaces and recovers the 3D position of light sources responsible for specular effects in the scene.

The proposed approach in chapter 4 represents an offline process which is performed prior to the MR scenario. In fact, the core idea consists in recovering luminance variations due to specular effects through a sequence of images captured from different viewpoints. The end-user must therefore browse the scene in order to bring out these specular reflections. This rises two practical constraints: first, the light sources in the scene must be static. Preserving the same illumination throughout the MR scenario can be constraining for a user who might need to change the lighting conditions; in this case, the scene's photometric registration must be performed again. Also, specular reflections are view-dependent cues which are observed only at a definite angle. Hence, capturing them might be, in some cases, laborious since the camera/observer's viewpoint must be roughly aligned with the ideal specular reflection direction.

Most importantly, two critical scenarios can not be handled by the previous approach. The first scenario consists in considering a real scene with mainly Lambertian surfaces (Figure 5.1-a). In this case, recovered luminance profiles do not retain enough variations to accurately achieve the photometric registration. The second scenario consists in considering a real scene where the specular effects that a light source creates are never or hardly observed. For instance, in figure 5.1-b, two light sources are responsible for the illumination in the scene. Nonetheless, only the specular effect due to one of them is captured (green-circled), the second one is mostly occluded by the brown box (red-circled). Within both described critical scenarios, one can notice the presence of another interesting cue from which the scene's photometric properties can be inferred: **shadows**. In fact, in figure 5.1-b, in presence of two light sources, both cast shadows are observed on the table.

In order to understand the information content of shadows, one must first recognize that shadows come in two types, depending on how they are formed on surfaces. We will refer to the two types as *cast* and *self-shadows*. Shadows are regions of a surface which receive no illumination from a light source. Self shadows are formed when a surface obstructs the light falling on itself. Cast shadows are formed on a surface when another surface occludes it from the light source. As such, shadows are potentially informative about the illumination in the scene. Furthermore, they represent strong and reliable visual cues since they are view-independent and omnipresent when real scenes are observed.

Several scenarios and applications, including MR, can benefit from the information

(a) (b)

Figure 5.1 – Critical scenarios for the method described in chapter 4. (a) Capture of a real scene where most surfaces hold only a Lambertian property. (b) Capture of a real scene with two main light sources: the specular effect created by the first light source is captured (green-circled) whereas the second one is mostly occluded by the brown box (red-circled).

brought by shadows. Consequently, detecting and deriving illumination from these cues has been extensively studied within the computer vision community [Sato et al., 1999][Finlayson et al., 2009][Panagopoulos et al., 2009][Zhu et al., 2010][Panagopoulos et al., 2011][Arief et al., 2012][Guo et al., 2013]. Nonetheless, it still remains an extremely challenging problem and no generic solution exists. For instance, in [Arief et al., 2012], scene geometry is reduced to a single cube casting a shadow on a single-color surface: the cuboid shape is used to establish correspondences between the sharp corners of the cast shadow and the upper corners of the cube. From these correspondences, lines are formed and their interesection corresponds to the 3D position of a single light source. Other approaches such as [Sato et al., 1999] consider scenes with textured surfaces but require a heavy user intervention such as removing the occluding objects from the scene and inserting them back. When both scene content and user intervention constraints are relaxed, processing time requirements for MR scenarios are not satisfied. For instance, in [Panagopoulos et al., 2009][Panagopoulos et al., 2011], it takes 3 to 5 minutes to process a single color image. Finally, within all these approaches, the dynamic lighting case is not addressed.

In this chapter, we propose a method which addresses these limitations. Specifically, we consider the problem of estimating the 3D position and intensity of multiple light sources without using any light probe. The light sources can be static and/or dynamic (e.g., turned on/off, moved, etc.). Moreover, considered real scenes can be composed of one or more 3D objects which can be of arbitrary shapes. Most importantly, no assumption is made with regard to the intrinsic color of scene surfaces: the method handles both single-color and textured surfaces. Finally, our approach recovers illumination characteristics (position and intensity) at an interactive frame rate. To summarize, the main contributions of this chapter are:

- Detection of cast shadows on textured surfaces using a coarse 3D model and color images of the scene.

- Estimation of the 3D position of static and/or dynamic light sources for each incoming frame using the information brought by the detected cast shadows.
- Estimation of the intensity of the recovered light sources.
- Near real-time implementation of the proposed method in order to meet MR scenarios requirements.

In the remainder of this chapter, we first describe the main challenges within the shadow detection task, especially for indoor real scenes with textured surfaces. Then, we describe our proposed approach to tackle these challenges. Finally, results are discussed and our estimates are used to show realistic MR scenarios.

5.1 Problem Description

A shadow occurs when illumination coming from a light source is partially or totally obstructed by one or more objects. Since less illumination reaches these regions, both cast and self-shadows have a lower luminance in comparison with their surrounding regions (Figure 5.2). Nonetheless, one can not precisely tell if a surface is "dark" due to its intrinsic color/texture or its shadowing/shading by considering only its local appearance. To illustrate, in figure 5.2, a patch located in a shadowed region retains -to some extent- the same color as a non-shadowed region (red boxes). Another case where such regions can be misinterpreted occurs when the real scene is lit by a spot light. This type of light source has a cone of influence: points outside of this cone do not receive illumination from the light source. Consequently, their local-appearance can be easily confused with shadowed regions (e.g., the green-box patches in figure 5.2).

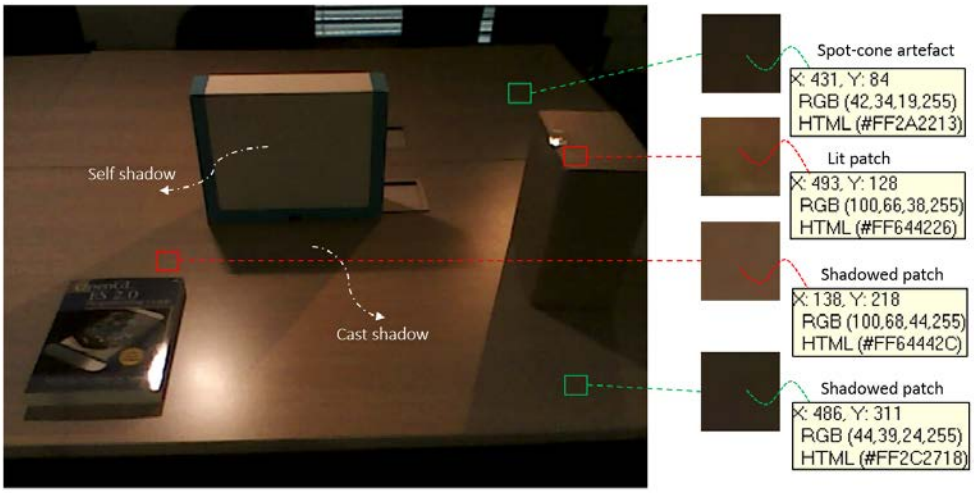


Figure 5.2 – The local appearance of surfaces can be ambiguous to recognize shadows within an image: red-box patches correspond to points under different lighting conditions, yet they have similar colors. The upper green-box patch is often erroneously detected as a shadowed region.

In presence of textured surfaces, such misinterpretations occur more often. For instance, in figure 5.3-a, within a small window (W), patches A and B have a similar local appearance although they are subject to different lighting conditions (A is in a shadowed region). The problem of separating the intrinsic color of surfaces from their shading/shadowing effects within an image, also referred to as *intrinsic image decomposition* (chapter 3), has been extensively studied [Grosse et al., 2009]. These approaches rely on the Retinex theory, proposed by [Land and McCann, 1971], stating that the intrinsic color is characterized by sharp edges while shading/shadowing varies slowly. However, these assumptions do not usually hold: in figure 5.3-b, both texture and shadowing exhibit strong discontinuities.

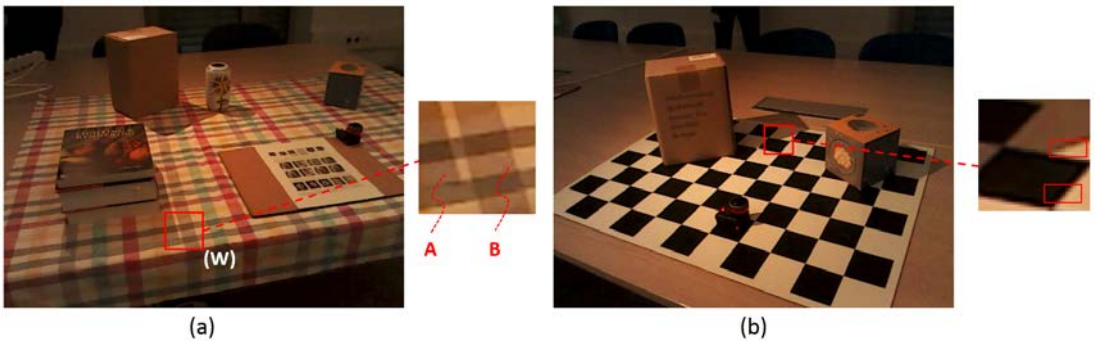


Figure 5.3 – Challenging shadow detection scenarios: (a) patches A and B have similar local appearance but A belongs to a shadowed region and B does not. (b) The Retinex theory assumptions [Land and McCann, 1971] do not hold since both texture and shadows have sharp edges.

An interesting alternative to detect shadows in captured image consists in comparing paired regions that are likely to be of the same material but are subject to different lighting conditions [Guo et al., 2013][Duchêne et al., 2015]. To illustrate, in [Guo et al., 2013], a shadowed region is detected by considering both its local appearance and surrounding regions. The local analysis is achieved by representing the color with a histogram in the L*a*b space (21 bins per channel) and texture with a texton histogram (128 textons) [Martin et al., 2004]. For the surrounding regions analysis, the image is first segmented into a set of regions using the mean shift algorithm [Comaniciu and Meer, 2002]. The recognition of same-reflectance regions is achieved using various similarity metrics such as the ratio of their intensities, their chromatic alignment, and their distance in the image. Pairwise relationships, together with local appearance features are incorporated in a shadow/non-shadow graph. Finally, the regions are jointly classified as shadow/non-shadow using graphcut inference (Figure 5.4).

The method proposed by Guo et al. [Guo et al., 2013] was tested on our captured indoor scenes images (Figure 5.5). Based on the generated results, we can see that the shadow detector is generally good at detecting shadows within regions with a roughly single color (green boxes). However, as demonstrated by the failure cases (red boxes), the detector delivers poor results in presence of textured surfaces (Figure 5.5- second row). Furthermore, the detector does not handle the presence of two cast shadows with different intensities as it only detects the darkest one (first row).

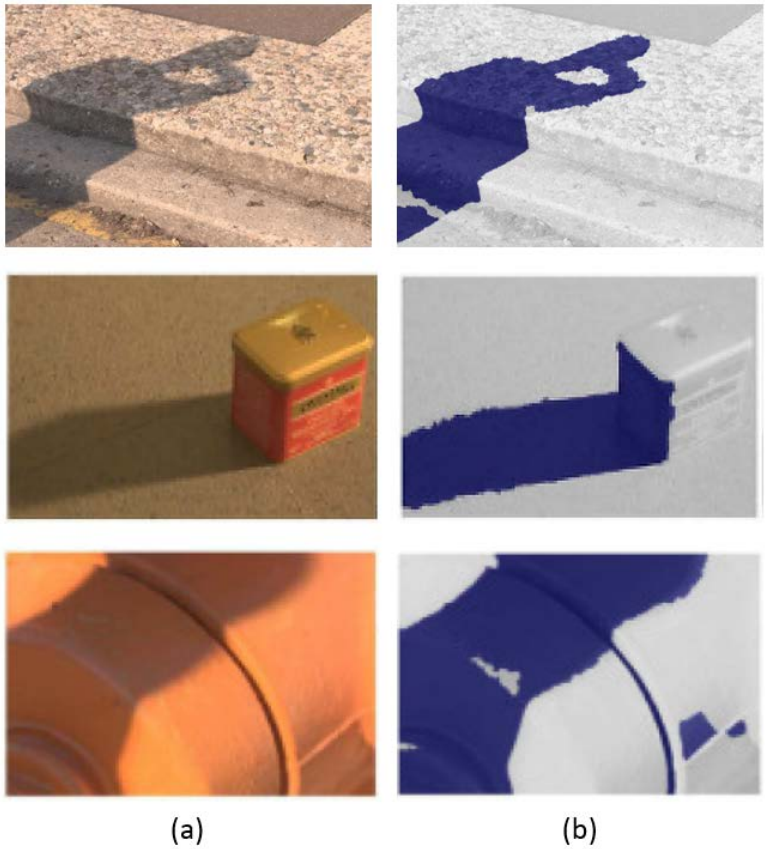


Figure 5.4 – Results of [Guo et al., 2013]: (a) input color images. (b) Dark blue pixels correspond to the detected shadows. Figures from [Guo et al., 2013].

Our contribution consists in addressing the problem of deriving illumination information from cast shadows within real scenes with arbitrary textures. To achieve this goal, we incorporate geometry into a framework which relies on the comparison of paired-regions with similar reflectance but subject to different lighting conditions. In fact, since shadows are caused by the occlusion of light sources by scene geometry, combining 3D information along with 2D analysis results in a more robust photometric registration (e.g., textured surfaces, multiple shadows, spot-cone effect, etc.).

5.2 Our Proposed Approach

The proposed framework takes three inputs: (a) 3D model of the scene (e.g., coarse model acquired with an RGB-D sensor (Intel R200)). (b) *near-ambient reference image* to which we will refer as *reference image*. The latter is acquired by simulating an ambient lighting which, in theory, does not generate any shading or shadowing. In practice, it can be easily produced by considering a fairly uniform indirect lighting. (c) color images of the scene from which illumination will be recovered overtime (Figure 5.6).

The assumptions we made for the proposed approach are: (i) scene geometry is as-

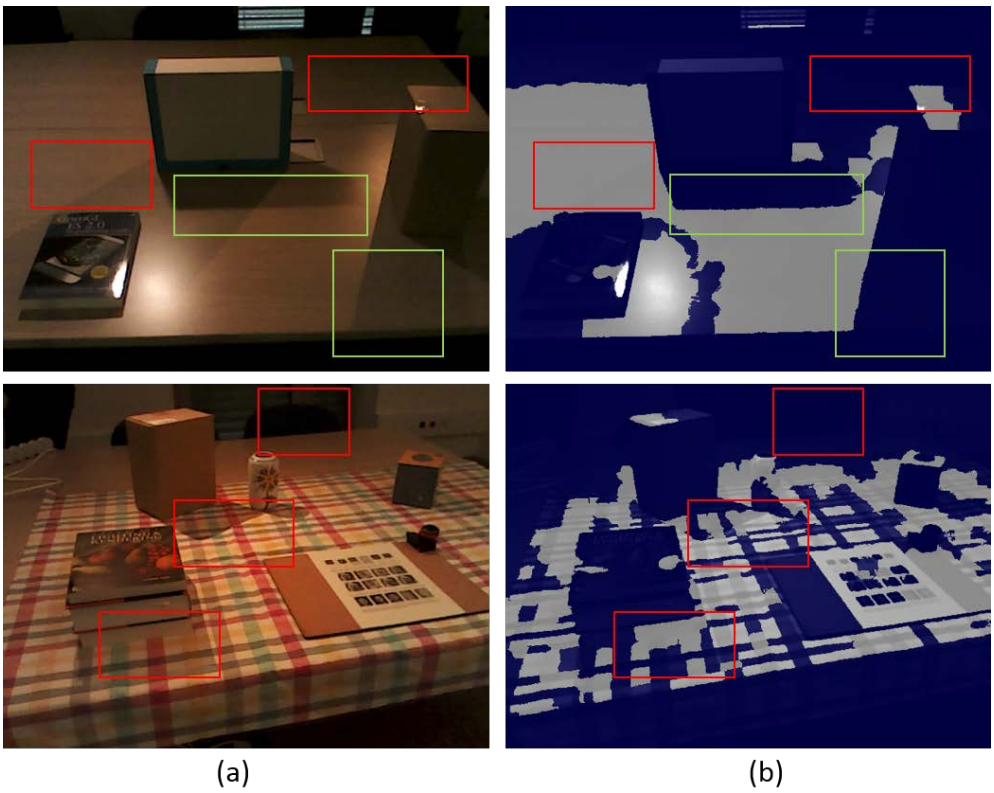


Figure 5.5 – (a) our input color images. (b) Dark blue pixels correspond to the detected shadows. Green and red boxes highlight respectively successful and failure cases of Guo et al. method for our captured indoor scenes. The code is available here.

Figure 5.6 – (a) Acquired 3D model of the scene using the Intel R200 sensor. (b) *reference image* of the real scene. (c) color image of the captured scene.

sumed to be static and contains a main planar surface on which shadows are cast (e.g., table, desk, floor). As previously mentioned, our framework uses the 3D model, acquired by the Intel R200, to recover illumination. This sensor internally refines planar surfaces within the scene [Keselman et al., 2017] and delivers accurate and smooth meshes (Figure 5.7). We therefore take advantage of this feature for robustness considerations. (ii) scene reflectance is described by the Lambertian reflection model. Hence, the way a diffuse point p in the scene reflects light can be described by [Phong, 1975]

as follows:

$$\mathbf{I}^p = \mathbf{k}_d^p(\mathrm{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathrm{L}_i\mathrm{O}_i^p) \tag{5.1}$$

where $\mathbf{I}^p$ is the color of 3D point p, $\mathbf{k}_d^p$ is its albedo and $\mathbf{n}^p$ is its normal vector. L_a and L_i are respectively the intensities of ambient lighting and light source i, ω_i^p is the incoming light direction vector of light source i, and M is the number of light sources present in the scene. O_i^p is a binary visibility term, equal to 1 if point light i is visible from the 3D point p and equal to 0 if occluded. (iii) the viewpoint is fixed within the analysis part but can vary within MR scenarios.

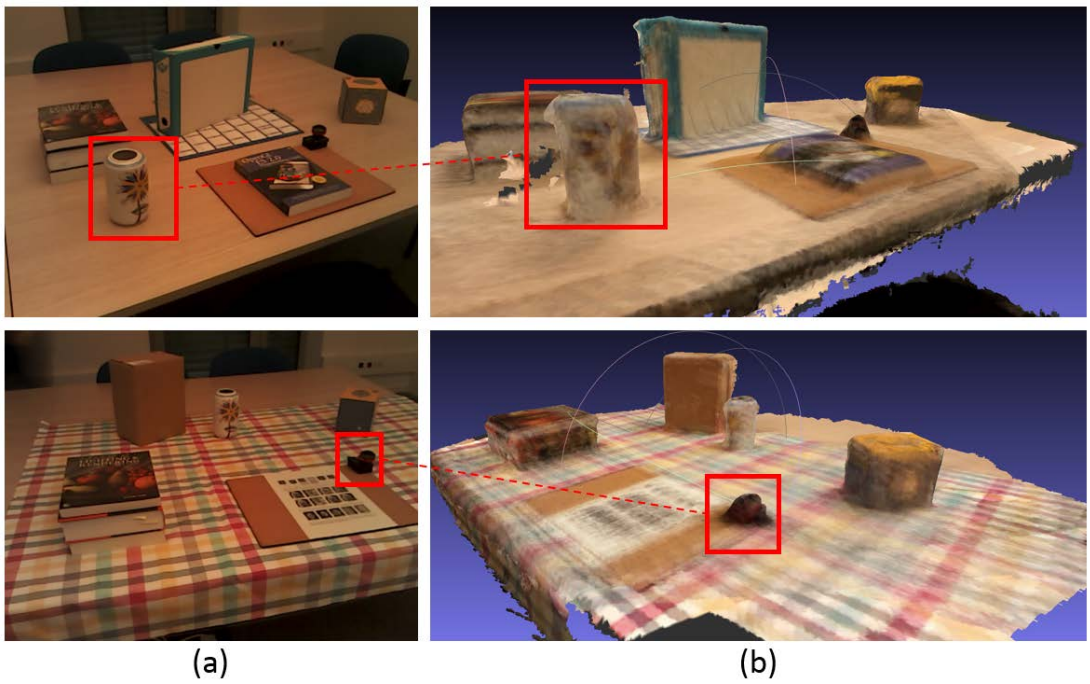


Figure 5.7 – (a) Color image of the captured scenes. (b) Acquired 3D model of the scenes in (a) using the Intel R200 sensor: the planar surface corresponding to the table within the scene is smooth. The reconstructed geometry of the specular cylinder and the black GoPro camera (red boxes) is of lower quality.

To recover the illumination in the scene, our approach relies on two key ideas (Figure 5.8):

- For every input color image, we separate texture/color variations from shadowing effects. This is achieved using a voting scheme where a 3D point p is compared, in terms of shadowing/shading, to 3D points holding similar diffuse reflectance as point p. In fact, if we consider a pair of points p and $\hat{p}$ with similar reflectance $\mathbf{k}_d^p$ but subject to different illumination conditions (p is occluded with regard to lighting whereas $\hat{p}$ is not), the ratio of their respective colors using [Phong, 1975] is described as follows:

$$\delta(p) = \frac{\mathbf{I}^p}{\mathbf{I}^{\hat{p}}} = \frac{\mathbf{k}_d(\mathrm{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathrm{L}_i\mathrm{O}_i^p)}{\mathbf{k}_d(\mathrm{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathrm{L}_i)} = \frac{\mathrm{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathrm{L}_i\mathrm{O}_i^p}{\mathrm{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathrm{L}_i} \tag{5.2}$$

Consequently, δ corresponds to a ratio of illumination since diffuse reflectance $\mathbf{k}_d^p$ cancels outs. In the following, we will refer to δ as the *illumination ratio map*. The identification of point pairs retaining the same diffuse reflectance is robustly achieved using the *reference image*. In fact, the color of a point p within this image $\mathbf{I}_{ref}^p$ is described as follows:

$$\mathbf{I}_{ref}^p = \mathbf{k}_d^p \mathrm{L}_a \tag{5.3}$$

Since all scene points are equally illuminated by the pseudo-ambient lighting L_a, the color of scene points within this image mainly contains texture/color (no apparent cast shadows) and represents a robust similarity feature.

- Using the 3D model of the scene and a set of hypothetical point light sources, we render a shadow map for every light source. By considering dense matching techniques, the 3D position of light sources corresponds to the best matches between the *illumination ratio map* and the generated shadow maps.

In the following sections, we describe in details the main components of the proposed photometric registration approach.

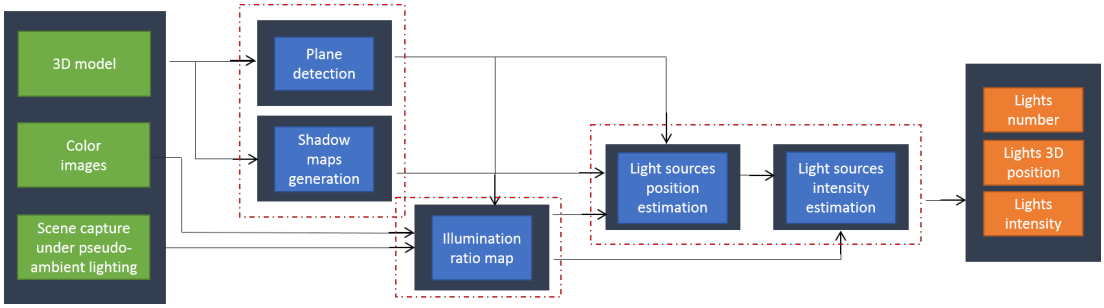


Figure 5.8 – Outline of our photometric registration approach using cast shadows.

5.2.1 Estimation of Illumination Ratio Maps

Shadows are caused by the occlusion of incoming light, and thus contain various pieces of information about the illumination of the scene. In this section, our goal is to separate texture variations from cast shadows. The considered inputs are the 3D model of the scene and the *reference image*. To achieve the texture and illumination separation for each incoming color image, we follow a two-fold procedure:

The first step is achieved only once since geometry is static: we detect the main planar surface on which shadows are cast. Specifically, we compute surface normals $\mathbf{n}$ using [PCL, 2013] and apply a region growing algorithm to cluster similarly oriented surfaces together (a deviation value of 3 degrees is typically allowed between normals). Then, we use a RANSAC estimator to fit each cluster to a planar surface model. Finally, the cluster including the largest number of inliers is considered as the main plane. Points above the detected plane are further grouped as belonging to the occluding objects (Figure 5.9).

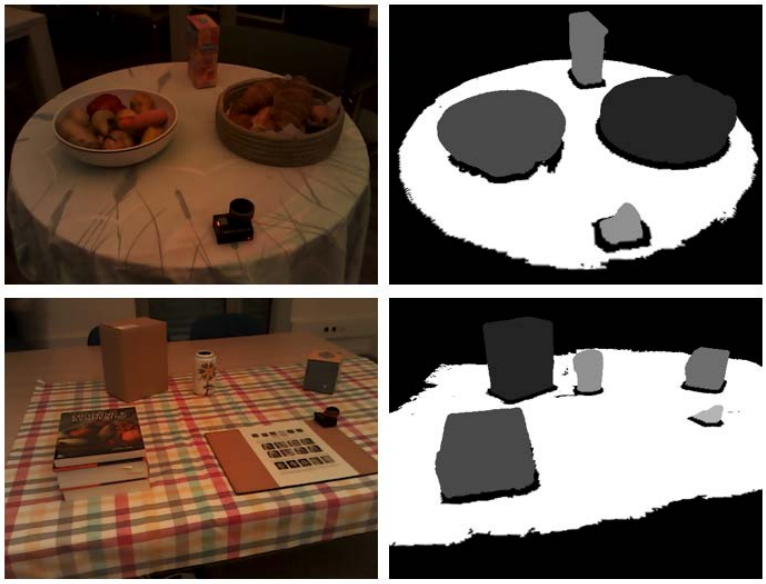


Figure 5.9 – (a) Color image of real scenes. (b) Clustered 3D models of scenes in (a): white pixels correspond to 3D points belonging to the planar surface. Points which belong to an occluding 3D object are represented by pixels with similar grayscale values. Black pixels correspond to either the background (the geometry of the background is not available) or points with noisy normals.

Secondly, we aim at separating texture/albedo and illumination in the current frame. The analysis is limited within the 2D projection of the previously detected plane where cast shadows can be encountered. Our separation is achieved through a voting scheme using pairs of points $(p, \hat{p})$ with the same reflectance $\mathbf{k}_d$ but subject to different lighting conditions (Figure 5.10).

The selection of pairs $(p, \hat{p})$ is based on two features: (i) L^2 norm of pixels color in the CIELAB color space within the *reference image* $\mathbf{I}_{ref}$ as it provides accurate similarity measures compared to using only the current frame (Figure 5.11). In fact, the use of the *reference image* $\mathbf{I}_{ref}$ makes our algorithm robust in presence of challenging textures, poor lighitng conditions and/or sensor noise. (ii) the lightness channel L of the CIELAB color space within the current image enables us to compare pairs illumination-wise. Hence, points with lower lightness values are prone to belong to shadowed regions. Using these two features, the voting scheme is applied as follows:

$$v(p) = \begin{cases} +1, & \text{if } \left\| \mathbf{I}^{p}_{ref} - \mathbf{I}^{\hat{p}}_{ref} \right\| \leq \epsilon_{ref} \text{ and } L^{\hat{p}} \geq L^{p} + \epsilon_{L} \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

where ϵ_{ref} and ϵ_L are respectively thresholds with respect to color vectors distance in $\mathbf{I}_{ref}$ and lightness difference in L. A value of 2.5 is chosen for ϵ_{ref} as it corresponds to an *almost noticeable difference* [Sharma, 2002]. ϵ_L is kept at a low value (typically 10) to handle both weak and strong shadows. Finally, pixels holding a significant voting value $v(p)$ are further considered to estimate their respective illumination ratio map

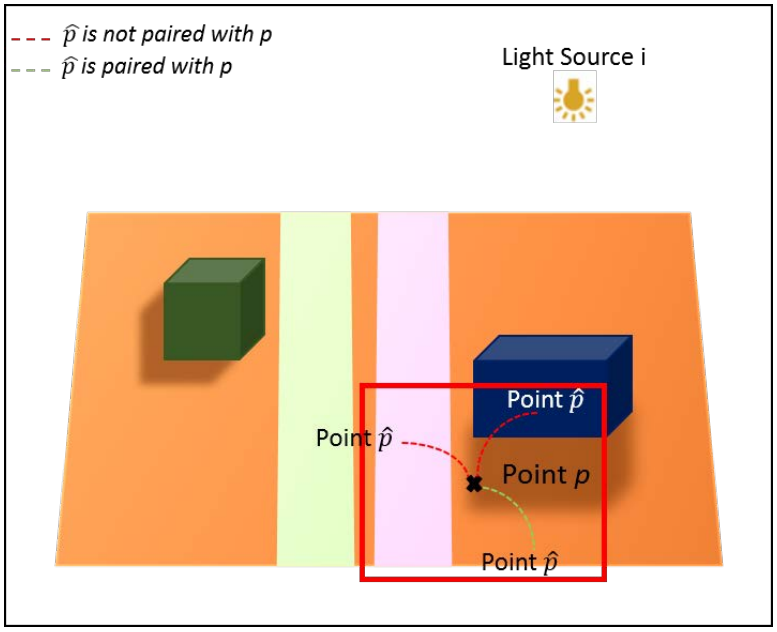


Figure 5.10 – Voting scheme to detect cast shadows: points paired with p correspond to 3D points $\hat{p}$ which belong to the detected planar surface and hold a similar diffuse reflectance as point p. In this scenario, point p received one vote out of three performed comparisons.

value $\delta(p)$:

$$\delta(p) = \frac{L^p}{\bar{L}} \qquad \text{with:} \qquad \bar{L} = \frac{\sum_{\hat{p}} L^{\hat{p}}}{v(p)} \tag{5.5}$$

where $\bar{L}$ is the mean lightness value of matched points $\hat{p}$. Furthermore, pixels for which similar-reflectance pairs are found but received a low voting value hold an illumination ratio value equal to 1. In fact, when these points are compared with their similar-reflectance points, they are never found to have a lower lightness value which implies that they must not be occluded with regard to the light sources in the scene. Last but not least, pixels for which no match is found are discarded (green pixels in figure 5.11).

5.2.2 Estimation of Light Sources 3D Position

In this section, our goal is to recover the 3D position of light sources responsible of cast shadows in the scene. An initial illumination distribution corresponds to a set of virtual point lights equally distributed in the 3D space above the detected plane (Figure 5.12-a). The core idea, with regard to estimating the 3D position of light sources, consists in extracting a subset (S) of point lights whose shadow maps (Figure 5.12-b) correlate with the estimated *illumination ratio map* in section 5.2.1.

The identification of subset (S) is carried within an iterative process as follows:

1. We initially compute correlation values by matching the *illumination ratio map* with the shadow maps of the sampled light candidates. The light source whose

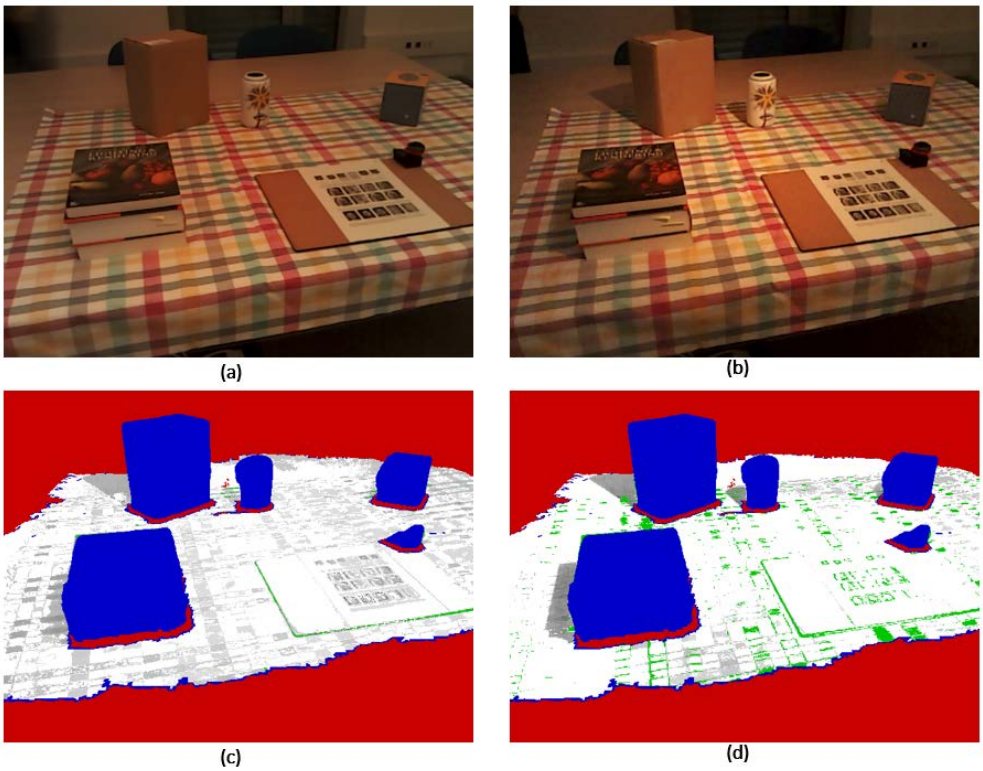


Figure 5.11 – (a): *reference image* $\mathbf{I}_{ref}$ of the scene. (b): current scene capture. (c)(d): recovered *illumination ratio maps* using respectively (b) and (a). (d) demonstrates a better separation of texture and lighting. Note the shadow of the front book in (d) compared to (c) as well as discarded pixels (green).

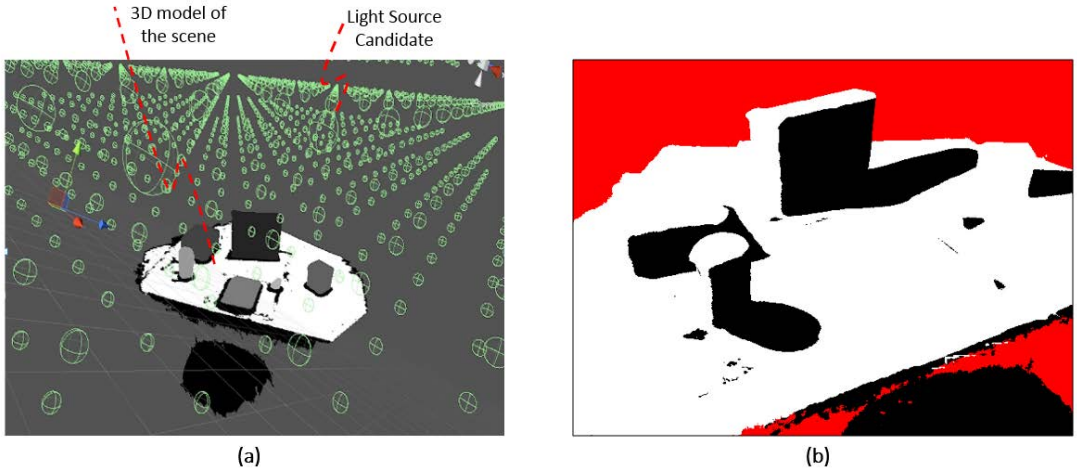


Figure 5.12 – (a) Initial distribution of candidate light sources. They are represented by point lights located at the center of the green spheres. (b) An example of a generated shadow map using the 3D model of the scene and a candidate light source from (a).

shadow map has the best correlation value is selected.

2. For each iteration, previously selected light sources are discarded. Also, the

matching operation within the current iteration is carried out by discarding previously matched pixels.

3. The process ends either when the currently selected shadow map has a significantly low matching value or if the number of selected lights is higher than N. In practice, we set N to be equal to 4.

The chosen correlation metric corresponds to Pearson's correlation coefficient given by the following formula:

$$\Phi_i = \frac{\sum_{j=1}^{P}(\delta(p_j) - \bar{\delta})(\mathrm{O}_i^{p_j} - \bar{\mathrm{O}}_i)}{\sqrt{\sum_{j=1}^{P}(\delta(p_j) - \bar{\delta})^2}\sqrt{\sum_{j=1}^{P}(\mathrm{O}_i^{p_j} - \bar{\mathrm{O}}_i)^2}} \tag{5.6}$$

where i iterates over the initial set of candidate light sources and j iterates over the set of pixels which belong to the 2D projection of the principle plane P. $\delta(p)$ and O_i^p are respectively the *illumination ratio map* and i^{th} shadow map values for pixel p and, $\bar{\delta}$ and $\bar{\mathrm{O}}_i$ represent their respective mean values. The coefficient Φ_i corresponds to the correlation value between the current *illumination ratio map* and the i^{th} shadow map. Φ_i has a range between 0 and 1: perfectly matching maps have a coefficient equal to 1.

The light sources in the scene can be turned on/off and moved within the MR scenario. We therefore recover illumination for each incoming frame. Note that only the correlation procedure is performed for each input frame. In fact, the initial distribution of virtual point lights along with their rendered shadow maps need to be generated only once since the geometry is static.

5.2.3 Estimation of Light Sources Intensity

In order to correctly render virtual shadows that are consistent with the observed cast shadows in the real scene, we must recover the characteristics of the light sources illuminating the real-world. Specifically, in addition to the 3D position of the light sources (Section 5.2.2), we must recover their respective intensities. As we consider small to middle scale scenes, we assume that the shading $((\mathbf{n}^p \cdot \omega^p) = \cos\theta)$ in equation 5.1 is equal across selected pairs. Subsequently, equation 5.1 can be rewritten as follows:

$$\mathrm{I}^p = \mathbf{k}_d^p(\mathrm{L}_a + \sum_{i=1}^{M} \mathrm{L}_i \mathrm{O}_i^p) \tag{5.7}$$

Consequently, the recovered *illumination ratio* map, previously described in equation 5.2, is rewritten as:

$$\delta(p) = \frac{\mathrm{L}_a + \sum_{i=1}^{M}(\mathrm{L}_i \mathrm{O}_i^p)}{\mathrm{L}_a + \sum_{i=1}^{M} \mathrm{L}_i} \tag{5.8}$$

The normalized color vector of a pure Lambertian white pixel is $\mathrm{I}^p = (1,1,1)^\top$. As its diffuse reflectance $\mathbf{k}_d^p = (1,1,1)^\top$ [Ward, 1992], we set $\mathrm{L}_a + \sum_{i=1}^{M} \mathrm{L}_i = 1$ and rewrite equation 5.8 as follows:

$$\mathrm{L}_a + \sum_{i=1}^{M}(\mathrm{L}_i \mathrm{O}_i^p) = \delta(p) \tag{5.9}$$

Finally, by considering all the points within the detected planar surface, we obtain a linear system:

$$\mathbf{AL} = \delta \tag{5.10}$$

where:

$$\mathbf{A} = \begin{pmatrix} 1 & \mathbf{O}_1^{p_1} & \cdots & \mathbf{O}_M^{p_1} \\ 1 & \mathbf{O}_1^{p_2} & \cdots & \mathbf{O}_M^{p_2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{O}_1^{p_N} & \cdots & \mathbf{O}_M^{p_N} \end{pmatrix} \mathbf{L} = \begin{pmatrix} \mathrm{L}_a \\ \mathrm{L}_1 \\ \vdots \\ \mathrm{L}_M \end{pmatrix} \delta = \begin{pmatrix} \delta(p_1) \\ \delta(p_2) \\ \vdots \\ \delta(p_N) \end{pmatrix} \tag{5.11}$$

The linear system 5.11 is solved using an iterative Least Squares with bounds and equality constraints:

$$\begin{aligned} &\widehat{\mathrm{L}} = \arg\min_{\mathrm{L}} (\tfrac{1}{2} \left\| \mathbf{W}(\mathbf{AL} - \delta) \right\|)^2) \quad \text{subject to:} \\ &\begin{cases} 0 \leq \mathrm{L}_i \leq 1 \quad \text{and} \quad 0 \leq \mathrm{L}_a \leq 1 \\ \mathrm{L}_a + \sum_{i=1}^{M} \mathrm{L}_i = 1 \end{cases} \end{aligned} \tag{5.12}$$

where $\mathbf{W}$ is a diagonal matrix whose weights are computed using Tukey's bisquare loss function [Yu et al., 2014]. Hence, small weights are discarded throughout iterations.

5.3 Experimental Results

The proposed approach has been tested on various real scenes with varying texture and lighting conditions. In figure 5.13, we illustrate our results within a selection of four scenes (S1, S2, S3 and S4). Further results are shown in this video.

In figure 5.13, scenes are grouped row-wise. In the first column, we overlay the contours of shadow maps generated by the recovered light sources on the input color images. For instance, red and green contours are used respectively for the first and second detected lights. In the second column, we demonstrate estimated *illumination ratio maps* where background/noise are represented by red color pixels and occluding objects by blue color pixels. On the detected planar surface, green color pixels represent 3D points for which no pairs are found. Finally, grayscale pixels are the illumination ratio maps values. These pixels correspond to 3D points partially or fully occluded with regard to lighting. For instance, white color pixels represent fully lit 3D points (their illumination ratio maps value δ is equal to 1).

As illustrated in figure 5.13-b, our algorithm estimates *illumination ratio maps* where texture/albedo is accurately separated from illumination effects. The proposed framework handles both uniform surfaces (S1) and challenging textured surfaces (S2, S3 and S4). Also, we demonstrate robustness in presence of poor geometry, especially when the scene contains specular objects (cylinser and books in S1 and, tea box in S3). In fact, the Lambertian assumption must mainly hold on the principal plane. Finally, the use of the *reference image* $\mathbf{I}_{ref}$ enables us to be robust even when lighting and/or

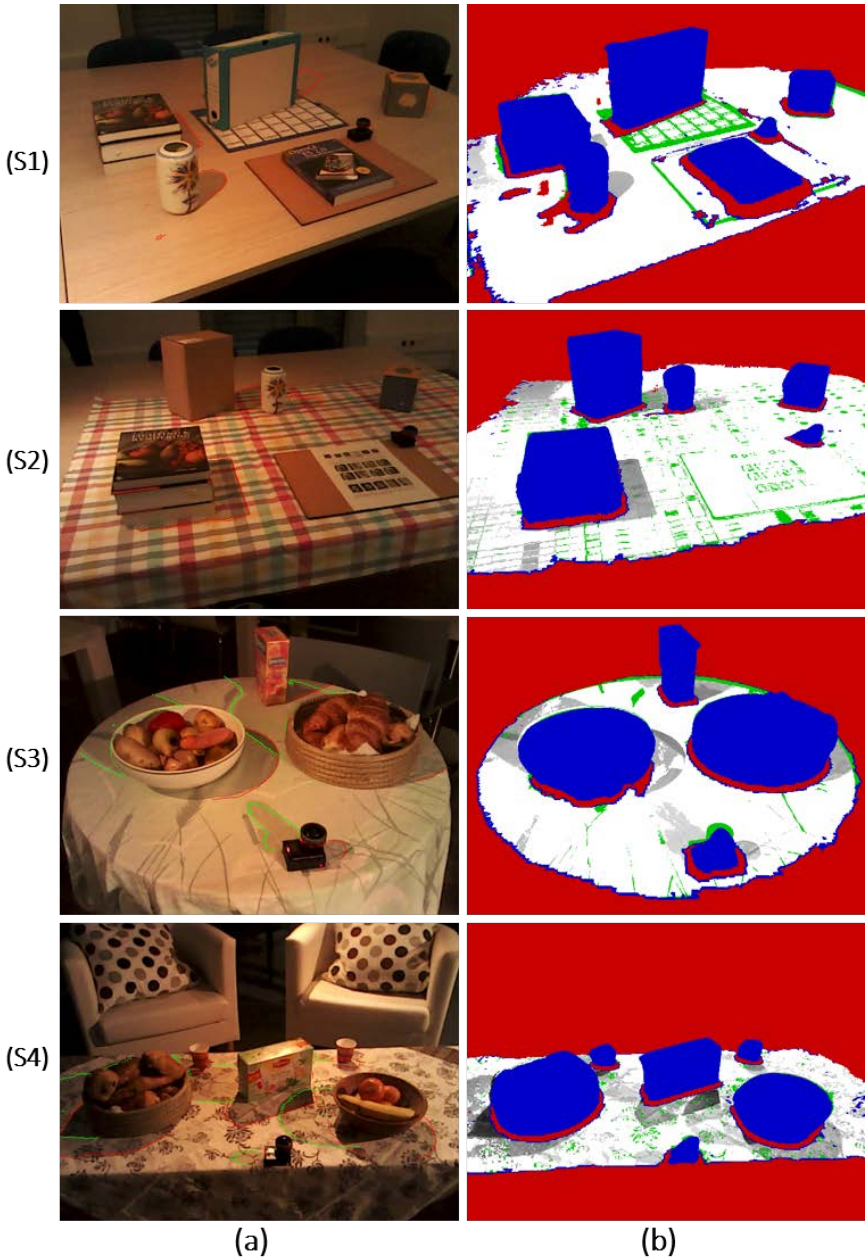


Figure 5.13 – (a) Overlay of the contours of selected shadow maps (i.e. selected point lights) on the input color image. (b) Estimated *illumination ratio maps* δ for real scenes in (a) with fairly uniform (S1) and textured surfaces (S2, S3 and S4).

image quality lack (S4). In fact, in case of low lighting, the information within shadowed regions can be noisy (sensor noise). Consequently, considering the color within the current image as a similarity feature is not as robust as considering the image $\mathbf{I}_{ref}$ captured under a pseudo-ambient lighting.

Recovered 3D Position of Light Sources

In figure 5.13-a, we overlay the contours of the selected shadow maps on the current color frame. For S2, the *illumination ratio map* contains shadows but also the effects corresponding to a narrow spot light cone. Nonetheless, our algorithm succeeds in recovering an accurate light position. Furthermore, our approach demonstrates good results in the presence of overlapping shadows (S3 and S4). In fact, since we perform a dense matching between shadowed regions in both the illumination ratio map δ and generated shadow maps, our iterative process efficiently matches the shadow pixels originating from the occlusion of light sources in the scene.

In order to evaluate the precision of recovered light sources positions, we used a special setup for several experimental scenes. First, we choose a world frame on the main plane and measure distance to real lights using a telemeter. Our algorithm is tested on various scenes and recovers light sources position with an average error of 17cm for a mean distance of 2.55m to the light source and a standard deviation of 3.5cm.

As mixed reality is our target application, temporal stability with regard to recovered light sources is of paramount importance. When the lighting is static, recovered lighting properties (position and intensity) must be temporally stable, otherwise virtual shadows might suffer from apparent flickering. In figure 5.14, we can notice that under the same lighting conditions, the selected light source (ID98) remains the same throughout the sequence with slightly different correlation values. The second best shadow map holds a low correlation value and is thus discarded (it corresponds to a matching with few noisy pixels within the illumination ratio map).

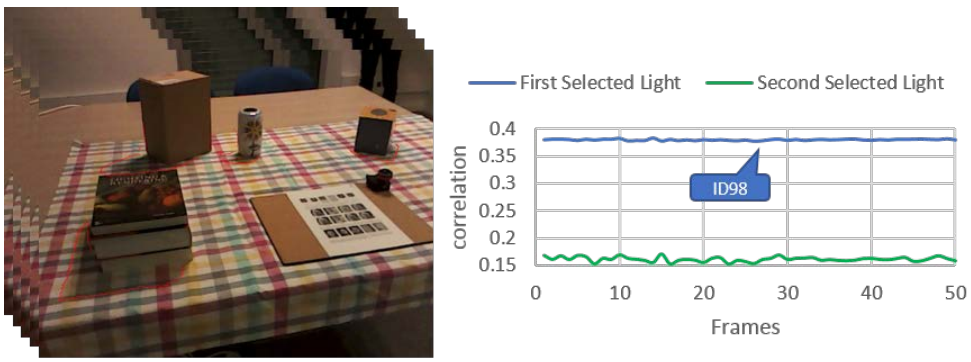


Figure 5.14 – First and second best correlation coefficients for a scene under static lighting.

Estimated Light Sources Intensity

In this section, we demonstrate the effectiveness of our approach with regard to the estimated intensity of light sources. First, we show the temporal stability of the intensity estimates throughout a sampled sequence for scenes S2 and S4 under moving light

sources (Figure 5.15). In fact, when the scenes are illuminated by a dynamic lighting, the 3D position of light sources changes but their intensity is fairly constant.

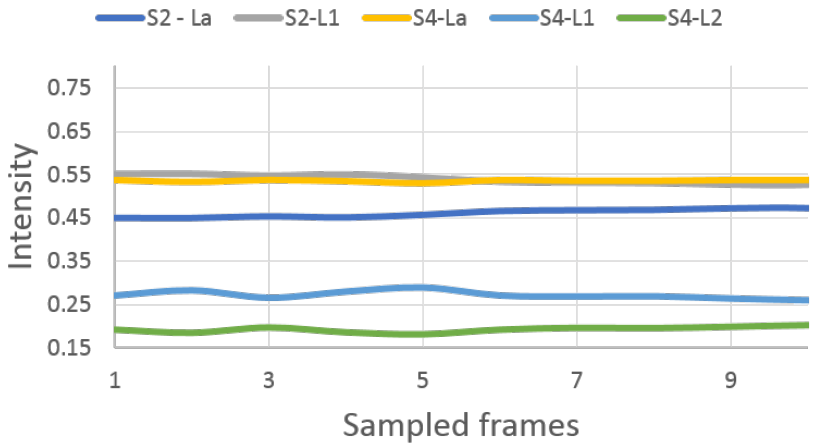


Figure 5.15 – Recovered ambient intensity L_a and point light sources intensity L_1 and L_2 for scenes S2 and S4.

Furthermore, recovered lighting intensities are used to render virtual shadows. In figure 5.16, the second column represents reconstructed shading $(L_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)L_i O_i^p)$ using scene geometry, light sources 3D position and intensity for scenes S3 and S4 (depicted in the first column). A virtual sphere is introduced as well to demonstrate its interaction with the real scene.

Mixed Reality Applications

The proposed photometric registration approach runs at an interactive frame rate (4fps) and recovers both 3D position and intensity of light sources in the real scene. These estimates are used within a rendering pipeline to illuminate virtual objects inserted in the mixed scene. In figure 5.17, we show augmented real scenes where virtual shadows are visually coherent with real shadows in terms of shape and intensity.

Moreover, as stated in section 5.2, the viewpoint is fixed. This is mainly because the generation of shadow maps with regard to different viewpoints (in case the camera moves) is time-expensive (20 seconds to generate 1176 shadow maps). Consequently, in order to allow the end-user to freely move the camera, we can either re-project the current color image, using camera pose, onto a reference viewpoint within which shadow maps are generated or follows the setup described in appendix A. In fact, the proposed approach in this chapter has been integrated within an industrial project at Technicolor (appendix A). The goal of this project is to deliver a realistic MR experience where the end-user can freely turn on/off and move the light sources in the real world and witness the changes within the virtual objects as well. In figure 5.18, we show captures of the MR demonstration running, in real-time, on a tablet. More details about the

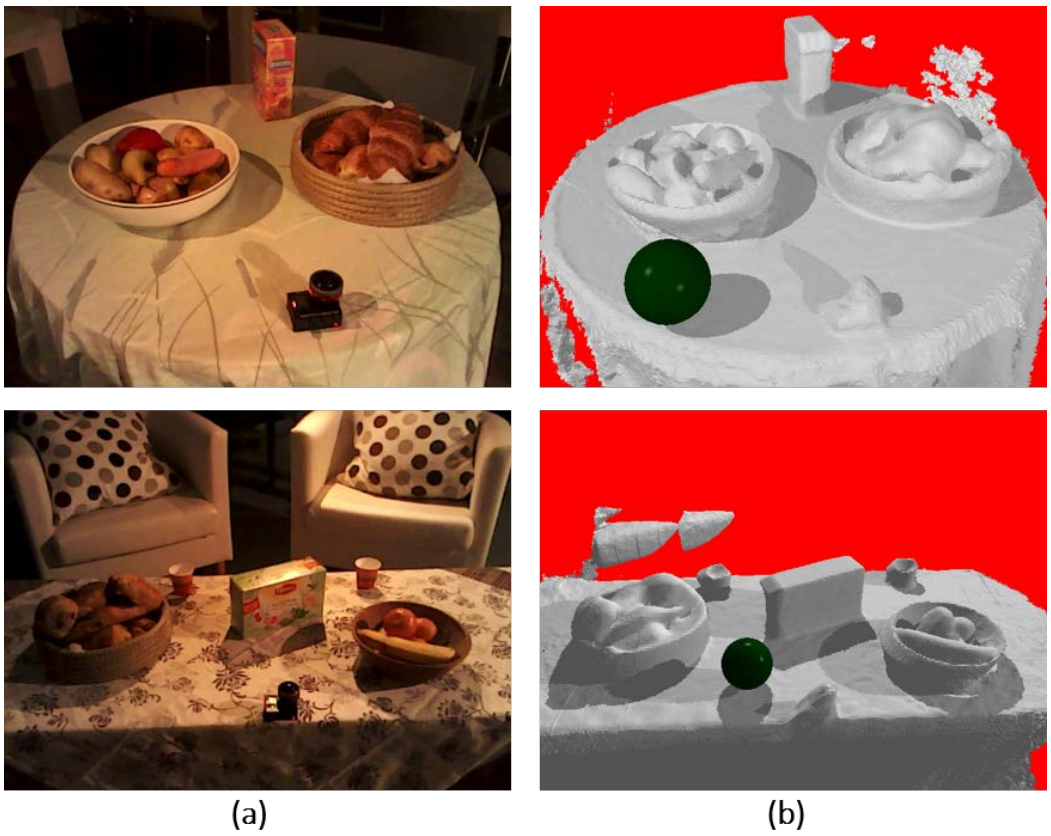


Figure 5.16 – (a) Reconstructed shading using scene geometry and recovered lighting properties (position and intensity) for S3 (a) and S4 (b).

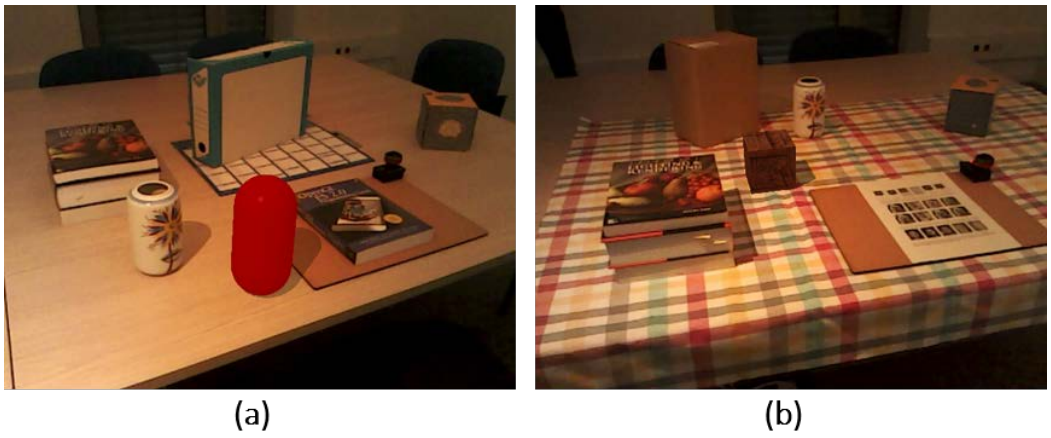


Figure 5.17 – Mixed reality scenarios with visually coherent virtual shadows such as the red capsule in S1 (a) and the brown cube in S2 (b).

demonstrator can be found in appendix A, video1 and video2.

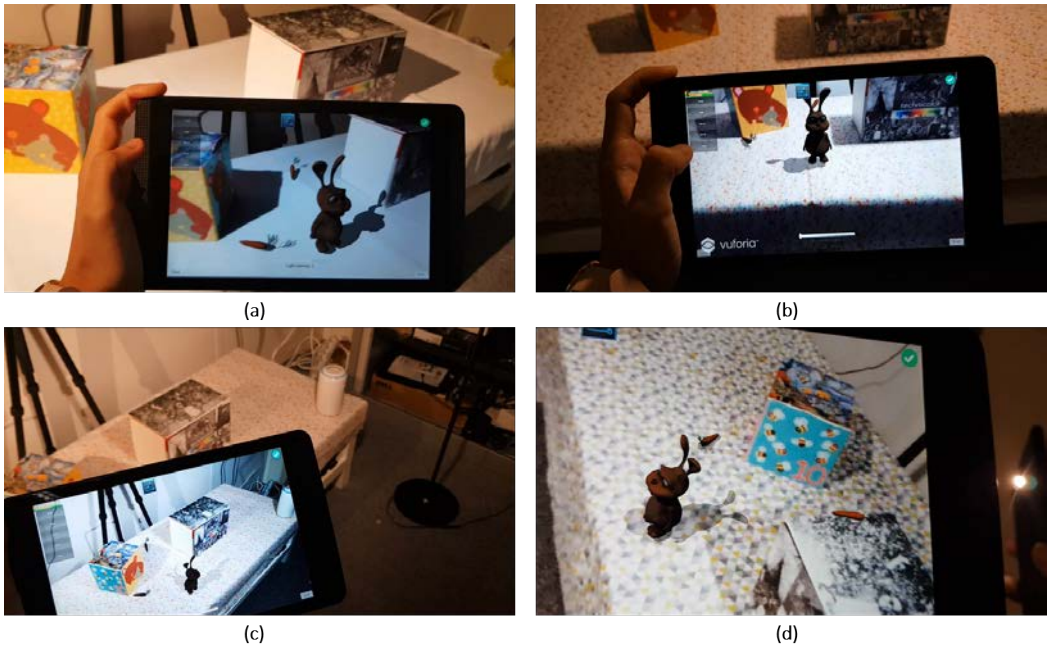


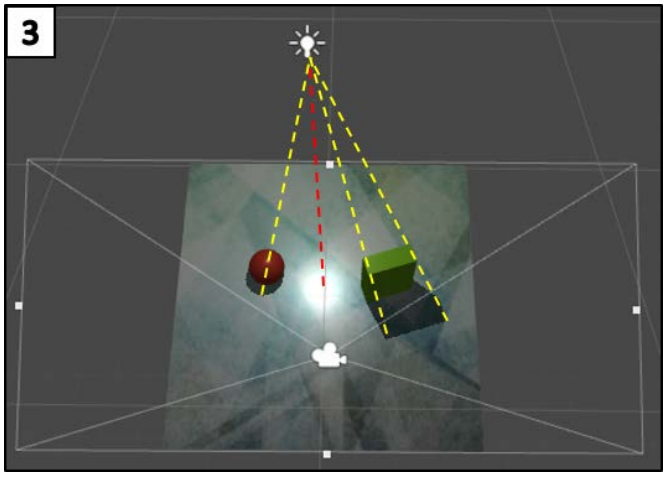
Figure 5.18 – Mixed reality scenarios where a virtual bunny augments scenes with uniform-color surfaces (a) and textured surfaces (b). Light conditions are various: a single spot light (b), two spot lights (c) and both a spot light and a phone's flash light (d).

5.4 Conclusions and Future Research Directions

In this chapter, we presented a probeless photometric approach which recovers both position and intensity of multiple light sources for indoor real scenes. The algorithm is based on a detection of cast shadows within surfaces where texture can spatially vary. The proposed method runs at interactive frame rate (4fps) which satisfies MR requirements. Furthermore, the dynamic aspect of lighting was tackled, which allows the end-user to freely turn on/off or move lights within the scene and notice near real-time changes with regard to the synthetic objects.

In the proposed framework, we assumed a Lambertian property for the main planar surface of the scene. This is due to the fact that specular reflections can disturb our cast shadow detection. Nonetheless, in real-world scenes, specularities are often encountered and must be handled. Consequently, we are interested in relaxing the Lambertian constraint and efficiently use the information brought by both cast shadows and specular reflections.

6 Photometric Registration using Multiple Cues



Contents

Our goal is to achieve a realistic blending between real and virtual worlds. This requires the estimation of the reflectance properties as well as the characteristics of light sources illuminating the real scene. One of the challenges within this task consists in recovering these properties using a single RGB-D sensor. In chapters 4 and 5, we presented two probeless photometric approaches which derive reflectance and illumination from the analysis of the scene's RGB-D stream. In fact, the first approach considers the information brought by observed specular reflections while the second method relies on the detection of cast shadows. In this chapter, we address the following question: how can we combine both of these cues to achieve a robust photometric registration ?

6.1 Problem Description

An important aspect of photometric registration approaches consists in their usability. Proposed approaches must be, as much as possible, independent from the scene's content (e.g., geometry, reflectance, illumination). In fact, constraining the MR user to have a single light source or a textureless surface reduces the range of possible scenarios. Hence, instead of enhancing the immersion, it adds cumbersomeness to the experience.

Our goal is to achieve a photometric registration which handles a variety of real scenes using a single sensor. In the previous chapter 5, we presented an approach that recovers the 3D position and intensity of multiple light sources in the scene. Moreover, the user can freely turn on/off and move the light sources since the dynamic aspect is tackled as well. However, we assumed that the surfaces hold a Lambertian property. This is mainly because specular reflections can disturb the cast shadow detection procedure. To illustrate, in figure 6.1-a, point p does not belong to a shadowed region. Nonetheless, when compared to its surrounding points (illumination-wise), it holds a lower lightness value. Hence, it can be erroneously recovered as a cast shadow point. To tackle this issue, a simple and commonly used technique consists in detecting these highlight regions within the image and discarding them within the analysis phase. While this procedure resolves the problem depicted in 6.1-a, it is not always as effective: in figure 6.1-b, handling the specular reflection (green-circled) as an outlier results in an inaccurate estimate of the 3D position of the light source. In fact, by discarding the specularity, only the cast shadow points within the red contours are recovered while the cast shadow actually extends within the blue-contoured region.

As demonstrated in chapter 4, specular reflections are informative in terms of surface reflectance and scene illumination. In fact, they represent interesting cues with regard to the direction of the light source creating them. Hence, they must be taken into account for this purpose within the photometric registration task. Moreover, when cast shadows can not be easily detected, specular reflections can be efficiently used to handle such scenarios. To illustrate, in figure 6.2-(a,b), cast shadows (red boxes) hold a significantly low intensity. It can be due to the presence of a spatially close specular effect or an additional light source with stronger intensity. In these scenarios, where weak cast shadows can be hard to detect, photometric registration approaches can benefit from the information brought by the observed specularities in the scene.

Figure 6.1 – (a) Example of a point p which can be erroneously recovered as a cast shadow point due to the presence of specular effects (using the method described in chapter 5). (b) An example where discarding specular reflections from the analysis can result in inaccurate estimates of the 3D position of light sources: recovering the 3D position only from the points within the red contours instead of considering the points within the blue contour.

Figure 6.2 – Examples of weak cast shadows, which can be hard to detect, due to the presence of close specular effects (a) and an additional stronger light source (b).

Generally speaking, most existing approaches either use specular reflections or cast shadows to derive reflectance and/or illumination. Nonetheless, the existence of a light source is more likely if it is supported by more than one cue. In fact, consistency among both cues and within each cue can lead to more robustness. To our knowledge, there are only two related works which jointly use within their photometric registration framework both specular reflections and cast shadows [Anusorn and Nopporn, 2016][Li et al., 2003]. In [Anusorn and Nopporn, 2016], specular highlights are used to estimate the light source direction. This is achieved by considering the ideal specular reflection direction roughly aligned, at the center of the detected specularity, with the view direction. Then, a search of the light source's 3D position is considered along the recovered direction. The final estimate of the light source position is achieved using the corners of a shadow cast on a single-color and Lambertian surface. Beside the imposed constraints of this method with regard to having a scene reduced to a simple known object and a

Lambertian surface, recovering the light source position along the estimated light direction can be inaccurate. To illustrate, in figure 6.3-a, depending on the thresholds used to recognize the highlights within the image, the direction of the light source can be recovered using either the red-circled specularity or the green-circled specularity. Consequently, this results in an error with regard to the estimated 3D position of the light.

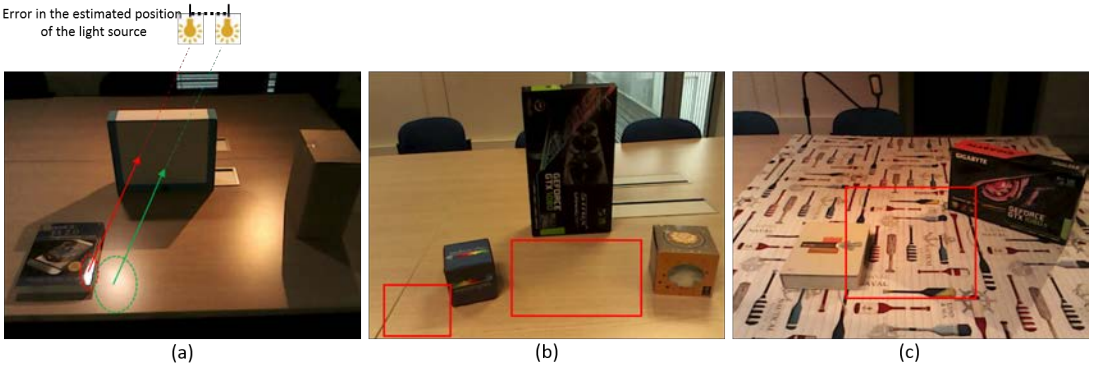


Figure 6.3 – (a) Resulting error within the position of the light source due to an inaccurate detection of specular effects. (b,c) Scenarios of weak cast shadows where applying a simple Canny edge filter does not deliver accurate results.

The second method proposed by [Li et al., 2003] considers the critical case of textured surfaces. The approach determines the expected positions of shadow edges and specularities for hypothetical lighting directions sampled from a hemisphere. After computing the expected positions of these cues for a given light direction, the method then checks whether these cues are present in the image at the predicted locations. For instance, shadow boundary points in the image are computed by a Canny edge filter and are compared to the generated hypothetical shadows using Euclidean distance. Beside the fact that this method considers the illumination within indoor real scenes to be distant and only recovers its direction, the information derived from the cast shadows is not robust. To illustrate, in figure 6.3-b, due to the presence of soft shadows, the thresholds used within the Canny edge detector must be lowered to detect smooth discontinuities. However, this results in noisy contours, especially in case of low-quality images (the camera's sensor noise can be detected as well). In presence of challenging textures and weak shadows (Figure 6.3-c), the detection of cast shadows is even more complex.

In this chapter, we propose a method which addresses these critical scenarios and robustly incorporates both shadow and specularity cues within a photometric registration framework. Specifically, we consider indoor real scenes composed of one or more objects with arbitrary shapes. Most importantly, scene surfaces can hold *arbitrary* textures and retain Lambertian and/or specular properties. Also, multiple light sources are handled and their respective locations can be freely changed by the user overtime. The proposed approach jointly exploits specular reflections and cast shadows to estimate the specular reflectance of scene surfaces and illumination characterisitcs (number of light sources, their respective 3D positions and colors). To summarize, the

main contributions of this chapter are:

- Detection of specular reflections and challenging cast shadows (e.g., weak shadows, overlapping shadows with different intensities) on arbitrary textured surfaces from a coarse 3D model and color images of the scene.
- Estimation of the 3D position of static and/or dynamic light sources for each incoming frame by robustly exploiting the information brought by specularities and shadows.
- Estimation of the color of recovered light sources.
- Estimation of the specular component of scene surfaces, namely the specular reflectance and the shininess coefficient.
- Near real-time implementation of the proposed method in order to meet MR scenario requirements.

The remainder of this chapter is organized as follows: we first present the inputs, assumptions and an overview of the proposed approach. Then, we describe the procedure of jointly exploiting specular reflections and cast shadows to estimate the reflectance and illumination of the scene. Finally, experimental results are presented and discussed along with two applications: realistic mixed reality and retexturing.

6.2 Our Proposed Method

Similarly to the approach described in chapter 5, the proposed method takes three inputs: (a) coarse 3D model of the scene acquired with an RGB-D sensor (Intel R200). (b) a color image of the scene captured under a *near-ambient* lighting which can be produced by considering a fairly uniform indirect lighting. This color image mainly contains the color and texture of scene surfaces and does not exhibit any shadowing or specular effects. We will refer to this image as the *reference* image ($\mathbf{I}_{ref}$). (c) color images of the scene from which illumination will be recovered (Figure 6.4). In comparison to the assumptions made within the previous approach (chapter 5), in this chapter we only retain the assumption of having a main planar surface on which arbitrary-shaped objects cast shadows. Consequently, the color of a scene point p is described using Phong model [Phong, 1975] as a combination of three components:

$$\mathbf{I}^p = \mathbf{I}^p_a + \mathbf{I}^p_d + \mathbf{I}^p_s \tag{6.1}$$

where $\mathbf{I}^p$, $\mathbf{I}^p_a$, $\mathbf{I}^p_d$ and $\mathbf{I}^p_s$ are respectively the color, the ambient, the diffuse and the specular components of point p. Using Phong model [Phong, 1975], ambient, diffuse and specular components in equation 6.1 are described as follows:

$$\mathbf{I}^p = \mathbf{k}^p_d \mathbf{L}_a + \mathbf{k}^p_d (\sum_{i=1}^{M} (\mathbf{n}^p \cdot \omega^p_i) \mathbf{L}_i \mathrm{O}^p_i) + \mathbf{k}^p_s \sum_{i=1}^{M} (\mathbf{r}^p_i \cdot \mathbf{v}^p)^{\alpha_p} \mathbf{L}_i \mathrm{O}^p_i \tag{6.2}$$

where $\mathbf{L}_a$, $\mathbf{L}_i$ are respectively the color vectors of ambient and light source i. $\mathbf{k}^p_d$ and $\mathbf{k}^p_s$ are respectively the diffuse and specular reflectances of point p, $\mathbf{n}^p$ is its normal vector,

$\mathbf{v}^p$ is its viewpoint vector, and α_p is its shininess parameter. $\mathbf{r}_i^p$ is the ideal reflection vector at point p with regard to light source i and ω_i^p is the direction of the light source i from point p. M is the number of light sources present in the scene. O_i^p is a binary visibility term that is equal to 1 if light i is visible from the 3D point p and equal to 0 if occluded.

Figure 6.4 – (a) Acquired 3D model of the scene using the Intel R200 sensor. (b) *reference* image ($\mathbf{I}_{ref}$) of the scene. One can notice the presence of mainly the surface's color/texture and hardly any shadowing or specular effects. (c) color image of the captured scene.

Given the color of a scene point p, its normal vector $\mathbf{n}^p$ and its current view vector $\mathbf{v}^p$ (all acquired or derived from the sensor's data), our goal is to estimate its reflectance properties ($\mathbf{k}_d^p$, $\mathbf{k}_s^p$, α_p) and the illumination in the scene (M, $\mathbf{L}_a$, $\mathbf{L}_i$, $\mathbf{r}_i^p$, ω_i^p, O_i^p) under which the current color image is captured. In order to robustly resolve this ill-posed problem, we jointly use the information brought by specular reflections and cast shadows as follows (Figure 6.5): using the *reference* image ($\mathbf{I}_{ref}$), we separate, for each incoming color image $\mathbf{I}$, surface texture from illumination effects (e.g., shading, shadowing and specular reflections). This step results in an image of the scene which mainly contains illumination-dependent variations and we will refer to it as the *illumination map*. From this *illumination map*, we detect the specular reflections and use them to estimate a rough direction of the light sources in the scene. Then, using detected specular effects and recovered lights directions, an adaptive and robust cast shadow detection is achieved and light sources positions and colors are estimated. Finally, using recovered scene illumination, specular reflectance parameters are estimated for scene points.

In the following sections, we describe in detail the main components of the proposed photometric registration approach.

6.2.1 Per-frame Texture Removal

In this section, our goal is to accurately separate texture/color variations from illumination-dependent effects such as shading, shadowing and specular reflections, within scene surfaces which can retain diffuse and/or specular properties. This is of interest for our photometric registration framework for two main reasons: (i) As previously stated, our approach uses both specular reflections and cast shadows to recover the reflectance

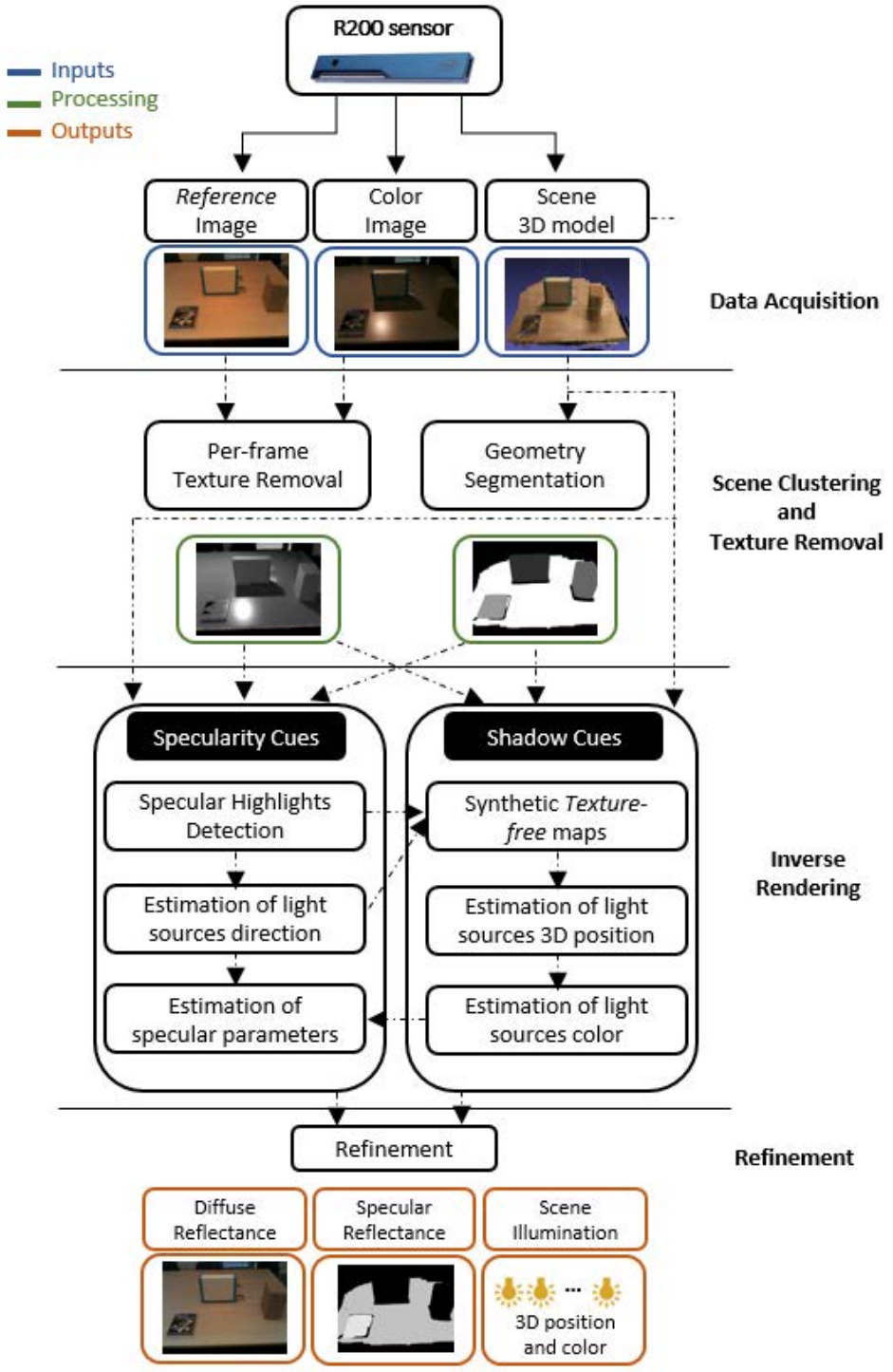


Figure 6.5 – Outline of the proposed photometric registration approach which jointly uses specular reflections and cast shadows to estimate the reflectance and illumination of the captured scene.

and illumination of the scene. Hence, to robustly detect and model these cues, it is important to isolate their effects from any color or texture changes. For instance, if one considers the main planar surface of the scene (Figure 6.6), the shading ($\mathbf{n}^p \cdot \omega_i^p$) smoothly varies (green box). Hence, if the texture variations (blue box) are accurately removed from the image, remaining abrupt and harsh discontinuities can be primarily associated to cast shadows (red box). (ii) The task of separating texture/color from illumination-dependent effects, also referred to as *intrinsic image decomposition* (Chapter 3), is a key step for many computer vision applications, especially retexturing. In fact, the core idea of this application consists in preserving illumination while partially or completely altering the texture of the scene. It is evidently important to accurately isolate both of these components.

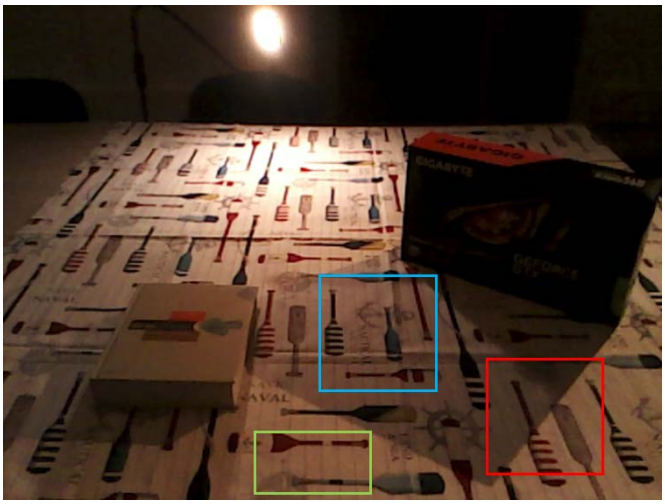


Figure 6.6 – A key step within our photometric registration approach consists in separating texture variation (blue box) from illumination-dependent variations such as shading (green box) and shadowing (red box)

The proposed approach for texture and illumination separation is two-fold. To begin with, we are interested in recovering the diffuse reflectance, corresponding to the intrinsic color/texture, for all scene points. Let us consider a pixel p within the *reference* image ($\mathbf{I}_{ref}$) shown in figure 6.7-a. The color of pixel p, corresponding to a 3D point in the scene, is described using Phong model [Phong, 1975] as:

$$\mathbf{I}^p_{ref} = \mathbf{k}^p_d \mathbf{L'}_a \tag{6.3}$$

where $\mathbf{I}^p_{ref}$ and $\mathbf{L'}_a$ are respectively the color vectors of point p and *near-ambient* lighting (used to produce the *reference* color image). We are interested in recovering the texture/color of the scene which is independent from the lighting conditions under which the image is captured. To achieve this task, the unknown color of the *near-ambient* lighting $\mathbf{L'}_a$ must be estimated. In this work, we use available constant-color regions within the scene (red boxes in figure 6.7-(a,b)) to estimate this unknown as follows:

$$\mathbf{L'}_a = \frac{\sum_{p \in W} \mathbf{I}^p_{ref}}{\#W} \tag{6.4}$$

where W is a constant-color (grayscale) region within the *reference* image and $\#W$ is the pixels count within W. If no such region is available, one can use a color constancy algorithm, such as [Thai et al., 2017] to estimate $\mathbf{L'}_a$. Consequently, using equation 6.3, the albedo/texture $\mathbf{k}_d^p$ of scene points (Figure 6.8-c) can be recovered within the scene as follows:

$$\mathbf{k}_d^p = \frac{\mathbf{I}_{ref}^p}{\mathbf{L'}_a} \tag{6.5}$$

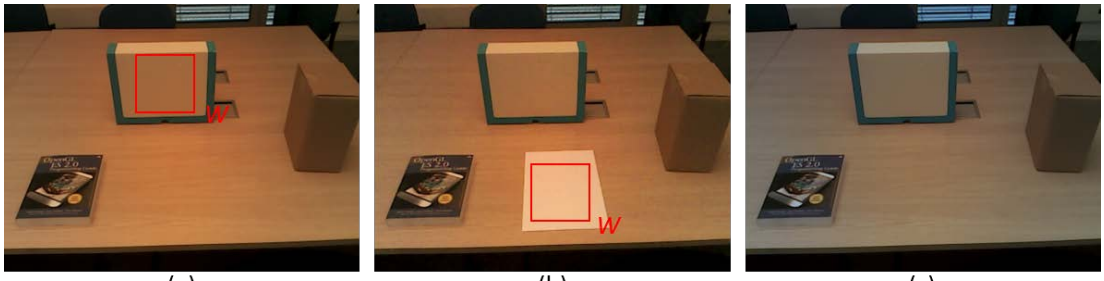


Figure 6.7 – (a,b) are examples of *reference* images of the scene with constant-color (e.g., white) regions (W) from which the color of lighting can be estimated using equation 6.4. (c) intrinsic texture/color of the scene which is independent from the lighting conditions under which it was produced (a).

Now that we have recovered the diffuse reflectance $\mathbf{k}_d^p$ of scene points, the second step of the proposed texture/illumination separation consists in recovering an *illumination map* δ for each incoming color image. This map must be texture-free and contain mainly shading, shadowing and specular reflections. The map δ is recovered, for each point p, using the diffuse reflectance estimate as follows:

$$\delta^p = \frac{\mathbf{I}^p}{\mathbf{k}_d^p} = \frac{\mathbf{k}_d^p(\mathbf{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathbf{L}_i \mathrm{O}_i^p) + \mathbf{k}_s^p \sum_{i=1}^{M}(\mathbf{r}_i^p \cdot \mathbf{v}^p)^{\alpha_p}\mathbf{L}_i \mathrm{O}_i^p}{\mathbf{k}_d^p} \tag{6.6}$$

Within equation 6.6, $\mathbf{k}_d^p$ cancels out with regard to the ambient and diffuse components. Also, since specular reflections do not cover significantly large regions of the image, and for clarity reasons, the equation 6.6 is rewritten as follows:

$$\delta^p = \mathbf{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathbf{L}_i \mathrm{O}_i^p + \epsilon_S^p \tag{6.7}$$

where δ^p represents the *illumination* map value at pixel p and ϵ_S^p corresponds to present specular highlights (Figure 6.8-b) at pixel p. As depicted in figure 6.8-b, δ contains mainly shading effects (green boxes) which correspond to the scalar product $(\mathbf{n}^p \cdot \omega_i^p)$ in equation 6.7, cast shadows (blue boxes) results from the occlusion term O_i^p and specular effects (red box) correspond to the term ϵ_S^p. One can notice that in case of a Lambertian scene ($\mathbf{k}_s^p = 0$), the ϵ_S^p term is equal to 0 and no specular effects are recovered (Second row in figure 6.8-b).

In the following sections, this recovered *illumination* map δ is used to detect and model both specular reflections and cast shadows within the scene.

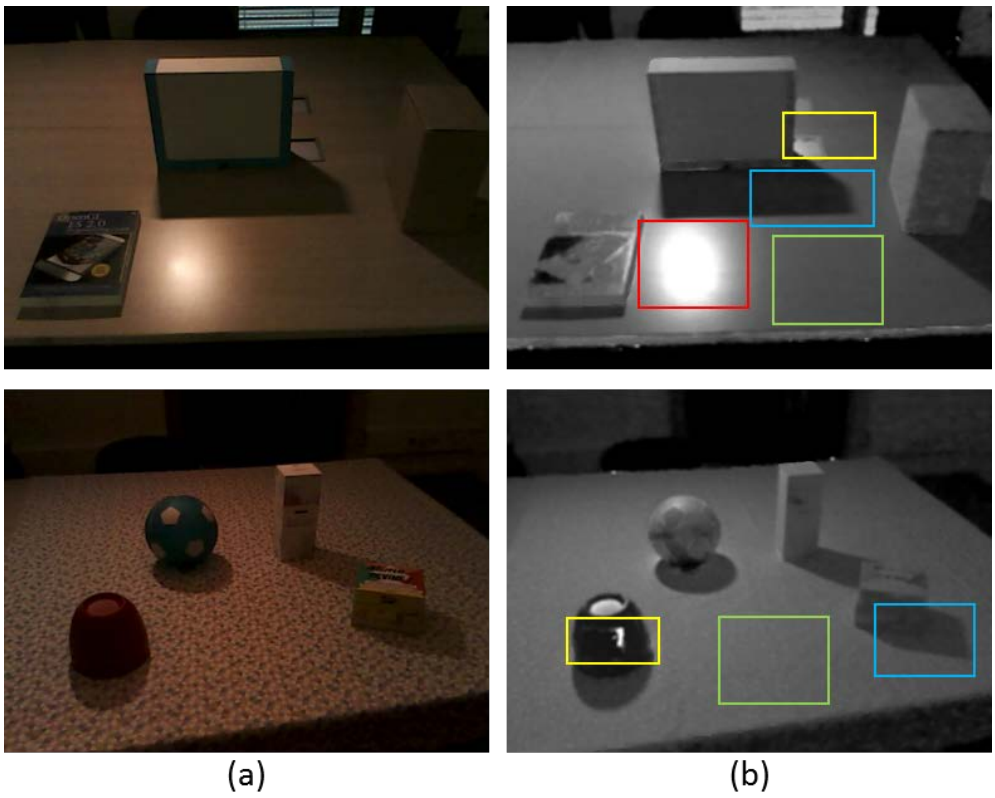


Figure 6.8 – (a) Input color image of captured scenes. (b) Recovered *illumination* maps for images in (a). Red, green, and blue boxes correspond respectively to specular, shading and shadowing variations in the scene. Yellow box are examples of noisy estimations due to low quality images or very shiny surfaces.

6.2.2 Specular Highlights Detection for Lights Direction Estimation

Specular reflections represent view-dependent cues which are informative about the direction of the light source in the scene. In fact, these cues are observed when the camera or the user's view direction is roughly aligned with the ideal specular reflection. In this section, our goal is to detect specularities within the recovered *illumination* map and use them to estimate the light sources direction.

To begin with, we detect specular highlights using [Ortiz and Torres, 2006]. By considering the Hue Saturation Value (HSV) color space, the approach recovers specular reflections at pixels where the color has high value (V) but low saturation (S). The value (V) corresponds to the maximum within the three-channel color vector (R,G,B) and the saturation (S) component is computed as follows:

$$S = \begin{cases} \frac{V - min(R,G,B)}{V}, & \text{if } V \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.8}$$

The chosen thresholds for minimum value and maximum saturation are respectively 0.8 and 0.2. The results are written into a binary mask (H) with 1 where the specularity is

detected and 0 otherwise. Because white surfaces may be misinterpreted as highlights, we make use of the *reference* image to improve our detection as follows:

$$\mathrm{H}^p = \begin{cases} 1, & \text{if } I^p_{ref} < I^p + \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{6.9}$$

where ϵ is a threshold of color intensity (linear combination of the three color channels) difference between the reference and current frames at p. In fact, since the *reference* image I_{ref} mainly contains the intrinsic color of scene surfaces, when specular effects are captured within the current frame, point p holds a significantly more important brightness value in I^p than in I^p_{ref}. Both detected specularities and discarded bright regions are shown in figure 6.9.

Figure 6.9 – (a) Input color image of the scene with two specular effects. (b) Detection of specular reflections: red pixels correspond to the detected specularities after discarding (magenta) pixels that do not check equation 6.9.

The second step consists in recovering the direction of light sources using the detected specularities (retrieved in the binary mask H). First, due to thresholding noise, small and/or isolated highlights can be detected. We handle these noisy regions using a simple blob detector: localy connected regions are initially recovered using [Teh and Chin, 1989] (Figure 6.10-a) and, each connected region is referred to as *blob* (Binary Large OBject). Blobs with significantly low points count are discarded (yellow ellipse) and the center of each kept blob is computed along with its euclidean distance with regard to other groups. Close blobs are merged to form one specular effect (green and red ellipses near the specular book) and its center and radius are computed.

Finally, for each detected specularity, the ideal specular reflection direction $\mathbf{r}^c$, at the center c of each blob, is recovered as roughly aligned with the view direction vector $\mathbf{v}^c$ (computed using camera pose and coordinates of the corresponding 3D point to pixel c). Consequently, an initial estimate of light sources direction, at the center of each blob c, is recovered as follows:

$$\omega_{c_k} = 2.(\mathbf{r}^{c_k} \cdot \mathbf{n}^{c_k}) \cdot \mathbf{n}^{c_k} - \mathbf{r}^{c_k} \tag{6.10}$$

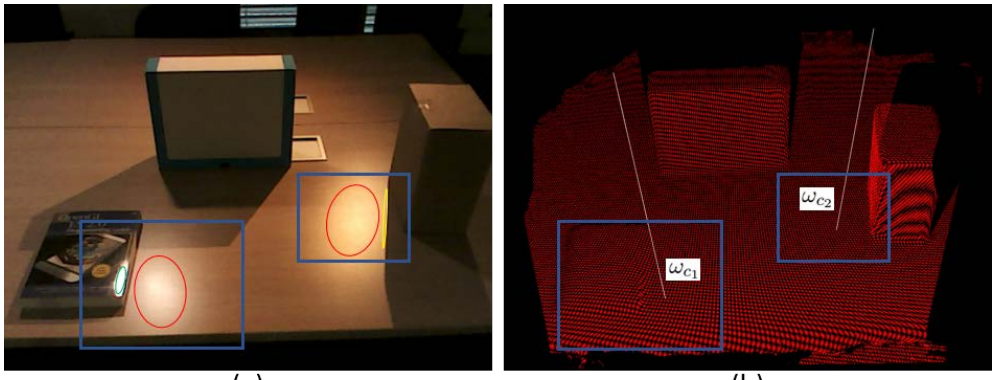


Figure 6.10 – (a) Detected specularities: blobs with a small pixels count (yellow ellipse) are discarded. Spatially close detected regions are merged to form a single detected specularity (green ellipse with regard to the near red ellipse). (b) Recovered light sources directions using the recovered specular effects (blue boxes) in (a).

where k iterates over the number of recovered blobs, ω_{c_k} is the light direction vector estimated using the k^{ith} blob, $\mathbf{r}^{c_k}$ is its ideal specular reflection vector and $\mathbf{n}^{c_k}$ is its normal vector. In figure 6.10-b, the scene is represented by a point cloud (red dots) and the recovered light directions are reported as white lines. In this scenario, the number of recovered blobs is equal to the number of light sources in the scene. Nonetheless, in case of recovering a number of blobs which is different from the light sources count (non-merged groups, noisy detection), we consider two iterators: k iterates over the recovered directions using specularities and i over the actual light sources present in the scene.

Unlike [Anusorn and Nopporn, 2016], we do not use recovered lights directions as search lines for their positions since it does not always deliver robust estimates (section 6.1). Our approach is described in the next section.

6.2.3 Cast Shadows Analysis for Lights Position Estimation

In this section, our goal is to efficiently incorporate the information brought by specular effects within an analysis approach of cast shadows in order to robustly estimate the position of light sources in the scene. This goal is achieved by considering two key steps: (i) the lighting in the scene is approximated by a set of equally distributed point lights (S_0). By using recovered lights directions in section 6.2.2, we are able to consider a small set within (S_0) which is more likely to contain actual real light sources. (ii) Using detected specular reflections along with the recovered *illumination* map δ (section 6.2.1), we robustly recover the 3D position of light sources in the scene. The core idea consists in an iterative matching procedure between δ and a set of synthetic *illumination* maps $\tilde{\delta}$ proper to the hypothetical point lights. The approach to recover the light sources position is achieved in a three-pass procedure and described in detail in the rest of this section.

The first step consists in recovering a set (S) of hypothetical point lights among which actual light sources will be identified later on. Within this task, we initially approximate the lighting in the scene by a set (S_0) of point lights equally distributed in the scene (Figure 6.11). Then, for each recovered light direction ω_{c_k} (section 6.2.2), we define a cone originating from the detected specularity's center c_k and oriented using ω_{c_k}. Finally, point lights located within the cone's volume constitute the set (S) as illustrated in figure 6.11. Beside the fact that, in comparison to the approach in [Anusorn and Nopporn, 2016], this method takes into account the inaccuracies which might exist within the roughly estimated lights direction, it also allows us to consider a smaller point lights set than the initial one (S_0). This is of interest for MR scenarios where the processing time requirements must be considered as well. For instance, from an initial set (S_0) counting 1176 point lights, only 352 are comprised within the set (S) using a cone-angle β of 10°.

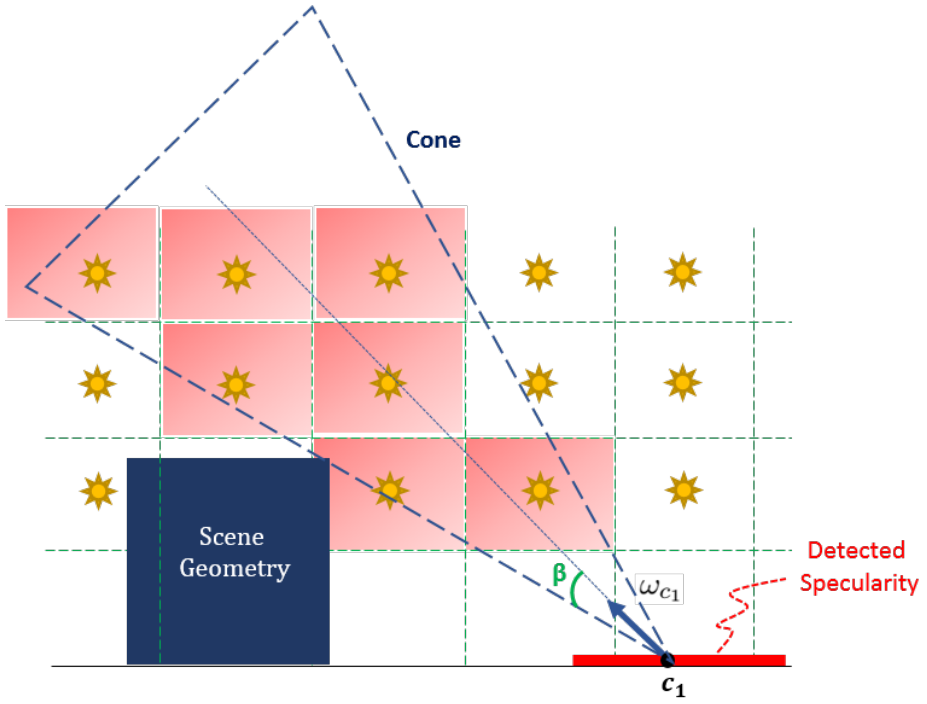


Figure 6.11 – Definition of a set of hypothetical point lights from which actual real light sources are recovered.

The second step consists in producing a synthetic *illumination* map $\tilde{\delta}$ for every point light in the set (S). Let us first consider equation 6.7, previously described to define the real *illumination map* δ, in presence of one point light:

$$\delta_i^p = \mathrm{L}_a + (\mathbf{n}^p \cdot \omega_i^p)\mathrm{L}_i\mathrm{O}_i^p + \epsilon_S^p \tag{6.11}$$

where δ_i^p represents the *illumination map* value at pixel p and ϵ_S^p corresponds to present specular highlight at pixel p. The scalar product $(\mathbf{n}^p \cdot \omega_i^p)$ represents the scene's shading and cast shadows result from the occlusion term O_i^p with regard to light source i. When a point p is occluded with regard to point light i, its *illumination map* value δ_i^p is equal

to L_a and it is located within a shadowed region ($O_i^p = 0$). On the contrary, when a point p is not occluded with regard to light i, its δ_i^p value is defined by equation 6.11 and it does not belong to a cast shadow ($O_i^p = 1$). By isolating the cast shadows effect, equation 6.11 can be rewritten as follows:

$$\begin{cases} \delta_i^p = L_a, & \text{if p is occluded} \\ \delta_i^p = L_a + (\mathbf{n}^p \cdot \omega_i^p) L_i + \epsilon_S^p, & \text{otherwise} \end{cases} \tag{6.12}$$

In order to render a synthetic *illumination map* $\tilde{\delta}$, we must know all the parameters present in equation 6.12. However, since lighting characteristics (L_a, L_i, ω_i^p) are not known in our case (our goal is to estimate them), we recover rough estimates of these parameters as follows:

$$\begin{cases} L_a = L_o, & \text{if p is occluded} \\ (L_a + (\mathbf{n}^p \cdot \omega_i^p) L_i) + \epsilon_S^p = L_v + \sigma S_i^p, & \text{otherwise} \end{cases} \tag{6.13}$$

where:

- σ is a binary term, equal to 1 if specular reflections are present and 0 otherwise. Specifically, specular reflections are considered to be present if the detected pixels count within the recovered specularity mask (H) in section 6.2.2 is not null.

- S_i^p is a synthetic specular map rendered, for a point light i, using Phong model [Phong, 1975]:

  $$S_i^p = \mathbf{k}_s^p (\mathbf{r}_i^p \cdot \mathbf{v}^p)^{\alpha_p} L_i O_i^p \tag{6.14}$$

  Because the 3D sensor (R200) delivers a coarse geometry of the scene, the rendering of S_i^p is achieved as follows: to begin with, the 3D model of the scene is clustered using the method described in chapter 5.2.1 where the main planar surface is detected along with 3D objects lying on it (Figure 6.12-a). The rendering of the specular map S_i^p is then limited within the main planar surface, which is substituted by a perfect plane, in order to take account of the geometry's inaccuracies. To illustrate, in figure 6.12-(b,c), one can notice the effects of noisy geometric data in comparison with a perfectly modeled planar surface. The specular parameters value used to render the synthetic specular maps are respectively 1.0, 1.0 and 0.9 for $\mathbf{k}_s^p$, L_i and α_p.

- The terms L_v and L_o correspond respectively to the overall brightness in non-occluded/visible and occluded regions. Their computation is achieved in a three-step procedure: (i) for each clustered object, we define a proportional region of interested (ROI) recovered as the intersection of a sphere comprising the 3D object and the detected plane (Figure 6.13). (ii) *illumination map* values δ^p, within the ROI, are increasingly sorted out and, L_o is recovered as the value at 25%. In fact, since these regions represent potential shadowed regions, the underlying assumption corresponds to having at least 25% of the ROI within a shadowed region. (iii) L_v is recovered as the mean of illumination values of pixels outside the ROI and detected highlights map H.

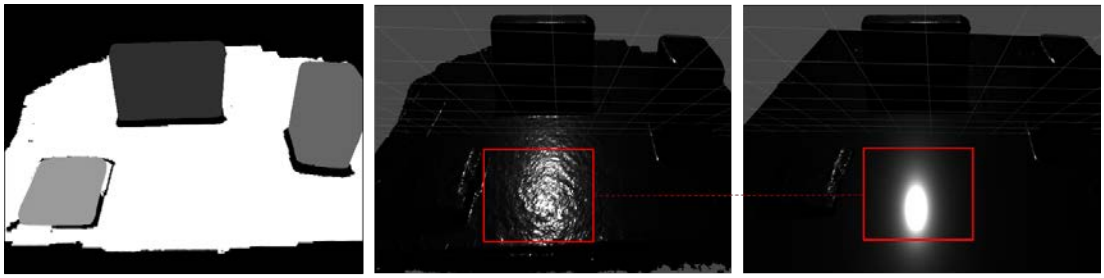


Figure 6.12 – (a) Segmented 3D model of the scene: white pixels correspond to the detected plane, grayscale pixels represent 3D objects and black pixels represent background or noisy data. (b) Rendered specular map using the 3D model of the scene. (c) Rendered specular map using a perfect plane corresponding to the detected planar surface in the scene.

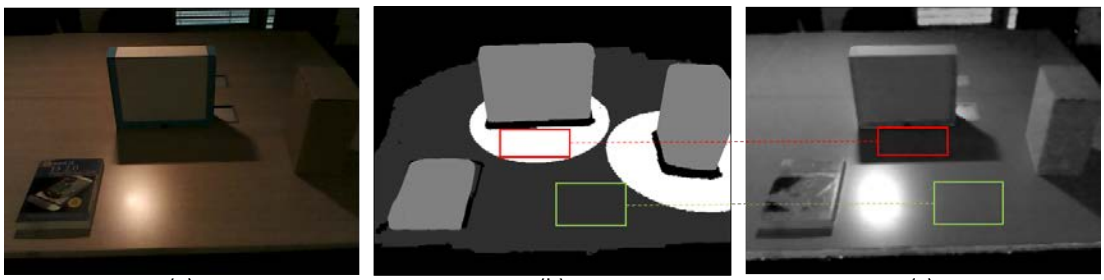


Figure 6.13 – (a) Color image of the scene. (b) Definition of the ROI: white pixels represent the ROI, grayscale pixels correspond to 3D objects in the scene and black pixels represent background or noisy data. (c) Recovered *Illumination map* with regard to (a).

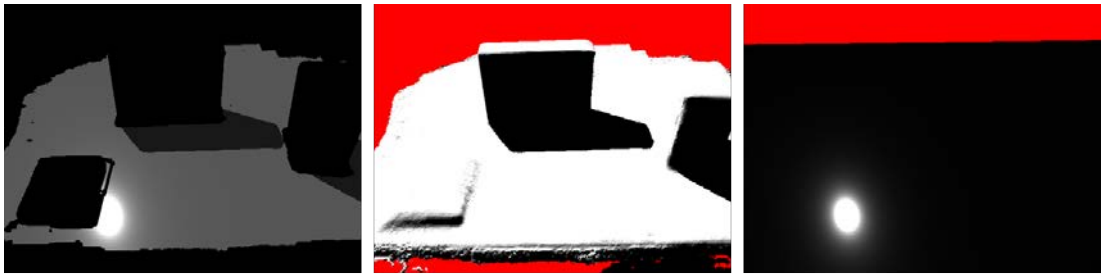
Finally, for every point light i in the set (S), the rendering of its synthetic illumination map $\tilde{\delta}_i$ (Figure 6.14-a) is achieved as follows:

$$\tilde{\delta}_i^p = \begin{cases} \mathrm{L}_o, & \text{if p is occluded} \\ \mathrm{L}_v + \sigma \mathrm{S}_i^p, & \text{otherwise} \end{cases} \tag{6.15}$$

In practice, since the geometry is static, occlusion O_i^p and specular maps S_i^p (Figure 6.14-(b,c)) are rendered only once. Only L_o, L_v and σ are evaluated for each incoming frame since lighting conditions may change over time.

The final step consists in identifying the actual real light sources within the subset (S). The identification is carried within an iterative process as follows:

- We initially compute correlation values by matching recovered *illumination map* δ and the rendered ones $\tilde{\delta}$. The light source whose synthetic *illumination map* has the best correlation value is selected.
- For each iteration, previously selected light sources are discarded. Also, previously matched pixels are not considered and point lights which are close to the



(a) (b) (c)

Figure 6.14 – (a) Example of a synthetic *illumination map* $\tilde{\delta}$ using rendered occlusion O_i^p (b) and specular S_i^p maps (c).

previously selected ones are discarded. Both L_o, L_v are re-considered to be able to take account of shadows with different intensities (Figure 6.15)

- The process ends either when the currently selected $\tilde{\delta}$ has a significantly low matching value or if the number of selected lights is higher than N. The chosen correlation corresponds to Pearson's correlation coefficient ranging between 0 and 1 and, N is set to 4.

(a) (b) (c)

Figure 6.15 – (a) Color image of the scene. (b,c) Cast shadows and specular effects corresponding to the first (b) and second recovered light sources (c).

The proposed approach in this section recovers the 3D position of light sources in the scene. In the following section, we estimate their respective colors.

6.2.4 Light Sources Color Estimation

Illuminating virtual objects within MR scenarios requires recovering the characteristics of light sources in the scene, namely the 3D position and three-channel color vector (R,G,B). In the previous section, we used both specular highlights and cast shadows to recover the number of light sources along with their 3D positions. In this section, our goal is to estimate their respective colors. To achieve this task, we consider both the recovered illumination map δ (Section 6.2.1) and subset (S) of point lights (Section 6.2.3). By considering equation 6.7 for a set of points p which belong to the main planar surface and are not detected as specular highlights (if the sensor saturates, their colors

are not accurate), we obtain:

$$\begin{cases} \mathbf{L}_a + \sum_{i=1}^{M} (\mathbf{n}^{p_1} \cdot \omega_i^{p_1}) \mathbf{L}_i \mathrm{O}_i^{p_1} = \delta^{p_1} \\ \mathbf{L}_a + \sum_{i=1}^{M} (\mathbf{n}^{p_1} \cdot \omega_i^{p_1}) \mathbf{L}_i \mathrm{O}_i^{p_2} = \delta^{p_2} \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \mathbf{L}_a + \sum_{i=1}^{M} (\mathbf{n}^{p_1} \cdot \omega_i^{p_1}) \mathbf{L}_i \mathrm{O}_i^{p_N} = \delta^{p_N} \end{cases} \quad \Rightarrow \quad \mathbf{AL} = \delta \tag{6.16}$$

where:

$$\mathbf{A} = \begin{pmatrix} 1 & (\mathbf{n}^{p_1} \cdot \omega_1^{p_1}) \mathbf{O}_1^{p_1} & \cdots & (\mathbf{n}^{p_1} \cdot \omega_M^{p_1}) \mathbf{O}_M^{p_1} \\ 1 & (\mathbf{n}^{p_2} \cdot \omega_1^{p_2}) \mathbf{O}_1^{p_2} & \cdots & (\mathbf{n}^{p_2} \cdot \omega_M^{p_2}) \mathbf{O}_M^{p_2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (\mathbf{n}^{p_N} \cdot \omega_1^{p_N}) \mathbf{O}_1^{p_N} & \cdots & (\mathbf{n}^{p_N} \cdot \omega_M^{p_N}) \mathbf{O}_M^{p_N} \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_a \\ \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_M \end{pmatrix}; \quad \delta = \begin{pmatrix} \delta_1^p \\ \delta_2^p \\ \vdots \\ \delta_N^p \end{pmatrix}$$

where M is the number of recovered light sources and N is the number of considered pixels within equation 6.16. The linear system 6.16 is solved, for each color channel, using iterative Least Squares with bounds and equality constraints:

$$\widehat{\mathbf{L}} = \arg\min_{\mathbf{L}} \left(\tfrac{1}{2} \left\| \mathbf{W}(\mathbf{AL} - \delta) \right\|^2\right) \quad \text{subject to:}$$

$$\begin{cases} 0 \leq \mathbf{L}_i \leq 1 \quad \text{and} \quad 0 \leq \mathbf{L}_a \leq 1 \\ \mathbf{L}_a + \sum_{i=1}^{M} \mathbf{L}_i = 1 \end{cases} \tag{6.17}$$

where $\mathbf{W}$ is a diagonal matrix with weights computed using Tukey's bisquare loss function. Small weights are discarded throughout iterations and we recover color vectors for each light source in the scene.

6.2.5 Scene Specular Reflectance Estimation

The estimation of scene reflectance is of interest for MR scenarios. For instance, if one considers adding a virtual light source in the real scene, real surfaces which retain a specular property must exhibit specular effects when viewed near the ideal specular reflection $\mathbf{r}$ of the virtual lighting. Hence, in order to correctly render the virtual specularity, we must recover the surface's specular parameters, namely the specular reflectance $\mathbf{k}_s^p$ and shininess coefficient α_p for points p in the scene. Our approach to estimate these parameters is two-fold:

In section 6.2.2, we recovered a mask H where detected specular reflections are retrieved. In the following, our goal is to estimate the specular reflection component

within detected specularities in H. By considering Phong model [Phong, 1975], the specular component $\mathbf{I}_s^p$ is described as follows:

$$\mathbf{I}_s^p = \mathbf{I}^p - \mathbf{I}_d^p = \mathbf{I}^p - \mathbf{k}_d^p(\mathbf{L}_a + \sum_{i=1}^{M}(\mathbf{n}^p \cdot \omega_i^p)\mathbf{L}_i \mathrm{O}_i^p) \tag{6.18}$$

where $\mathbf{I}^p$ is the color of point p within the input color image and ($\mathbf{k}_d^p$, $\mathbf{L}_a$, ω_i^p, $\mathbf{L}_i$, O_i^p) are all parameters which we have estimated. Examples of recovered specular components $\mathbf{I}_s^p$ are shown in figure 6.16-b.

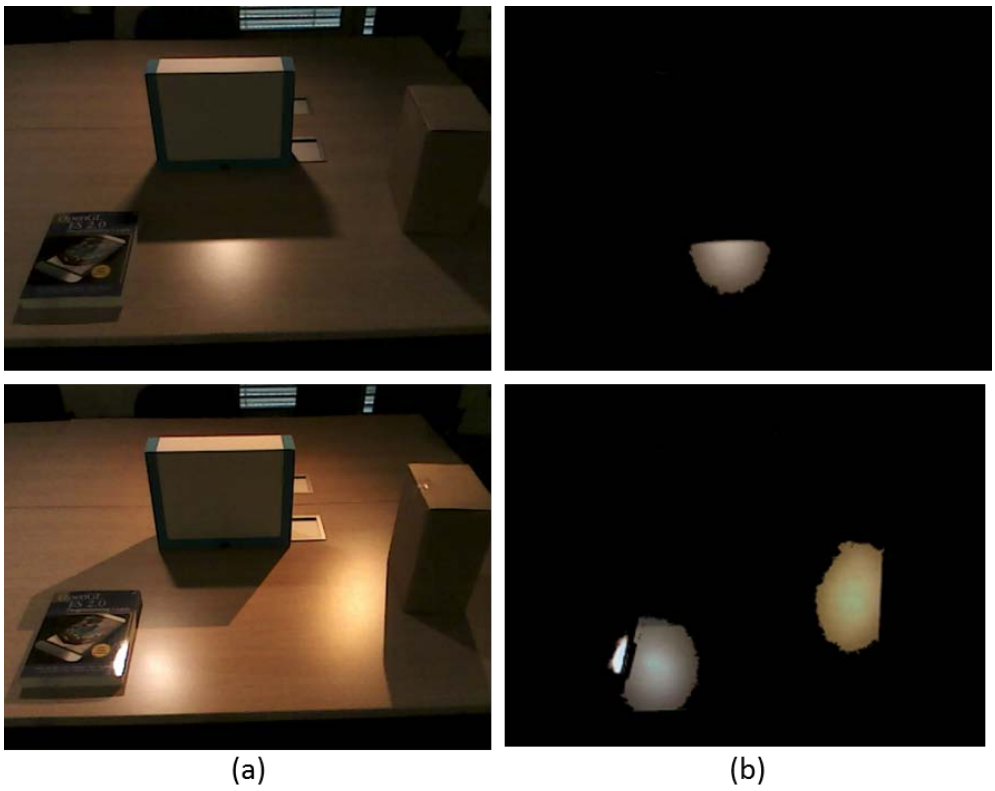


Figure 6.16 – (a) Input color images of the scene. (b) Recovered specular component for input images in (a).

In order to estimate the specular reflectance $\mathbf{k}_s^p$ of point p, we use Phong model [Phong, 1975] as follows:

$$\mathbf{k}_s^p = \frac{\mathbf{I}_s^p}{\sum_{i=1}^{M}(\mathbf{r}_i^p \cdot \mathbf{v}^p)^{\alpha_p}\mathbf{L}_i \mathrm{O}_i^p} \tag{6.19}$$

In fact, since specular reflections are viewed near the perfect specular reflection direction, vectors $\mathbf{r}_i^p$ and $\mathbf{v}^p$ are assumed to be roughly aligned. This assumption simplifies the denominator within equation 6.19 where the parameter α_p is unknown. Consequently, equation 6.19 can be rewritten as:

$$\mathbf{k}_s^p = \frac{\mathbf{I}_s^p}{\sum_{i=1}^{M}\mathbf{L}_i \mathrm{O}_i^p} \tag{6.20}$$

Furthermore, because specular reflections are often observed only within parts of the scene, and we aim at estimating $\mathbf{k}_s^p$ for all scene points, we assume that each 3D object in the scene retains a unique specular reflectance. Consequently, the specular reflectance is recovered, for each object, as the maximum value of recovered $\mathbf{k}_s^p$ within points p belonging to the same object (Figure 6.17).

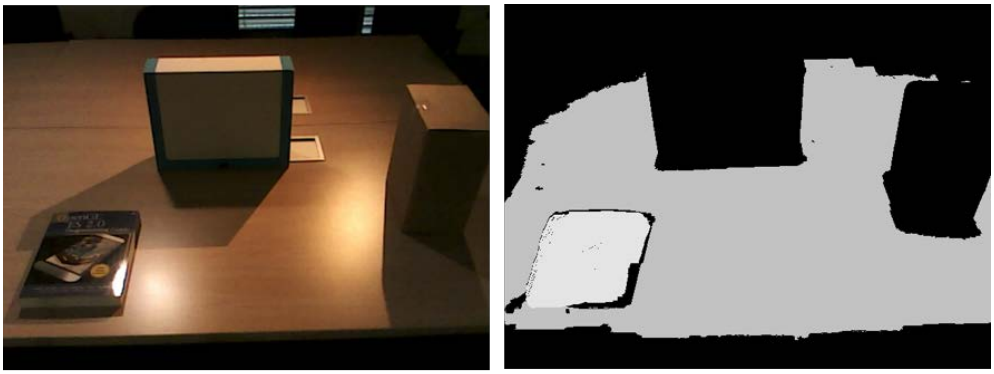


Figure 6.17 – (a) Input color image of the scene. (b) Estimated specular reflectance for each 3D object in the scene: the higher the brightness value, the more the surface is specular.

Finally, for each 3D object in the scene, the shininess parameter α is recovered using the following loss function:

$$F(\alpha, \mathbf{k}_s, \omega) = \sum_j (\mathbf{I}^j - \tilde{\mathbf{I}}^j(\alpha, \mathbf{k}_s, \omega))^2 \tag{6.21}$$

where j iterates over pixels that belong to the considered object/cluster, $\mathbf{I}$ is the input color image and $\tilde{\mathbf{I}}$ is a rendered color image using Phong model (equation 6.2). The optimization of function F is achieved using a Levenberg Marquardt algorithm where only the shininess coefficient α, specular reflectance $\mathbf{k}_s$ and light sources positions ω are varied by the solver. In fact, we refine $\mathbf{k}_s$ in order to take account for the approximation introduced in equation 6.20. Also, since our light sources are recovered from a discrete set of hypothetical point lights (section 6.2.3), a trade-off between fine sampling and real-time constraints must be considered. Hence, we initially define a coarse sampling (1176 point lights with a sampling step of 20cm) and refine the recovered positions using equation 6.21.

6.3 Experimental Results

A calibrated RGB-D sensor browses the scene with a fixed aperture, shutter speed and gain. Using the acquired 3D model, we recover for each incoming color image, the reflectance and illumination of the scene. The framework runs at an interactive frame rate of 4fps. In the following, we evaluate the accuracy of the proposed framework within both synthetic and real scenes.

Synthetic Data

We consider synthetic scenes where ground-truth reflectance and illumination are available for comparison with our estimates. The synthetic dataset is composed of six scenes ('SynS1'to 'SynS6') with various shapes of 3D objects located on a main planar surface (Figure 6.18) with various textures.

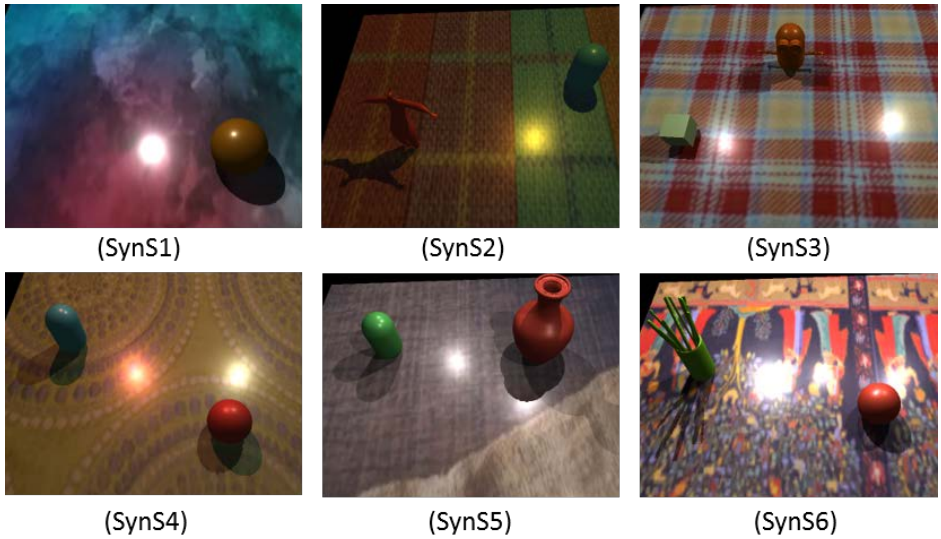


Figure 6.18 – Color image of six synthetic scenes ('SynS1'to 'SynS6') with various textures and geometries.

For each virtual 3D scene, the same inputs acquired or derived from the R200 sensor are rendered, namely color images of the scene and the *reference* image using an ambient lighting (Figure 6.19). With regard to the scene's lighting, we dispose of three point lights which we freely move and turn on/off in the scene. Furthermore, we consider different lighting color vectors (R,G,B) in order to evaluate the accuracy of our illumination estimation. The rendering is achieved in Unity engine [Unity, 2018] using Phong model [Phong, 1975]. In the following, we evaluate both illumination and reflectance using the Root Mean Square Error (RMSE) between the ground truth and estimated parameters.

Our first test consists in evaluating the position of light sources. Table 6.1 shows a comparison between ground-truth and recovered positions for the six virtual scenes presented in figure 6.18. The results demonstrate the robustness of our approach in presence of challenging textures and lighting conditions. For instance, for scenes SynS1 and SynS2 lit by one point light, the average RMSE within the 3D coordinates ($\mathbf{x}$,$\mathbf{y}$,$\mathbf{z}$) of the light source is respectively 0.026 for $\mathbf{x}$, 0.035 for $\mathbf{y}$ and 0.0254 for the $\mathbf{z}$ coordinate. Another interesting scenario is reported for SynS5: the scene is illuminated by three point lights which create overlapping cast shadows with different intensities. The approach recovers the correct number of light sources along with their 3D positions. Scene SynS6 is very challenging both in terms of texture and lighting: the cast shadows

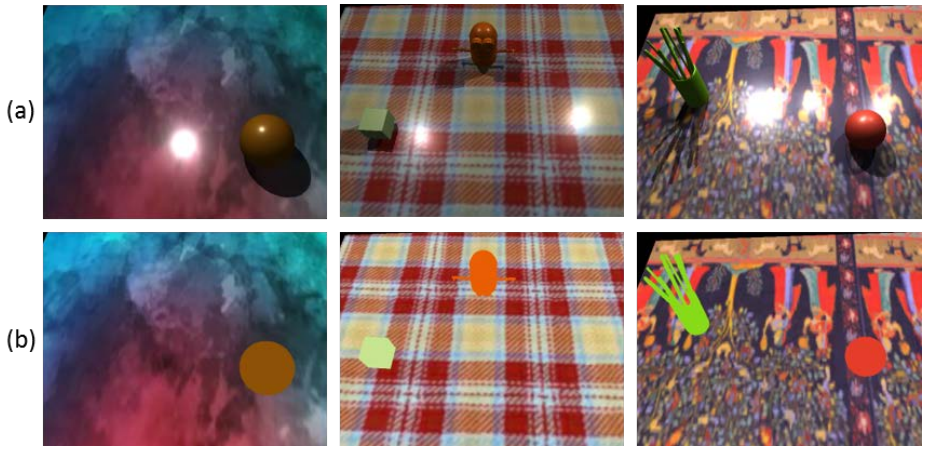


Figure 6.19 – Ground-truth images: (a) color image of scenes SynS1, SynS3 and SynS6. (b) *reference* image of scenes in (a).

are weak due to a strong lighting in the scene. Although the 3D position of the second and third recovered light sources is not as accurate in comparison with the other scenes, we recover the correct number and orientations of light sources by exploiting the observed specular reflections.

Scene	Ground truth position (x,y,z)			Estimated position (x,y,z)			Distance error
SynS1	-0.78	3.89	2.54	-0.75	3.85	2.57	0.058
SynS2	0.92	2.01	2.23	0.94	2.04	2.21	0.041
SynS3	-1.38	2.71	2.62	-1.37	2.66	2.69	0.086
	1.57	2.65	2.16	1.62	2.67	2.11	0.073
SynS4	1.01	1.59	2.98	0.98	1.51	3.04	0.104
	-0.84	1.95	2.79	-0.86	1.89	2.71	0.101
SynS5	0.54	2.91	-1.62	0.59	2.87	-1.69	0.094
	0.67	1.59	1.98	0.72	1.51	2.04	0.111
	-0.87	1.12	2.79	-0.92	1.07	2.71	0.106
SynS6	-1.12	2.26	2.29	-1.19	2.21	2.37	0.117
	-1.32	2.41	2.47	-1.47	2.72	2.65	0.388
	-1.48	2.54	2.40	-1.65	2.69	2.57	0.283

Table 6.1 – Comparison between ground truth and estimated 3D position of light sources in the virtual scenes.

The second evaluation concerns the color vector (R,G,B) of light sources as well as the specular reflectance of the scene. Within this task, we compare ground-truth images with rendered images using our estimates (Figure 6.20). The RMSE between input and rendered images for synthetic scenes lit by one point light (SynS2), two point lights (SynS5) and three point lights (SynS6) is reported in table 6.2.

We observe a general agreement between input and rendered views with a RMSE less than 5%. In scene SynS2, the color of lighting as well as the specular reflectance are

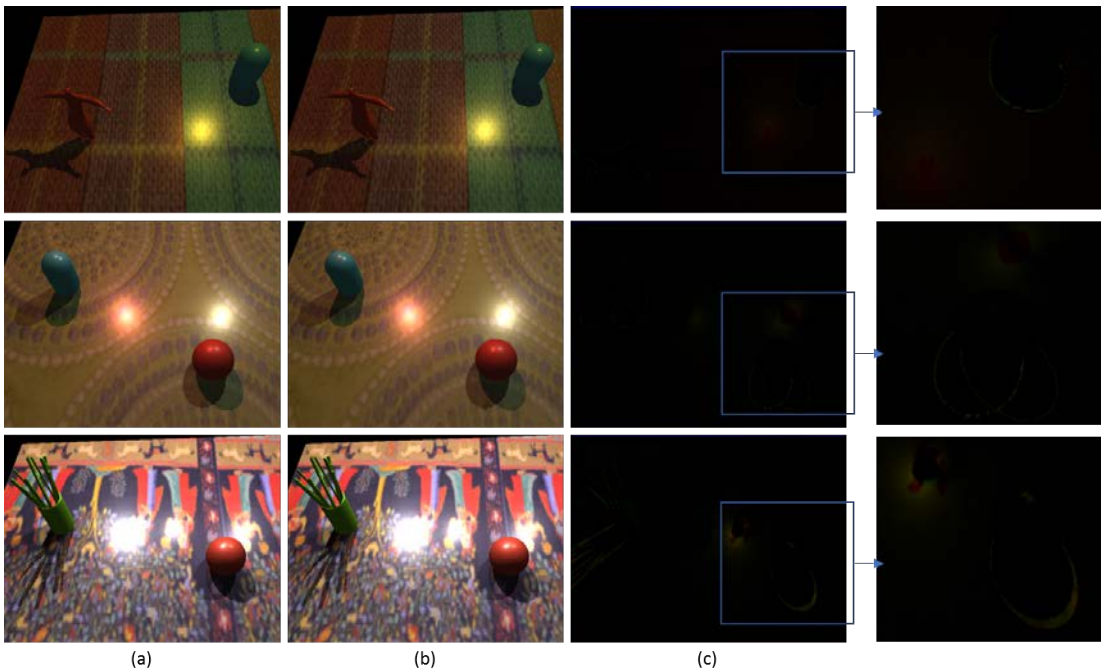


Figure 6.20 – Input color images of scene SynS2, SynS4 and SynS6. (b) Rendered images using our reflectance and illumination estimates. (c) RMSE of the difference between images in (a) and (b).

accurately recovered. The error is mainly present within the shadow contours resulting from the recovered 3D position of the light source. The most noticeable error is reported for scene SynS6 where the difference between cast shadows created by the third light source is significant due an error within its recovered position.

Scene	RMSE (%)
SynS2	2.79
SynS4	3.47
SynS6	4.54

Table 6.2 – RMSE of difference between ground truth and rendered virtual scenes using our reflectance and illumination estimates.

Real Data

In the following, we illustrate our results within a selection of five real scenes 'S1' to 'S5' grouped row-wise in figure 6.21. The considered scenes are composed of more than two objects located on a main planar surface. Both texture and reflectance properties vary within scene surfaces. For instance, scenes S1, S3 and S5 contain a planar surface with challenging textures. Also, scenes S2 and S4 exhibit specular reflections. Illumination-wise, scenes S1 and S2 are lit by one light source whereas scenes S3, S4 and S5 are lit by two light sources.

In figure 6.21-a, we present the captured *reference* images under a *near-ambient* lighting. These images are used to achieve the task of texture removal within color images from which illumination is recovered (Figure 6.21-b). This results in an *illumination* map δ which mainly contains shading, shadowing and specular effects. As illustrated in figure 6.21-c, our algorithm recovers accurate *illumination* maps where the intrinsic texture/albedo is accurately separated from illumination-dependent effects.

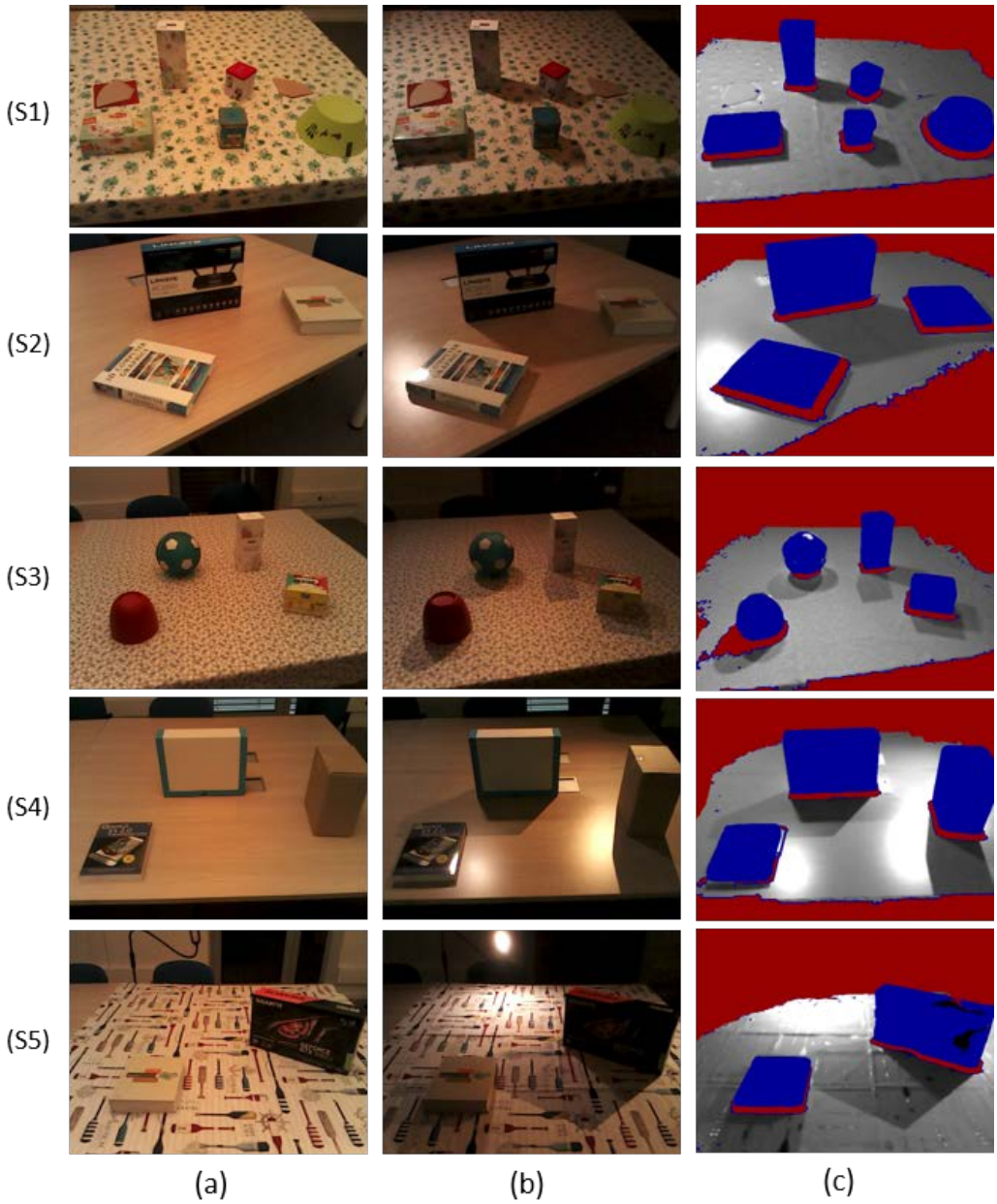


Figure 6.21 – (a) *Reference* image of the scene captured using an ambient lighting. (b) Input color image of the scene from which illumination is recovered. (c) Estimated *illumination maps* for uniform (S2,S4) and textured surfaces (S1,S3,S5): background/noise are represented by red pixels and occluding objects by blue pixels. Grayscale values correspond to the intensity of illumination values δ (Section 6.2.1).

Recovered *illumination* maps are considered to estimate the 3D position of light sources in the scene. This is achieved by initially using specular reflections, when available, to estimate a rough direction of light sources. Then, within a matching process, we recover the position of light sources represented by point lights. In figure 6.22-b, we show detected specular effects (cyan pixels) for the five real scenes. While the method delivers good results for scenes (S1, S2, S3 and S4), it erroneously detects a light source captured within the color image but since it is not part of the acquired 3D model, this detected area is discarded. Furthermore, in figure 6.22-c, we overlay the shadow maps of recovered light sources on the current color image. Our approach robustly recovers illumination in the scene in presence of specular effects (S2,S4) and challenging textures (S1,S3,S5). Furthermore, within (S2, S4), the detected specular reflections allow us to consider less than 30% of the initial hypothetical point lights within the cast-shadow analysis. The proposed method handles overlapping shadows (S3), weak cast shadows (green pixels in S4) as well as shadows which do not retain a uniform intensity due to a strong near lighting (S5). In fact, in contrast to the method proposed in chapter 5, scenarios such as (S4) could not be addressed before since the synthetic illumination map used within the matching procedure was a binary map (specular effects non taken into account within this map).

In order to evaluate the precision of recovered light sources positions, we used a telemeter to measure the distance from a chosen world coordinate system to the light sources in the scene. Results for the five real scenes are shown in table 6.3. Our algorithm recovers light sources position with an average error of 9cm for a mean distance of 1.62m to the light source and a standard deviation of 3.2cm.

Scene	Measured Distance (m)	Estimated Distance (m)	Distance error (m)
S1	1.83	1.92	0.09
S2	1.68	1.61	0.07
S3	1.74	1.81	0.07
	1.72	1.83	0.11
S4	1.44	1.52	0.08
	1.63	1.75	0.12
S5	1.12	1.19	0.07
	1.52	1.61	0.09

Table 6.3 – Comparison between measured and estimated distances using our proposed approach.

The goal of photometric registration algorithms consists in achieving realistic mixed reality scenarios. In figure 6.23, we show realistic augmentations of real scenes using our photometric estimates. For instance, for scenes (S1,S3,S5), virtual shadows cast by a sphere are consistent with real shadows in terms of shape and color. Furthermore, within scene S2, we show a correct occlusion of a real specularity by a virtual object. One can notice the reconstructed texture within the specular area.

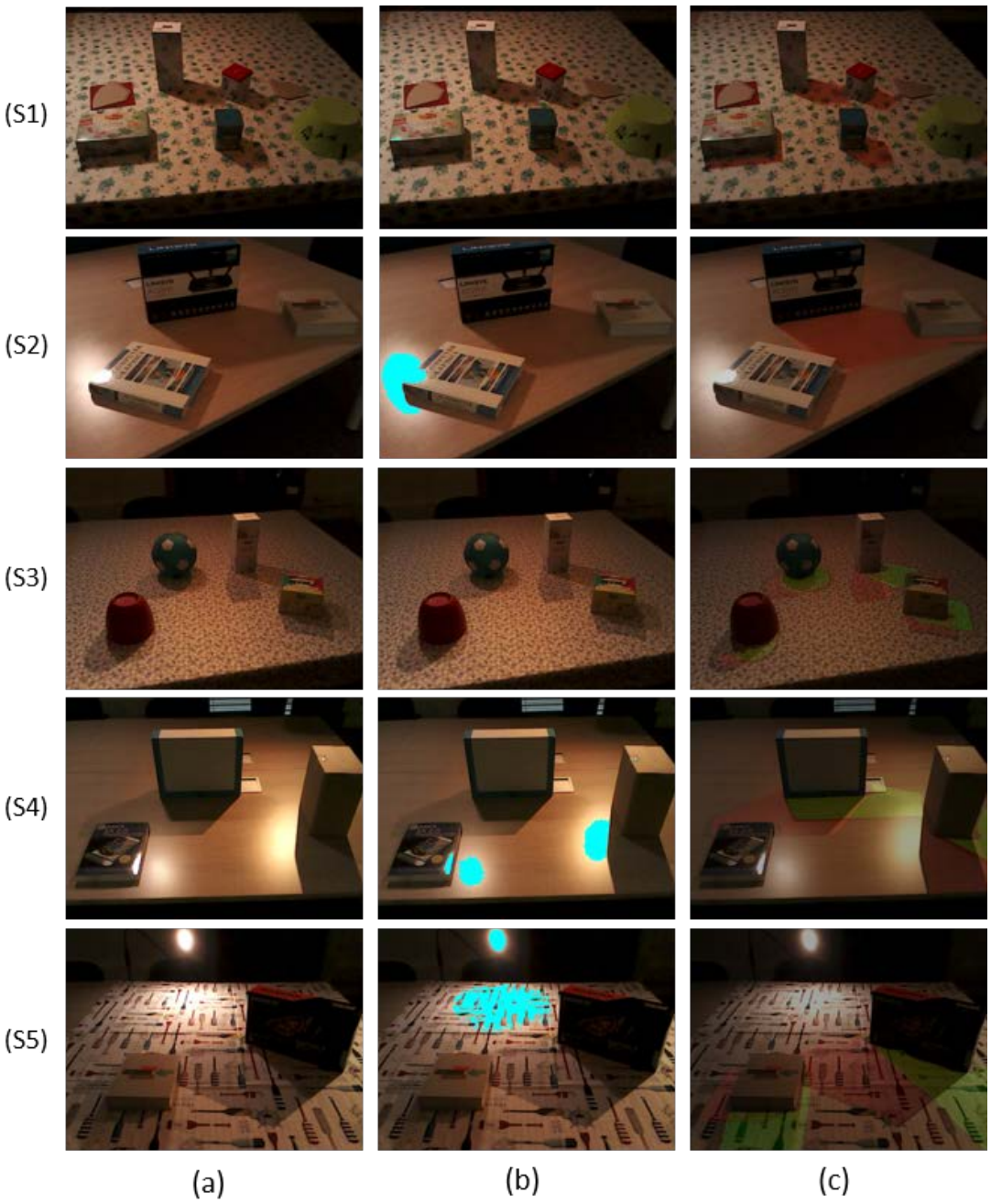


Figure 6.22 – (a) Color image of captured scenes. (b) Detected specular reflections are represented by cyan pixels. (c) Overlay of shadow maps corresponding to estimated light sources: first and second best matches are respectively represented by red and green pixels.

We further consider the scenario of retexturing the scene while preserving the current illumination. This is achieved in real-time and corresponds to the product of the *illumination* map δ and a target texture $\mathbf{T}$ (target diffuse reflectance):

$$\mathbf{I}^p_{retex} = \delta^p \mathbf{T}^p \tag{6.22}$$

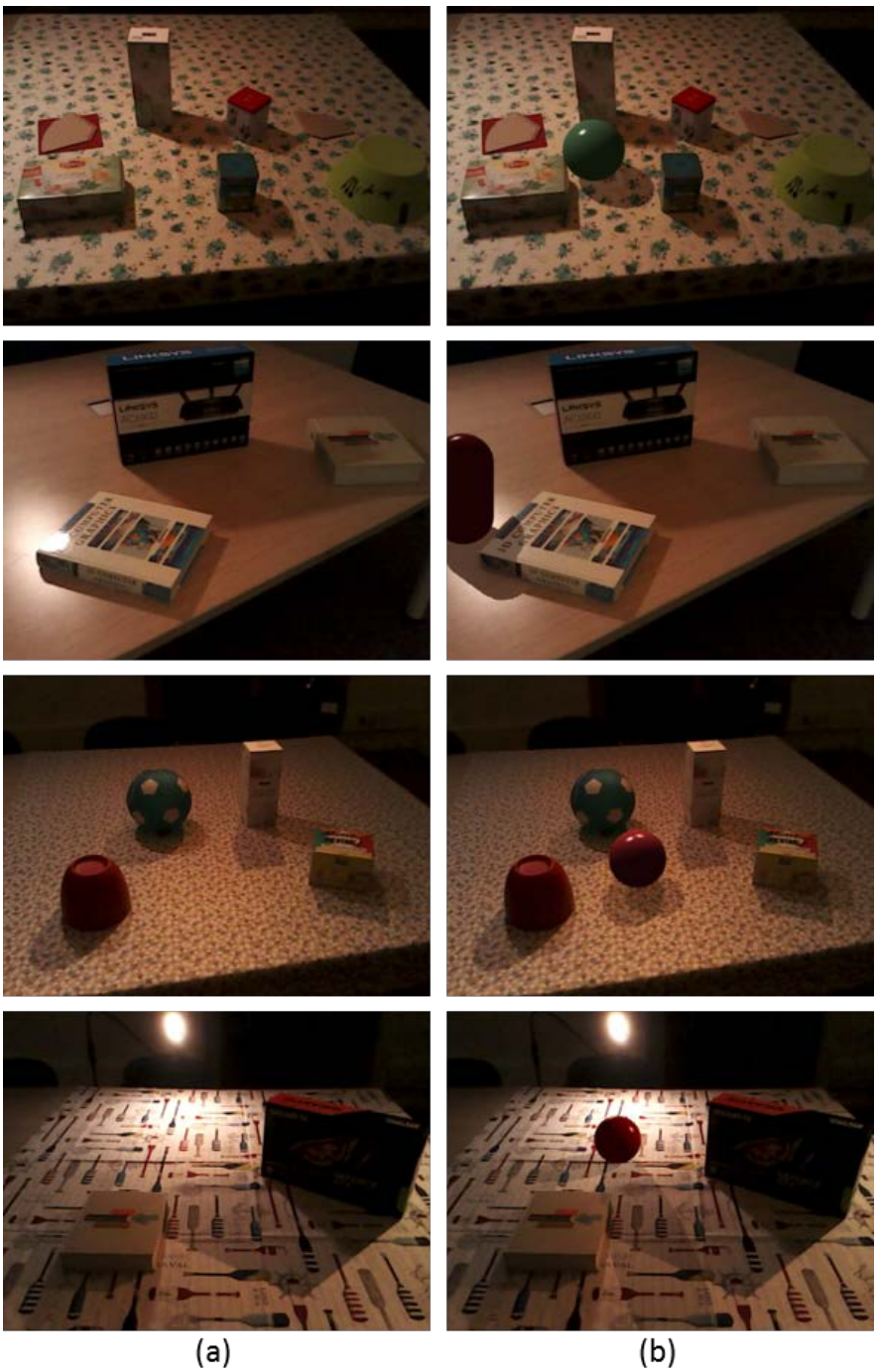


Figure 6.23 – Realistic augmentations of real scenes using our reflectance and illumination estimates. We demonstrate visually consistent virtual shadows in terms of shape and color with regard to real cast shadows. The second row is an example of a virtual object occluding a real specularity. One can notice a correct reconstruction of texture within the specular region.

where $\mathbf{I}^{p}_{retex}$ is the color of the re-textured scene and p corresponds to points which belong to the main planar surface in the example shown in figure 6.24. Inaccuracies

are mainly due to coarse or unavailable geometry (red pixels in figure 6.21-c).

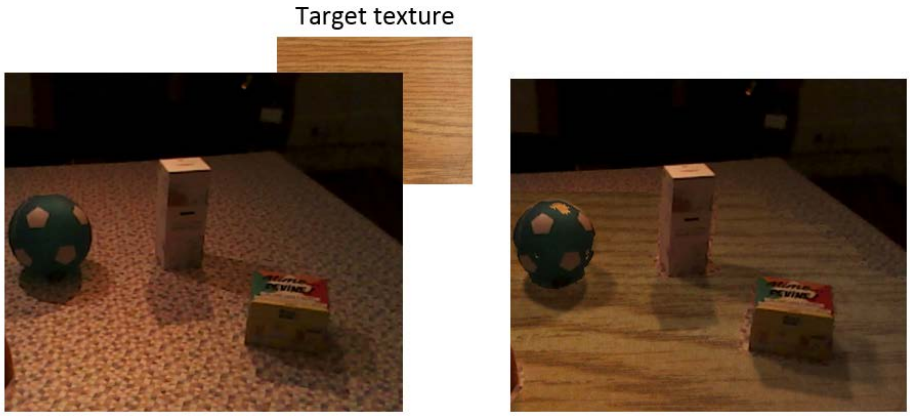


Figure 6.24 – (a) Input color image of the scene with the target texture **T** (top right). (b) Retextured main planar surface using the *illumination* map δ and texture **T**.

6.4 Conclusions and Future Research Directions

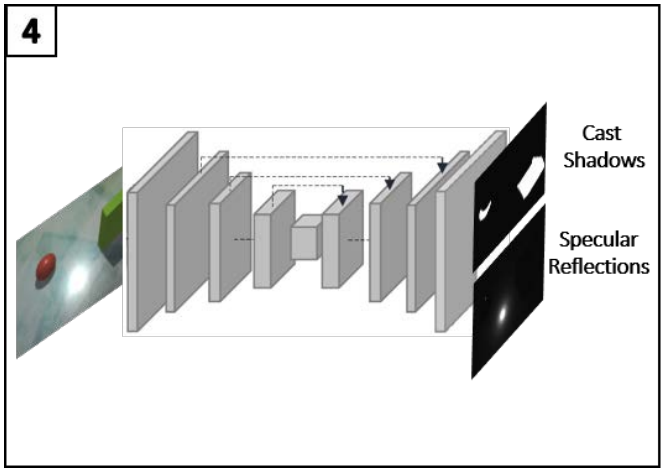
We presented a photometric registration approach which jointly incorporates the information brought by specular reflections and cast shadows to recover reflectance and illumination in the scene. Specifically, our method estimates both 3D position and color of dynamic light sources as well as the specular reflectance of scene surfaces. Our experimental results show satisfactory results on both synthetic and real data where challenging textures are correctly handled and the presence of specular effects is efficiently used within the framework instead of being discarded.

Although the assumption of having a main planar surface is not very constraining for MR scenarios (e.g., table, desk, playground, floor, etc.), one may encounter configurations where this assumption does not hold. In fact, the main reason behind this assumption is related to the coarse geometry provided by common RGB-D sensors. We are therefore interested in handling more generic 3D models.

Finally, the proposed approach requires to capture the scene under an indirect lighting in order to accurately separate texture from illumination within color images. An interesting and challenging research direction corresponds to achieving this task using only the color image of the scene and its 3D model.

7

Specularity and Cast Shadow Detection using a Deep Learning Approach



Contents

In previous chapters, we presented three photometric registration approaches which exploit specular reflections and cast shadows to estimate reflectance and illumination of real scenes. An accurate detection of these cues is at the heart of our work. For instance, in the previous chapter 6, we presented an approach based on specularіy and cast shadow analysis with the aim of estimating the scene's specular reflectance and illumination characteristics. The approach requires three inputs: a 3D model of the scene, a color image captured from a static camera and a *reference* image which corresponds to a capture of the scene from the same viewpoint under a pseudo-ambient lighting. Acquiring this *reference* image is key to robustly separating texture/color from illumination variations. Although this image can be easily produced by an end-user, we are interested in relaxing this constraint.

In the last three years, data driven approaches, especially deep Convolutional Neural Networks (CNN) based methods have outperformed the state of the art in many visual recognition tasks [Girshick et al., 2014][Krizhevsky et al., 2012]. Specifically, several works considered the problem of detecting cast shadows from a single image [Hu et al., 2017][Hosseinzadeh et al., 2017][Shen et al., 2015][Khan et al., 2014] and the results showed significant advances with regard to previous related works [Vicente et al., 2013][Guo et al., 2013]. In this chapter, we address the following question: can deep learning based approaches robustly detect, within our indoor scenes, both specularities and cast shadows from a single image ?

7.1 Problem Overview

Various approaches for shadow detection have been proposed in recent years. Guo et al. [Guo et al., 2013] proposed to model interaction between pairs of regions of the same material, with two types of pairwise classifiers: same illumination condition and different illumination condition. These pairwise classifiers and a shadow region classifier were combined within a Conditional Random Field (CRF) in order to label shadow and non-shadow regions. Similarly, Vicente et al. [Vicente et al., 2013] proposed a Markov Random Field (MRF) that combines a unary region classifier with pairwise and shadow boundary classifiers. These approaches achieved good shadow detection results, but required expensive ground-truth annotation.

Throughout the last four years, convolutional neural networks (CNN) proved to be a very powerful tool to learn pertinent features for detecting shadows, with results clearly outperforming the previous approaches. Khan et al. [Khan et al., 2014] were the first to use deep learning for shadow detection. They combined a CNN for shadow patches and a CNN for shadow boundaries with a Conditional Random Field (CRF), achieving state-of-the-art results at the time. Vicente et al. [Vicente et al., 2017] optimized a multi-kernel model for shadow detection, obtaining even better shadow predictions than [Khan et al., 2014]. Hosseinzadeh et al. [Hosseinzadeh et al., 2017] detected shadows using a patch-level CNN and a shadow prior map generated from handcrafted features. Nguyen et al. [Nguyen et al., 2017] developed scGAN, a novel extension of conditional Generative Adversarial Networks (GAN) tailored for shadow

detection in images. More recently, Hu etl al. [Hu et al., 2017] proposed a novel network for single-image shadow detection and removal by harvesting direction-aware spatial context. The core idea consists in analyzing multi-level spatial context within a spatial Recurrent Neural Network (RNN).

We have tested the proposed shadow detection approaches of [Guo et al., 2013] and [Hu et al., 2017] within our captured images of indoor scenes (Figure 7.1). Although the accuracy keeps improving on the benchmarks [Zhu et al., 2010][Vicente et al., 2016], existing methods still misrecognize dark regions as shadows and poorly handle the critical scenario of textured surfaces (red boxes). False shadow detection includes specular reflections as well (blue boxes).

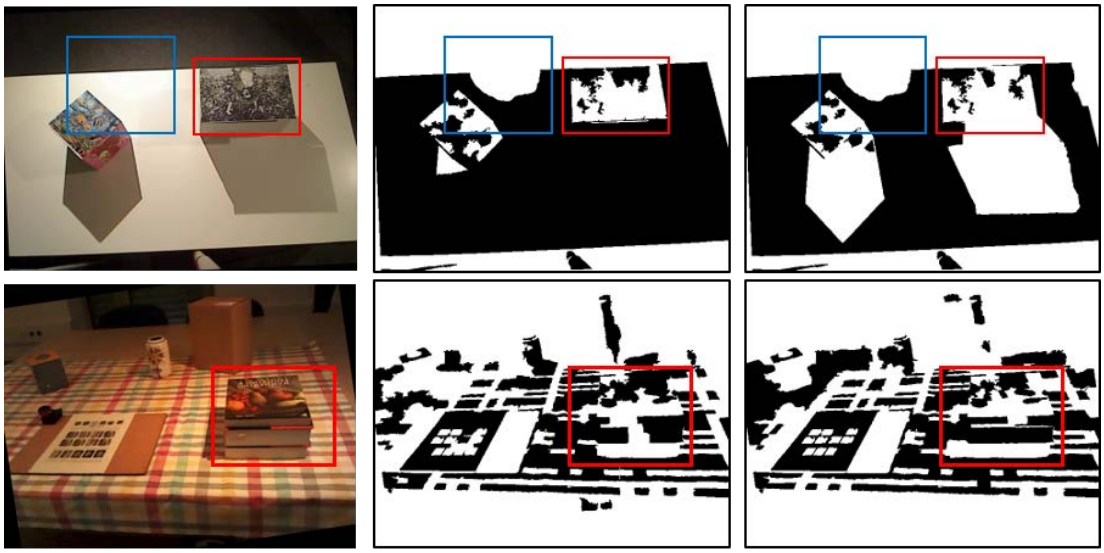


Figure 7.1 – Critical scenarios within the shadow detection task. (a) Color image of the scene. (b) Results using [Guo et al., 2013]: white and black pixels correspond respectively to shadow and non-shadow labels. (c) Results using [Hu et al., 2017]. Red and blue boxes correspond respectively to false detected shadows (dark regions and specularities).

One of the reasons behind such poor results is related to the used dataset during the training phase. For instance, [Hu et al., 2017] use the ImageNet dataset [Deng et al., 2009] which corresponds to a large database with over 14 million of images grouped in 20 thousand categories. Typical groups correspond to semantic objects (e.g., cat, dog, car, etc.) which are very different, in terms of content, from the considered scenes in our case. With regard to specularity detection, to the best of our knowledge, there are no related works which address this task using deep learning approaches.

In this chapter, our goal is to address these challenges. Specifically, we build a large dataset for the task of specularity and shadow detection where scene surfaces retain various, simple and challenging, textures. Also, we jointly detect both cues within captured images using CNN based networks. Moreover, since depth sensing is nowadays

available within consumer phones and tablets, we further incorporate the 3D model of the scene to achieve robust detection. To summarize, the main contributions of this chapter are:

- A large dataset comprising synthetic and real scenes for specularity and shadow detection tasks.
- Joint detection of specular reflections and cast shadows using a deep learning framework.
- Incorporation of the 3D model of the scene to improve our classifier.

The remainder of this chapter is organized as follows: we first present our built dataset. Then, we describe the considered networks architecture to jointly detect specular reflections and cast shadows and demonstrate the effectiveness of incorporating the 3D model. Finally, experimental results are presented and discussed within both our proposed dataset and available benchmarks.

7.2 Our Proposed Method

Our goal is to detect both specular reflections and cast shadows within real scenes. This task is achieved within a classification procedure which takes as input a color image of the scene and results in a map with three labels: cast shadows, specularities and a third class which represents points that are neither cast shadows nor specularities (e.g., self-shadows, background, etc.). In the following, we present our built dataset and demonstrate the effectiveness of incorporating the 3D model within the classification procedure.

7.2.1 Built Dataset

The success of deep convolutional neural networks is dependent on the availability of annotated large-scale datasets [Russakovsky et al., 2014][Lin et al., 2014]. Collecting and annotating large-scale datasets of real scenes takes considerable time and effort for most of the deep learning related classification tasks. An alternative consists in the use of synthetic data which proved to produce competitive performance [Mayer et al., 2015]. Nonetheless, to achieve better generalization, real data must be considered as well.

With regard to the shadow detection task, available datasets [Zhu et al., 2010][Vicente et al., 2016] mainly contain images of outdoor scenes (Figure 7.2-(a,b)). Consequently, since the sun is the primary light source in these scenarios, one often encounters a single shadowed region within the images of these datasets. Furthermore, considered surfaces on which shadows are cast often retain fairly simple textures. However, in indoor real scenes composed of one or more objects, several shadows cast on arbitrary textures can be present (Figure 7.2-c). With regard to the specularity detection task, to our knowledge, there is not any dataset available. Consequently, in order to address the task of jointly detecting specular and shadow cues, we must built an adequate dataset.

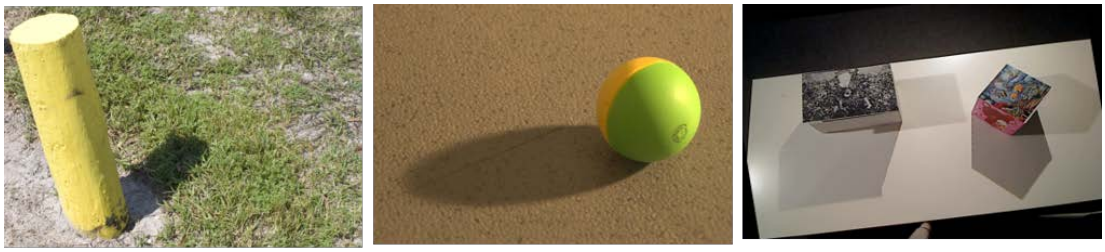


Figure 7.2 – Examples of images from [Zhu et al., 2010] (a), [Vicente et al., 2016] (b) and an image of considered indoor scenes in our work (c).

To create a comprehensive dataset of specularity and shadow cues, we consider both synthetic and real scenes where various geometries, textures and illumination conditions are present. In the following, we present our data crafting procedure along with the produced ground-truth images.

Synthetic Scenes

Our synthetic dataset contains 11956 images of virtual scenes. The production of these images is achieved using Unity [Unity, 2018] where 80 virtual scenes with various shaped-objects are located on a main planar surface (Figure 7.3).

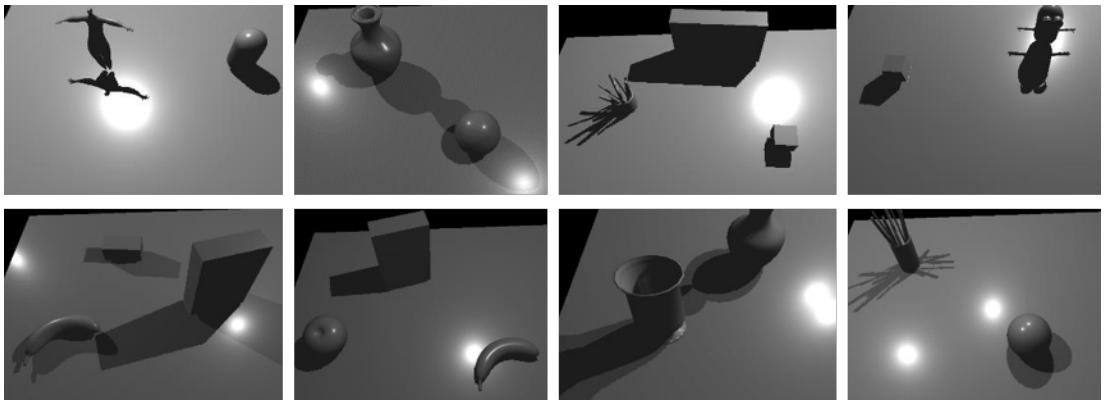


Figure 7.3 – Examples of virtual scenes from our dataset where differently shaped objects are located on a main planar surface. The images correspond to *illumination* maps which contain shading, shadowing and specular effects.

Furthermore, we consider 80 texture maps (Figure 7.4, Appendix B.1) used along with different specular reflectance properties to render the virtual scenes. This large collection of textures results in various challenging cast shadow scenarios. Equally, specular reflections are rendered in order to address indoor scenes where both cues can be present.

For each virtual scene, we randomly vary the viewpoint and freely move 3 point lights with different intensities in the scene. Consequently, for each color image, we derive the

Figure 7.4 – Examples of collected texture maps which are used to create our synthetic dataset. Further examples are shown in Appendix B.1.

following ground-truth data (Figure 7.5): (1) Depth map of the scene rendered from a given viewpoint. (2) 3D segmentation of the scene retrieved in a 2D map with three labels: main planar surface, 3D objects located on the plane and background. (3) Diffuse reflectance map which contains only the intrinsic color/texture of scene surfaces. (4) Illumination map which contains illumination-dependent effects (shading, shadowing and specular reflections). (5) Mask of cast shadows represented by white pixels. (6) Mask of specular reflections represented by white pixels. (7) Light sources positions. (8) Camera parameters (intrinsic and extrinsic).

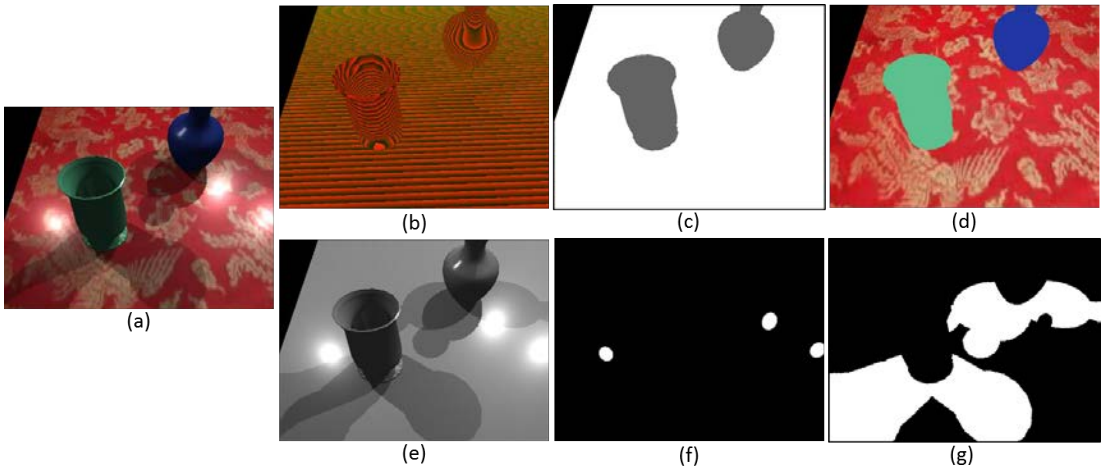


Figure 7.5 – Example of produced ground-truth data. (a) color image of the scene. (b) 16-bits depth map. (c) Segmented scene map where white pixels correspond to the main planar surface, grayscale pixels are objects located on the plane and black pixels correspond to background. (d) Diffuse reflectance map which retains the intrinsic color/texture of the scene. (e) Illumination map which retains shading, shadowing and specular effects with regard to the color image (a). Specularities and cast shadows are respectively retrieved in the binary maps (f) and (g) where they are represented by white pixels.

Real Scenes

Our real dataset contains 3052 images of real indoor scenes (Figure 7.6). The production of ground-truth images is achieved using the method described in chapter 6 where color images and 3D model of the scene are acquired using an RGB-D sensor (R200).

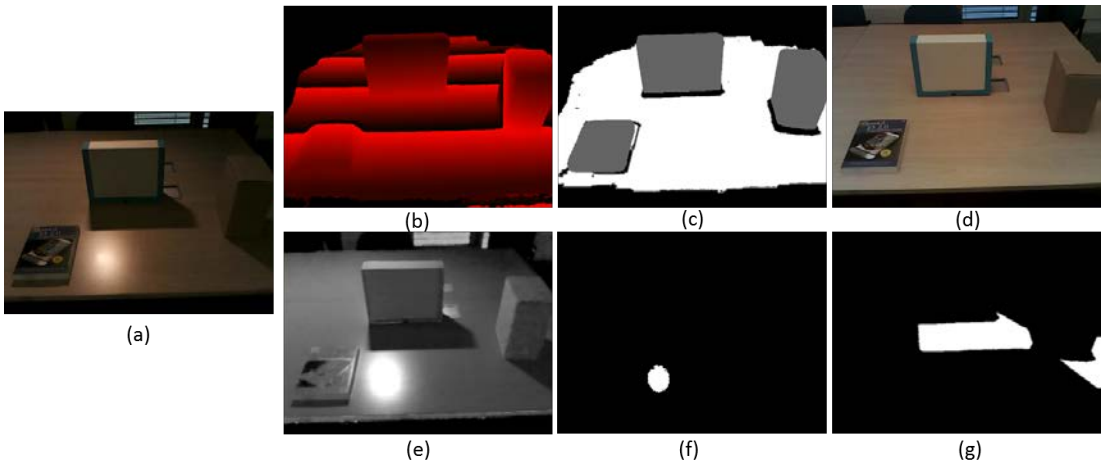


Figure 7.6 – Example of acquired and estimated ground-truth data using an RGB-D sensor (R200). (a) color image of the scene. (b) 16-bits depth map. (c) Segmented scene map where white pixels correspond to the main planar surface, grayscale pixels are objects located on the plane and black pixels correspond to the background/noise. (d) Diffuse reflectance map which mainly retains the intrinsic color/texture of the scene. (e) Illumination map which mainly retains shading, shadowing and specular effects with regard to the color image (a). Specularities and cast shadows are respectively retrieved in the binary maps (f) and (g) where they are represented by white pixels.

Some inaccuracies can be noticed within our real dataset when compared to the synthetic one. For instance, the diffuse reflectance map corresponds to a captured image of the scene under a *pseudo-ambient* lighting (Figure 7.6-d). Also, the coarse geometry provided by the R200 introduces inaccuracies with regard to the segmented scene map (Figure 7.6-c). Nonetheless, this data brings valuable information to the task of specularity and shadow detection for indoor real scenes and its effectiveness is demonstrated later on in this chapter.

7.2.2 Network Architectures

The typical use of convolutional neural networks is within classification tasks, where the output to a 2D image is a single class label (e.g., an image of a car would have the label 'car'). However, since we are interested in a per-pixel labeling, we consider two recently proposed network architectures which are adequate for this task: U-Net [Ronneberger et al., 2015] and Ternaus-Net [Iglovikov and Shvets, 2018]. These architectures work with fewer training images and yield more precise classification results in comparison with previous proposed networks [Long et al., 2014].

U-Net Architecture

The U-Net network architecture proposed by [Ronneberger et al., 2015] is illustrated in figure 7.7. It consists of a contracting path (left side) and an expansive path (right side). The contracting path (encoder) follows the typical architecture of a convolutional network. It consists of the repeated application of two 3×3 convolutions, each followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step, the number of feature channels is doubled. Every step in the expansive path (decoder) consists of an upsampling of the feature map followed by a 2×2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. At the final layer a 1×1 convolution is used to map each feature vector to the desired number of classes. In total the network has 23 convolutional layers.

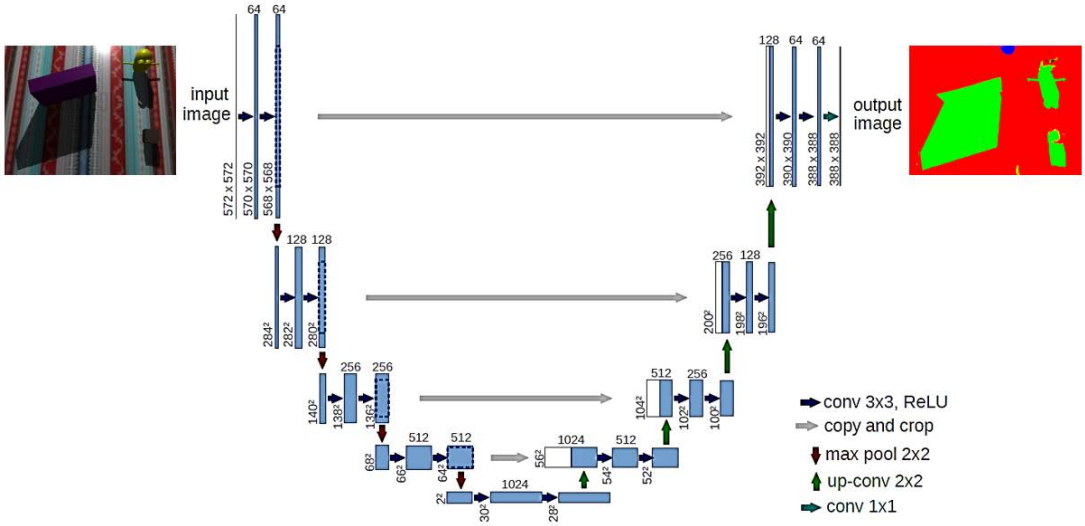


Figure 7.7 – U-net architecture: blue boxes corresponds to multichannel feature maps. White boxes represent copied feature maps. Figure from [Ronneberger et al., 2015].

As illustrated in figure 7.7, the output to a 2D image from our dataset is a 2D map, with the same resolution of the input image, where the color corresponds to the probability of belonging to three classes: green and blue channels represent respectively shadow and specularity classes. Red channel corresponds to the probability to belong to neither shadows nor specularity.

Ternaus-Net Architecture

Recently, an improvement of the U-Net architecture was proposed in [Iglovikov and Shvets, 2018]. Typically, neural networks initialized with weights from a network pre-trained on a large dataset like ImageNet [Deng et al., 2009] show better performance than those trained from scratch on a small dataset. Hence, Iglovikov et al. [Iglovikov and Shvets, 2018] proposed Ternaus-Net, a U-Net architecture where the encoder is pre-trained using ImageNet. Specifically, the pretrained encoder corresponds to the

convolutional network VGG-16 [Simonyan and Zisserman, 2014] consisting of 16 convolutional layers which proved to well perform for the extraction of image features.

Training

The input images and their corresponding ground-truth classification maps are used to train the networks. The loss function corresponds to the Cross Entropy between input images and ground-truth labels. The obtained output is an image where each pixel color channel corresponds to the probability to belong to a class.

In our work, the classification output is of interest because an accurate detection of specularities and cast shadows can be incorporated in the photometric registration method described in chapter 6. Consequently, since the 3D model of the scene is available, we further consider incorporating 3D information within our training. This is achieved by considering a 4^{th} channel within the input RGB image which corresponds to the map of the segmented scene (Figures 7.5-c and 7.6-c). In this case, only the pixels belonging to the plane within which we aim at detecting specular and shadow cues are considered within the loss function.

7.3 Experimental Results

In this section, we present experiments to evaluate our classification results. The built dataset is split into training, validation and test sets with the respective following proportions: 80%, 10% and 10%. For each evaluation, we perform 40 epochs which lasts approximately 10 hours on an NVIDIA GTX 1080. Within each epoch, we apply scaling, rotation and mirroring operations to augment our data variety. Saved models for considered networks correspond to the minimum validation loss. The framework used in this work is PyTorch [PyTorch, 2018].

With and without the segmentation map

Since the 3D model of both synthetic and real scenes is available, we dispose of a map with three labels representing a clustering of the scene into the main planar surface (white pixels), 3D objects located on the plane (grayscale pixels) and background points represented by black pixels (Section 7.2.1). Considered networks are trained using first only the color image as input an then using both the color image and the corresponding segmentation map. The classification results within the U-Net architecture are shown in figure 7.8. The output is a 2D color map where the pixel's color channels correspond to the probability to belong respectively to cast shadows (green), specularities (blue) and the rest of the scene (red). Thus, a pixel's dominant color channel corresponds to the most likely class it belongs to.

In figure 7.8, one can notice the improvement of the classification when using the segmentation map. In fact, cast shadows as well as specular effects are better detected within the principal planar surface (first row). Furthermore, the use of the segmentation shows its effectiveness with regard to discarding black regions as shadows such as

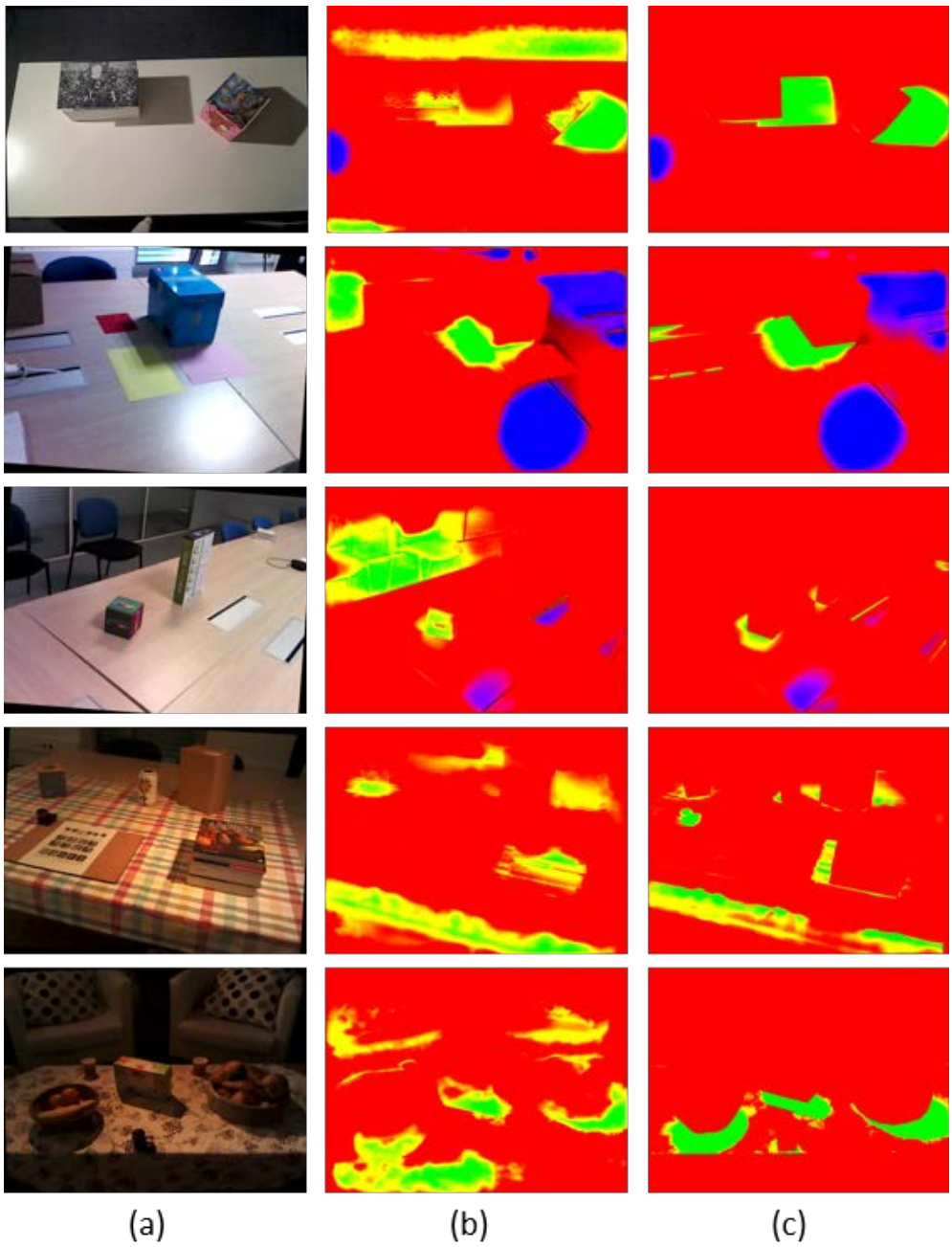


Figure 7.8 – Comparison of classification results with regard to the use of the scene's segmentation map. (a) Input color images from our test set. (b) Classification results without the segmentation map: pixel color corresponds to the probability to belong to specularity (blue), cast shadows (green) or the rest of the scene (e.g., background, objects). (c) Classification results using the segmentation map.

the brown box (second row) and background chairs (third row). The results in figure 7.8 are obtained by training the networks on synthetic data only. Although it performs well even in presence of challenging textures (fourth row), the classifier does not handle weak and soft shadows (third row) and specular reflections can be erroneously detected

(second row).

With and without real data

In the following, we demonstrate the effectiveness of our real dataset with regard to improving the classification results. The U-Net network is trained, using color images and segmentation maps as input, by considering first only synthetic data and then both synthetic and real data. The results are shown in figure 7.9 within a selection of test images from our built database.

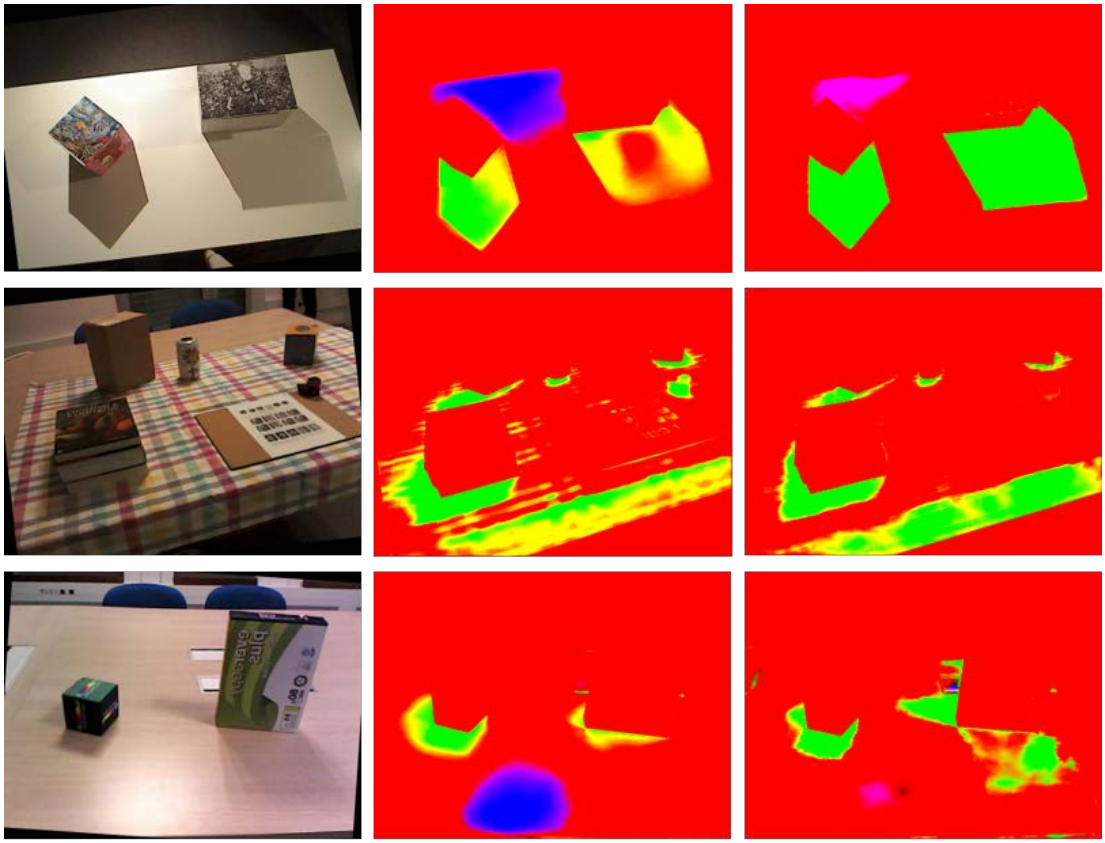


Figure 7.9 – Comparison of classification results with regard to the use of our real dataset (the segmentation map is used as well for (b) and (c)). (a) Input color images from the test set. (b) Classification results using synthetic data for the train set: pixel color corresponds to the probability to belong to specularity (blue channel), cast shadows (green channel) or the rest of the scene (e.g., background, objects) (red channel). (c) Classification results using synthetic and real data for the train set.

The classifier achieves better results when trained on both synthetic and real data. In fact, cast shadows are better detected on both single-color (first row) and textured surfaces (second row). Although the classifier does not perform as good with regard to *very* weak shadows (first row), the use of real data shows improvement in detecting

soft shadows (third row). With regard to specularity detection, the pixels classified within this class have a lower probability when using real data. Nonetheless, since the output maps are usually converted into binary maps using a max operator within the three labels, these pixels are eventually detected as specular effects. Further results are shown in Appendix B.2.

U-Net and Ternaus-Net performances

We evaluated the performance of a lighter trained-from-scratch architecture (U-Net) and a pre-trained architecture (Ternaus-Net) where the initial weights within the encoder correspond to VGG-16 and only the decoder is trained with our dataset. Results of the classification are shown in figure 7.10.

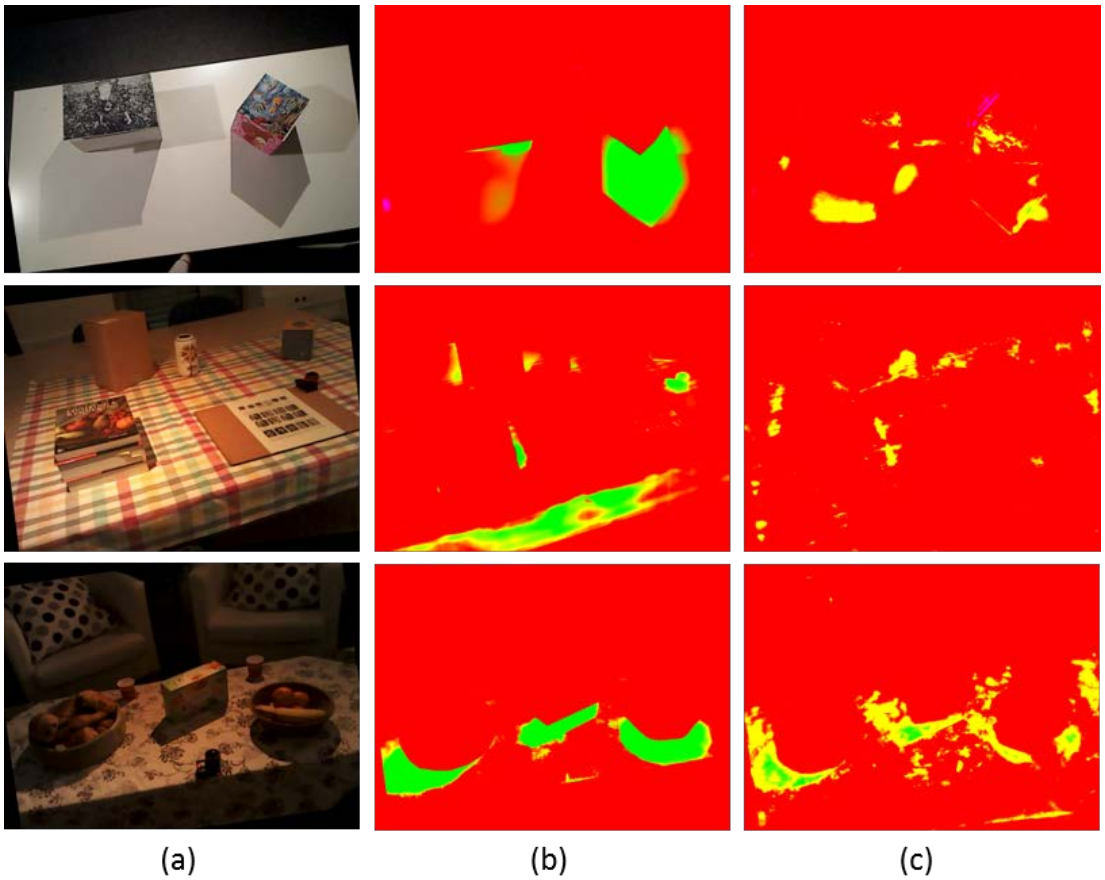


Figure 7.10 – Classification results for images in (a) using U-Net which is trained from scratch with our dataset and (b) Ternaus-Net where the encoder is pre-trained on ImageNet [Deng et al., 2009] and only the decoder is trained using our dataset.

The U-Net architecture outperforms the pre-trained Ternaus-Net. In fact, one can notice in the first and second rows that the cast shadows are better detected using U-Net (Figure 7.10-b) than Ternaus-Net (7.10-c). Although, in the last row, Ternaus-Net delivers good results, the achieved classification is not as good as using U-Net. The

reason behind such results is explained by the fact that the U-Net counts 3.5 million model parameters, all trained using our dataset. On the other hand, the Ternaus-Net counts 18 million parameters where only 3.4 million (decoder) are trained using our dataset. The rest of the parameters are retrieved from a pre-training on the ImageNet dataset containing significantly different images than ours.

Evaluation within Available Benchmarks

We evaluate our classifier within two available shadow detection benchmarks. The first one is the SBU Shadow Dataset [Zhu et al., 2010], which is the largest publicly available annotated shadow dataset with 4089 training images and 638 testing images. The second dataset is the UCF [Vicente et al., 2016]. It includes 145 training images and 76 testing images, and covers outdoor scenes with various backgrounds. The only available ground-truth data within these benchmarks corresponds to shadows. Hence, in order to test our trained U-Net using the segmentation map, we manually created such maps for a selection of images (Figure 7.11).

Figure 7.11 – (a) Images from available benchmarks [Zhu et al., 2010][Vicente et al., 2016]. (b) Manually crafted segmentation map for images in (a).

Although these images are different from our indoor scenes, the classifier achieves good results with regard to detecting cast shadows. The effectiveness of the segmentation map is stressed again within the fifth row where the results are not as good as the other shown images. In fact, since the occluding object is not captured within the image, the segmentation map only contains white pixels representing the plane (ground floor). Further classification results are shown in Appendix B.3 .

A quantitative comparison with regard to state-of-the-art methods on the available benchmarks requires crafting the segmentation map for several images which is very laborious. Another way of comparison consists in changing the considered inputs within related work architectures [Hu et al., 2017][Nguyen et al., 2017] from only the color image to both the color image and segmentation map, and training them using our

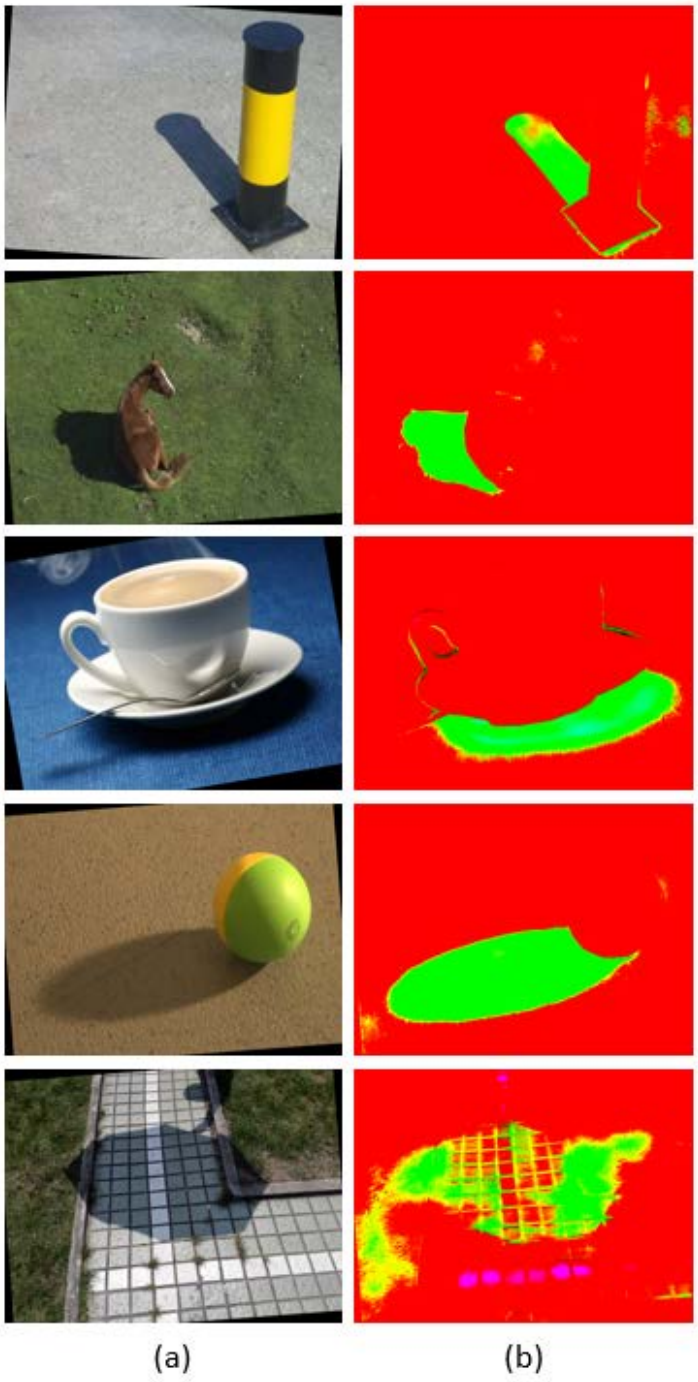


Figure 7.12 – Results of our classifer (U-Net with color image and segmentation map as inputs) within a selection of images from the (SBU) [Zhu et al., 2010] and (UCF)[Vicente et al., 2016] datasets.

dataset. This evaluation represents a good measure of the effectiveness of our dataset as well as incorporating 3D information and should be considered within future work.

7.4 Conclusions

In this chapter, we presented a deep learning approach to detect both specularities and cast shadows within indoor real scenes. Since available datasets for this task mainly contain outdoor scenes where the sun is the primary light source, we built a new dataset comprising synthetic and real scenes where challenging textures and lighting conditions can be encountered. Furthermore, we demonstrated the effectiveness of using 3D information about the scene to improve the classification results. In fact, by constraining the loss function within pixels belonging to the projection of the main planar surface, specularities and cast shadows are robustly detected.

Conclusion and Perspectives

8

In this thesis, entitled "**Photometric Registration of Indoor Real Scenes using an RGB-D Camera with Application to Mixed Reality**", we focused on developing, using a single RGB-D camera, novel photometric registration approaches for indoor real scenes. The goal of such algorithms is to estimate the reflectance and illumination of the scene. These estimates are key to achieving realistic mixed reality scenarios where virtual shadows are visually coherent with shadows cast by real objects and, real specularities occluded by virtual objects are correctly rendered. Existing approaches either introduce additional devices (e.g., chrome sphere, camera equipped with fish-eye lenses) which can be cumbersome within the MR experience or constrain the scene's content (e.g., scene reduced to a single object, illumination represented by a single light source, Lambertian surfaces). Furthermore, scene illumination is often assumed to be static. Nonetheless, within MR scenarios, an end-user might necessitate to change the lighting indoors. In this thesis, we addressed these limitations by taking advantage of acquired RGB-D information.

In chapter 4, we presented a probeless photometric registration method which recovers reflectance and illumination by analyzing observed specular reflections throughout an RGB-D sequence. In contrast to methods that recover specular reflections as saturated regions in input images, we robustly handle specularities by retrieving a luminance profile which retains the evolution of the pixel's luminance when the camera browses the scene. The proposed approach handles real scenes where the texture spatially varies and several objects with different shapes can be encountered. Furthermore, it recovers the position of multiple light sources without any user intervention. Our estimates were integrated within a rendering pipeline to demonstrate realistic MR scenarios such as virtual objects correctly occluding real specularities (accurate reconstruction of the specularity region using estimated diffuse reflectance) as well as realistic virtual shadows in terms of shape and intensity.

In chapter 5, we addressed the problem of deriving illumination characteristics, namely the 3D position and intensity, from cast shadows. Most importantly, we tackled the critical scenario of textured surfaces. The proposed approach handles multiple light sources which can be static and/or dynamic and runs at an interactive frame rate (4 fps). Consequently, the MR user can freely move or turn on/off the lights in the scene and notice the online changes within synthetic objects as well. Finally, the proposed approach was integrated within an industrial project at Technicolor aiming at demonstrating the effectiveness of realistically rendering virtual objects in indoor real scenes.

In chapter 6, we proposed a method which efficiently incorporates the information

brought by both specular reflections and cast shadows. This approach covers a large panel of indoor scenes since it handles both Lambertian and glossy surfaces. Moreover, it considers scenes which can be composed of one or more objects with arbitrary shapes and textures. The proposed approach recovers multiple light sources characteristics (3D position and color) and estimates specular reflectance of scene surfaces. The framework runs at an interactive frame rate (4fps) and tackles the dynamic aspect of lighting which is of interest in MR scenarios. We demonstrated the effectiveness of our method through a range of applications including real-time mixed reality scenarios where the rendering of synthetic objects is consistent with the real environment (e.g., correct specularity removal, visually coherent shadows) and retexturing where the texture of the scene is altered whereas the incident lighting is preserved.

Finally, since the detection of specular reflections and cast shadows is at the heart of our work, we proposed in chapter 7 a deep-learning framework to jointly detect both cues. To achieve this, we built a large and comprehensive dataset which comprises synthetic and real scenes with various textures, specular reflectance properties and lighting conditions. An additional advantage of this dataset consists in containing not only specularity and cast shadow ground-truth images but further useful information about the scene (e.g., depth map, diffuse reflectance, segmented scene). Moreover, we demonstrated the effectiveness of incorporating the 3D model of the scene in our classifier to robustly detect specular reflections and cast shadows. The proposed approach was tested within a variety of synthetic scenes and challenging indoor real scenes. Furthermore, it achieved good results within available benchmarks despite their content difference with regard to our dataset (e.g., outdoor scenes).

Short-term perspectives

Several improvements can be considered in the short-term with regard to the proposed photometric registration approaches:

Main planar surface assumption (Chapters 5 and 6): in most of the presented work, we considered the scenario of an indoor real scene with a main planar surface on which are located objects with arbitrary shapes. Although this assumption still allows us to cover a large panel of real scenes (e.g., table, desk, floor), we are interested in more generic photometric registration solutions. The main reason behind this assumption consists in taking advantage of the surface-smoothing feature with regard to planar surfaces that numerous RGB-D sensors provide. An alternative consists in refining the coarse geometry of *any* scene acquired using an RGB-D camera. For instance, algorithms such as [Choe et al., 2016][OrEl et al., 2015] can be considered for this task.

Retexturing application (Chapter 6): we demonstrated the utility and convenience of recovering the *illumination* maps within a retexturing application. In fact, since the estimation of these maps is achieved in real-time, the retexturing application runs in real-time as well. The main inaccuracies within the shown scenarios (Figure 6.24) are due to coarse geometry. Similarily to the previously discussed short-term perspective,

this application can benefit from a refined 3D model of the scene. Furthermore, an interesting aspect to address within this scenario is the interaction of the user with the scene. For instance, using texture-mapping techniques, we can establish an atlas of the scene (Figure 8.1) where the user can freely choose which part of the real scene he/she would like to alter.

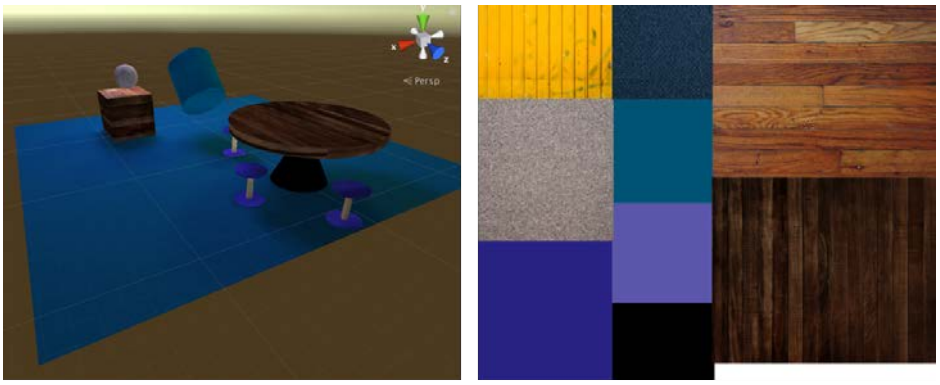


Figure 8.1 – (a) 3D view of a virtual scene. (b) Atlas of the scene in (a) containing the texture of each object separately.

Illumination estimation using deep learning approaches: in chapter 7, we presented a deep learning framework to jointly detect specular reflections and cast shadows. In fact, this framework was considered in order to relax the constraint made within chapters 5 and 6 with regard to acquiring a capture of the scene under a pseudo-ambient lighting. Although capturing this image is only getting easier with the increasing camera capabilities integrated in common end-user phones (Figure 8.2), we are interested in deriving illumination properties using the classification results.

Figure 8.2 – (a) Capture of the scene using a pseudo-ambient lighting (top): the lighting corresponds to a soft distant window lighting. (b) Capture of the scene with the lights turned off (top) using a Samsung S8.

Joint Specularity and Shadow Detection (JSSD) dataset: in chapter 7, we

presented a large dataset adequate for the task of specular reflections and cast shadows detection. The latter mainly comprises scenarios of indoor real scenes. Within our experimental results, we further evaluated our framework on two available datasets. However, since the inputs are not similar, it was laborious to manually craft the segmented scene map. A next step consists in integrating outdoor real scenes in our dataset. Also, we are interested in modifying the inputs of already existing shadow detection networks to take into account both the color image and segmentation map in order to evaluate the accuracy of the classifcation. Finally, the built dataset should be prepared for free access within the research community.

Long-term perspectives

Realism assessment: in [Jacobs and Loscos, 2006], several quality criteria have been proposed to assess existing photometric registration approaches: input requirements, processing time, level of automation, level of interaction and amount of realism. Most of these criteria can be measured or precisely listed except for the amount of realism. In fact, the concept of *realism* can be subjective and differs within individuals. In order to better assess it, several perceptual metrics within an available and exhaustive benchmark should be carried out using statistical measures.

Advanced reflection models: in this thesis, we mainly considered Phong reflection model [Phong, 1975] to describe the way a point in the scene reflects incident light. This model owes its popularity to its simplicity and convenience with regard to scene analysis and real-time rendering of MR scenarios. Nonetheless, this reflection model uses point light representation for real light sources. Such representation produces only hard shadows while soft shadows cast by real objects can be encountered. An interesting improvement consists in considering more sophisticated models such as [Cook and Torrance, 1981] and addressing accurate soft shadows modeling and detection. Furthermore, since several recently proposed global illumination techniques run in real-time on common end-user devices [Lecocq et al., 2016], a next step consists in considering such approaches for more realistic effects (e.g., inter-reflections between real and virtual objects, soft shadows).

Dynamic scene geometry: in this work, we tackled the dynamic aspect of light sources in the scene which allows the end-user to experience MR in less constraining environments. Equally, addressing the problem of dynamic geometry is an important improvement with regard to MR requirement, especially for scenarios where the scene is often subject to changes (e.g., gaming). Consequently, proposed approaches within this task such as [Newcombe et al., 2015] could be considered.

Appendix: Realistic Mixed Reality Demonstrator

A

Within an industrial project at Technicolor aiming at delivering realistic Mixed Reality (MR) scenarios in the context of dynamic lighting, we have integrated, for this purpose, the framework described in chapter 5. Specifically, we have designed and implemented an interactive demonstrator that shows a realistic MR application without using any light probe. The proposed demonstrator was presented at both ISMAR and APMAR conferences in 2018. The proposed system takes as input the RGB stream of the real scene, and uses these data to recover both the position and intensity of light sources. The lighting can be static and/or dynamic and the geometry of the scene can be partially altered. Our system is robust in presence of specular effects and handles both uniform and/or textured surfaces.

Figure A.1 – Captures of our Mixed Reality (MR) application for real scenes with a uniform and Lambertian surface (left), a uniform and specular surface (center) and a challenging textured surface (right). The lighting condition can be changed within the MR experience.

A.1 Introduction

Although MR technology is already present in the consumer-product market for a number of industrial applications such as training and entertainment, it still lacks *realism* [Jacobs and Loscos, 2006]. For instance, in most cases, virtual objects are rendered using an arbitrary and static lighting that does not correspond to the real-world lighting condition. In order to address this problem, proposed algorithms and systems must recover real-world lighting and surface reflectance properties in order to achieve realistic computer-graphics renderings (e.g., realistic shadows in terms of orientation and intensity, specular reflections, etc.). Furthermore, if the real lighting changes, the system must be able to detect these changes and update, in real-time, the virtual lighting characteristics. In this demonstrator, we have designed and implemented a system which

analyzes the real scene in order to recover its photometric properties. This mainly corresponds to the work presented in chapter 5 where:

- Without using any light probe, we recover both the 3D position and intensity of light sources present in the scene. The lighting can be static and/or dynamic.
- Our system handles both uniform and challenging textured surfaces.
- The MR application runs in real-time on a tablet and delivers interactive and visually coherent augmentations.

The remainder of this chapter is structured as follows: we first specify the main requirements of our demonstrator, then describe the overall architecture of our system and briefly present its components. Finally, we overview the experience which an end-user has as well as the features that our demonstrator offers.

A.2 Demonstrator Description

A.2.1 Requirements

The main requirements of our demonstrator are depicted in figure A.2. The considered scenario is the following:

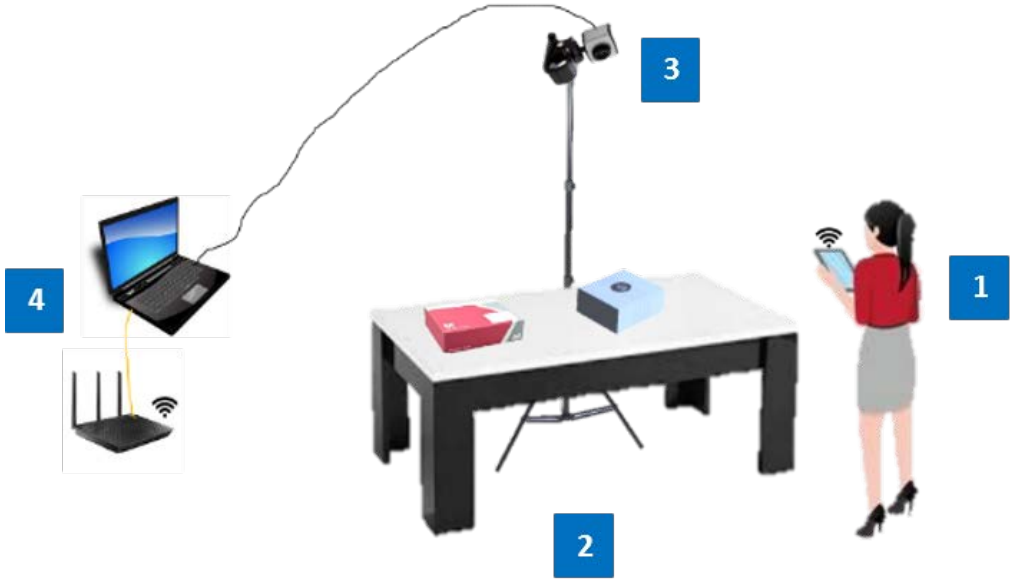


Figure A.2 – The overall setup of our Mixed Reality demonstrator.

- (1) An end-user stands in front of a real scene, holding a tablet through which he/she sees the augmentations.
- (2) The scene is mainly composed of a principal planar surface (table, desk, etc.) and two objects. One of the considered objects can be freely moved within the user's experience whereas the remaining one is fixed.

- (3) In addition to the tablet, a static camera observes the scene from the top. The RGB stream of this camera is used for scene analysis.
- (4) Finally, expensive computations are launched on a laptop that communicates the results using wireless network , in real-time, to the tablet.

A.2.2 Architecture Overview

In our system requirements described in section A.2.1, we mentioned a shared processing between a PC and an Android tablet. The application running on the tablet is developed using Unity [Unity, 2018] whereas the modules running on the laptop are developed in C++. In figure A.3, a diagram depicts the overall architecture of the proposed system:

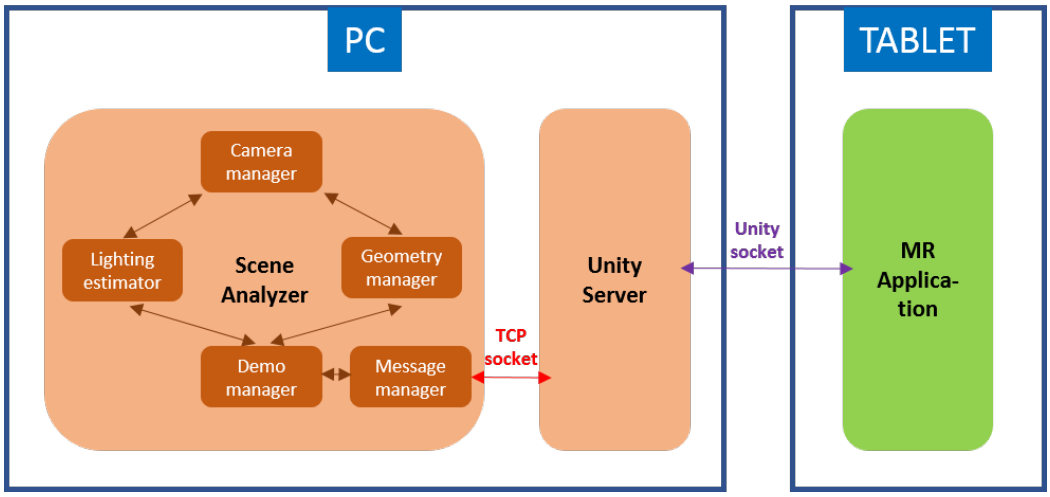


Figure A.3 – Main components of our MR demonstrator's architecture.

The architecture contains three main components:

- **Scene Analyzer:** it mainly handles lighting and geometry changes in the scene.
- **MR Application:** it is the only module running on the tablet and it handles the rendering of the virtual world using updated lighting and geometry characteristics.
- **Unity Server:** it manages the communication between the *Scene Analyzer* and the *MR application*.

The *Scene Analyzer* contains five different modules:

- **Camera manager** controls the stream of the static camera.
- **Demo manager** updates the changes that might occur in the scene's lighting and/or geometry.

- **Message manager** is the interface between the *Scene Analyzer* and the *Unity Server*. The communication between these modules is achieved via messages delivered over a TCP socket.
- **Geometry manager** aims at detecting and tracking the objects on the principal planar surface. This is achieved using the Vuforia SDK [Vuforia, 2018]. Changes are detected with regard to the initial position and orientation of the second object and updates are sent to the *Demo manager* module.
- **Lighting estimator** uses the RGB stream provided by the static camera as well as the 3D model recovered using *Geometry manager* in order to estimate both the 3D position and intensity of light sources. The processing is achieved for each captured image using the approach described in chapter 5.

A.2.3 Experience

An end-user stands in front of a real scene holding a tablet through which the MR scenario can be viewed. Our demonstrator runs in real-time and takes into account changes that can occur in lighting (lights switched on/off or moved) and geometry (moving an object on the planar surface). For instance, using a remote control, the user can switch on/off the lights in the scene and see the changes occur on the virtual objects as well. Furthermore, the user can interact with the virtual object in two different ways: (1) by moving its finger on the tablet screen, he changes the position of the virtual object on the planar surface. (2) several animations are accessible via the UI-toggles located on the left side of the tablet screen. The *Amount of realism* achieved by our demonstrator [Jacobs and Loscos, 2006] is convincing. For instance, the orientation and intensity of virtual shadows are consistent with real cast shadows. Also, real lighting conditions are automatically detected and used to update the virtual lighting model. Our demonstrator is able to detect up to three light sources (Figure A.4) and handles, to some extent, windows lighting by recovering it as a distant point light (Figure A.5). Further results of our demonstrator are shown in this video.

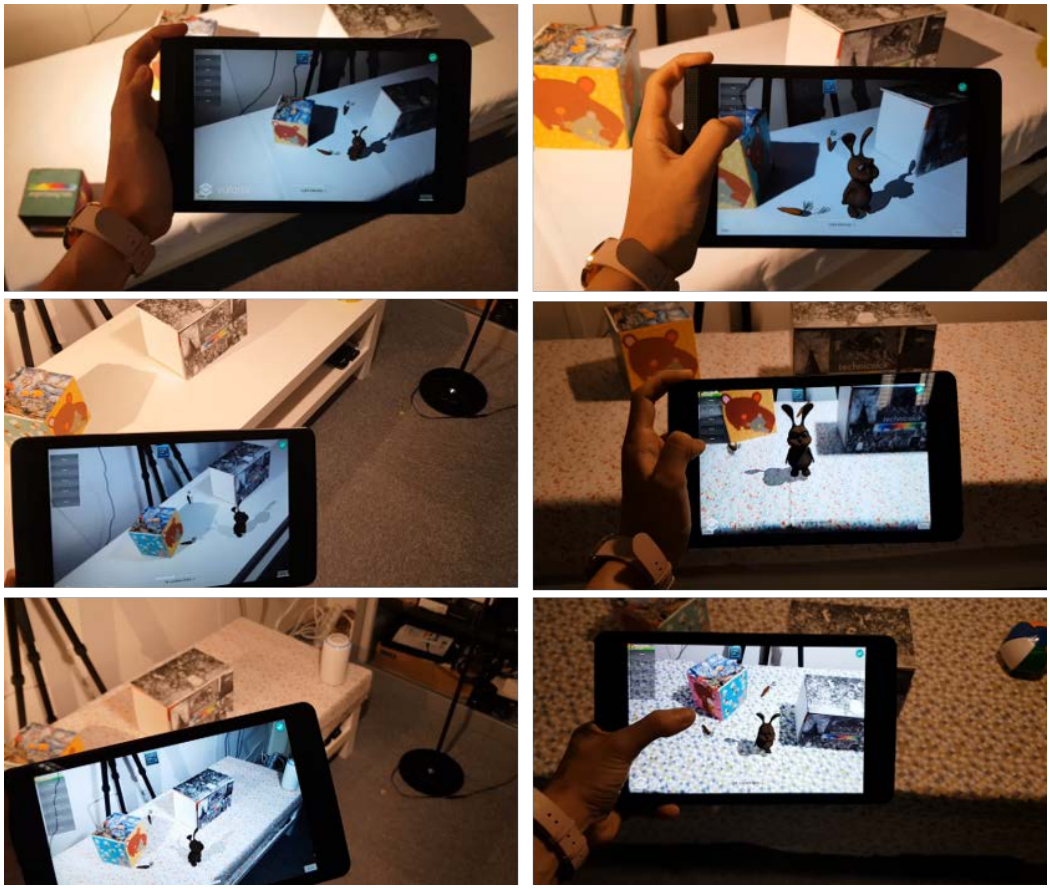


Figure A.4 – Captures from the proposed mixed reality demonstrator: the real scene is illuminated by artificial indoors lighting.

Figure A.5 – Captures from the proposed mixed reality demonstrator: the real scene is illuminated by natural windows lighting.

Appendix: Joint Specularity and Cast Shadow Detection using a Deep Learning Approach

B

Figure B.1 – Examples of collected texture maps which are used to create our synthetic dataset.

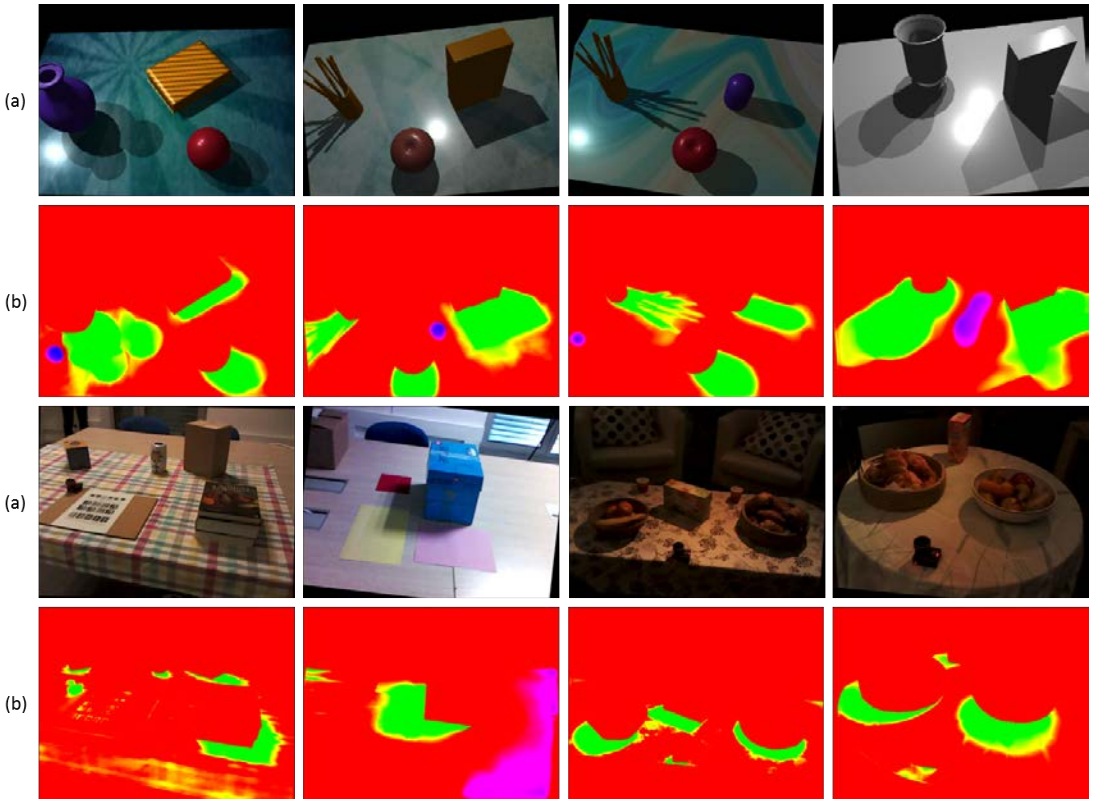


Figure B.2 – (a) Input color images from the test set. (b) Classification results using our dataset: pixel color corresponds to the probability to belong to specularity (blue channel), cast shadows (green channel) or the rest of the scene (e.g., background, objects) (red channel).

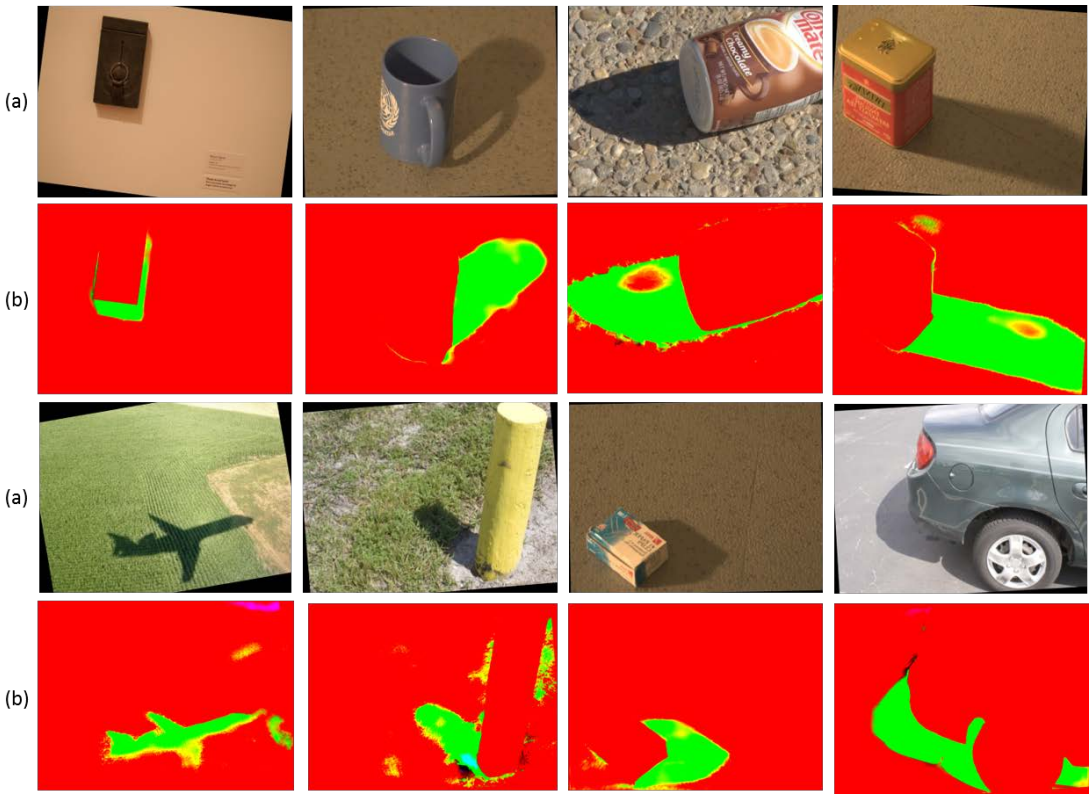


Figure B.3 – Results (b) of our classifer (U-Net with color image and segmentation map as inputs) within a selection of images (a) from the (SBU) [Zhu et al., 2010] and (UCF)[Vicente et al., 2016] datasets.

List of Tables

List of Figures

Bibliography

Aimone, C. and Mann, S. (2007). Camera response function recovery from auto-exposure cameras. In *2007 IEEE International Conference on Image Processing*, volume 4, pages IV – 233–IV – 236. 37

Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 214–227, Berlin, Heidelberg. Springer Berlin Heidelberg. 25

Alsaleh, S. M., Aviles, A. I., Sobrevilla, P., Casals, A., and Hahn, J. K. (2015). Automatic and robust single-camera specular highlight removal in cardiac images. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 675–678. 53

Ambai, M. and Yoshida, Y. (2011). Card: Compact and real-time descriptors. In *2011 International Conference on Computer Vision*, pages 97–104. 25

Anusorn, B. and Nopporn, C. (2016). Light source estimation using feature points from specular highlights and cast shadows. In *International Journal of Physical Sciences*, volume 11, pages 168–177. 99, 108, 109

Arief, I., McCallum, S., and Hardeberg, J. Y. (2012). Realtime estimation of illumination direction for augmented reality on mobile devices. *Color and Imaging Conference*, pages 111–116. 45, 46, 79, 159

Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoper. Virtual Environ.*, 6(4):355–385. 2

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vision*, 56(3):221–255. 57

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg. Springer Berlin Heidelberg. 25

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256. 22

Boom, B., Orts, S., Ning, X., McDonagh, S., Sandilands, P., and Fisher, R. B. (2013). Point light source estimation based on scenes recorded by a rgb-d camera. In *British Machine Vision Conference*. 52

Botella, C., Breton Lopez, J., Quero, S., Banos, R., and Garcia Palacio, A. (2010). Treating cockroach phobia with augmented reality. *Computers in Human Behavior*. 2, 175

Brooks, Jr., F. P. (1996). The computer scientist as toolsmith ii. *Commun. ACM*, 39(3):61–68. 2, 175

Calhoun, G. L., Draper, M. H., Abernathy, M. F., Delgado, F., and Patzek, M. (2005). Synthetic vision system for improving unmanned aerial vehicle operator situation awareness. In *SPIE Enhanced and Synthetic Vision*, pages 219–230. 2

Caudell, T. P. and Mizell, D. W. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, volume ii, pages 659–669 vol.2. 1, 175

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. ii. advanced approaches [tutorial]. *IEEE Robotics Automation Magazine*, 14(1):109–118. 25

Choe, G., Park, J., Tai, Y., and Kweon, I. S. (2016). Refining geometry from depth sensors using IR shading images. *CoRR*, abs/1608.05204. 142

Choi, C. and Christensen, H. I. (2012). Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features. *Int. J. Rob. Res.*, 31(4):498–519. 22

Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619. 81

Comport, A. I., Marchand, E., Pressigout, M., and Chaumette, F. (2006). Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628. 22

Cook, R. L. and Torrance, K. E. (1981). A reflectance model for computer graphics. *SIGGRAPH Comput. Graph.*, 15(3):307–316. 33, 144

Costanza, E., Kunz, A., and Fjeld, M. (2009). Mixed reality: A survey. *Human Machine Interaction*, pages 47–68. 2, 175

Dame, A. and Marchand, É. (2010). Accurate real-time tracking using mutual information. *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 47–56. 24

Dame, A. and Marchand, E. (2012). Second-order optimization of mutual information for real-time image registration. *IEEE Transactions on Image Processing*, 21(9):4190–4203. 58

Debevec, P. (1998). Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and*

Interactive Techniques, SIGGRAPH '98, pages 189–198, New York, NY, USA. ACM. 42, 43, 159

Debevec, P. E. and Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. 36, 37, 42, 48

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*. 127, 132, 136

Drummond, T. and Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946. 22

Duchêne, S., Riant, C., Chaurasia, G., Moreno, J. L., Laffont, P.-Y., Popov, S., Bousseau, A., and Drettakis, G. (2015). Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Trans. Graph.*, 34(5):164:1–164:16. 81

Dutre, P., Bala, K., Bekaert, P., and Shirley, P. (2006). *Advanced Global Illumination*. AK Peters Ltd. 30

Ferwerda, J. A. (2003). Three varieties of realism in computer graphics. *Proc.SPIE*, 5007:5007 – 5007 – 8. 7

Finlayson, G. D., Drew, M. S., and Lu, C. (2009). Entropy minimization for shadow removal. *International Journal of Computer Vision*, 85:35–57. 79

Fuchs, H. and Ackerman, J. (1999). Displays for augmented reality: Historical remarks and future prospects. *Mixed Reality Merging Real and Virtual Worlds, Ohta Y and Tamura H, Ohmsha Ltd*, pages 31–40. 2

Gamerant (2018). https://gamerant.com/pokemon-niantic-occlusion-tech-demo/. 3

Ganz, M., Yang, X., and Slabaugh, G. (2012). Automatic segmentation of polyps in colonoscopic narrow-band imaging data. *IEEE Transactions on Biomedical Engineering*, 59(8):2144–2151. 53

Garding, J. (1992). Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2:630–638. 16

Georgoulis, S., Rematas, K., Ritschel, T., Gavves, E., Fritz, M., Gool, L. V., and Tuytelaars, T. (2018). Reflectance and natural illumination from single-material specular objects using deep learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1932–1947. 42

Gibson, S., Howard, T., and J. Hubbold, R. (2001). Flexible image-based photometric reconstruction using virtual light sources. *Comput. Graph. Forum*, 20. 15

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA. IEEE Computer Society. 126

GoldmanSachs (2016). Virtual and augmented reality. understanding the race for the next computing platform. 3

Gouraud, H. (1971). Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6):623–629. 34, 35

Greenberg, D. P., Cohen, M. F., and Torrance, K. E. (1986). Radiosity: A method for computing global illumination. *The Visual Computer*, 2(5):291–297. 35

Grossberg, M. D. and Nayar, S. K. (2003). Determining the camera response from images: what is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467. 37

Grosse, R., Johnson, M. K., Adelson, E. H., and Freeman, W. T. (2009). Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2335–2342. 81

Gruber, L., Langlotz, T., Sen, P., Höherer, T., and Schmalstieg, D. (2014). Efficient and robust radiance transfer for probeless photorealistic augmented reality. In *2014 IEEE Virtual Reality (VR)*, pages 15–20. 48

Gruber, L., Richter-Trummer, T., and Schmalstieg, D. (2012). Real-time photometric registration from arbitrary geometry. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 119–128. 48, 160

Guo, R., Dai, Q., and Hoiem, D. (2013). Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967. 79, 81, 82, 126, 127, 160

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proc. of Fourth Alvey Vision Conference*, pages 147–151. 24

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition. 22

Herpich, F., Guarese, R., and Tarouco, L. (2017). A comparative analysis of augmented reality frameworks aimed at the development of educational applications. *Creative Education*, pages 1433–1451. 2, 175

Horn, B. K. P. and Brooks, M. J., editors (1989). *Shape from Shading*. MIT Press, Cambridge, MA, USA. 15, 16

Hosseinzadeh, S., Shakeri, M., and Zhang, H. (2017). Fast shadow detection from a single image using a patched convolutional neural network. *CoRR*, abs/1709.09283. 126

Hu, X., Zhu, L., Fu, C.-W., Qin, J., and Heng, P.-A. (2017). Direction-aware spatial context features for shadow detection. *CoRR*, abs/1712.04142. 126, 127, 137

Huang, R. and Smith, W. A. P. (2009). A shape-from-shading framework for satisfying data-closeness and structure-preserving smoothness constraints. In *British Machine Vision Conference*. 16

Iglovikov, V. and Shvets, A. (2018). Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation. *CoRR*, abs/1801.05746. 131, 132

Irani, M. and Anandan, P. (1998). Robust multi-sensor image alignment. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 959–, Washington, DC, USA. IEEE Computer Society. 24

Jachnik, J., Newcombe, R. A., and Davison, A. J. (2012). Real-time surface light-field capture for augmentation of planar specular surfaces. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '12, pages 91–97, Washington, DC, USA. IEEE Computer Society. 41, 52, 62, 159

Jacobs, K. and Loscos, C. (2006). Classification of illumination methods for mixed reality. *Computer Graphics Forum*, 25:29–51. 39, 144, 147, 150

Kajiya, J. T. (1986). The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150. 29

Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., and Bhowmik, A. (2017). Intel(r) realsense(tm) stereoscopic depth cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1267–1276. 83

Khan, S. H., Bennamoun, M., Sohel, F. A., and Togneri, R. (2014). Automatic feature learning for robust shadow detection. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1939–1946. 126

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, Washington, DC, USA. IEEE Computer Society. 23

Knecht, M., Tanzmeister, G., Traxler, C., and Wimmer, M. (2012). Interactive brdf estimation for mixed-reality applications. *Journal of WSCG*, 20(1):47–56. 42, 43, 52, 159

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc. 126

Land, E. H. and McCann, J. J. (1971). Lightness and retinex theory. *Journal of the Optical Society of America*, 61 1:1–11. 47, 81

Leap, M. (2018). Magic leap one headset. https://www.magicleap.com/. 2

Lecocq, P., Dufay, A., Sourimant, G., and Marvie, J.-E. (2016). Accurate analytic approximations for real-time specular area lighting. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, pages 113–120, New York, NY, USA. ACM. 144

Lee, K. J., Zhao, Q., Tong, X., Gong, M., Izadi, S., Lee, S. U., Tan, P., and Lin, S. (2012). Estimation of intrinsic image sequences from image and depth video. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 327–340, Berlin, Heidelberg. Springer-Verlag. 47, 160

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555. 25

Li, Y., Lin, Lu, H., and Shum, H.-Y. (2003). Multiple-cue illumination estimation in textured scenes. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1366–1373 vol.2. 99, 100

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312. 128

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116. 25

Long, J., Shelhamer, E., and Darrell, T. (2014). Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038. 131

Loscos, C., Frasson, M., Drettakis, G., Walter, B., Gainer, X., and Poulin, P. (1999). Interactive virtual relighting and remodeling of real scenes. *Rendering Techniques*, pages 329—340. 5

Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(5):441–450. 22

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. 25

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 24

Mallick, S. P., Zickler, T., Belhumeur, P. N., and Kriegman, D. J. (2006). Specularity removal in images and videos: A pde approach. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 550–563, Berlin, Heidelberg. Springer Berlin Heidelberg. 54

Mandl, D., Yi, K. M., Mohr, P., Roth, P. M., Fua, P., Lepetit, V., Schmalstieg, D., and Kalkofen, D. (2017). Learning lightprobes for mixed reality illumination. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 82–89. 48

Marchand, E., Uchiyama, H., and Spindler, F. (2016). Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2633–2651. 23

Martin, D. R., Fowlkes, C. C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549. 81

Mashita, T., Yasuhara, H., Plopski, A., Kiyokawa, K., and Takemura, H. (2013). Parallel lighting and reflectance estimation based on inverse rendering. In *2013 23rd International Conference on Artificial Reality and Telexistence (ICAT)*, pages 102–107. 40

Maxwell, J., Beard, J., Weiner, S., Ladd, D., and Ladd, S. (1973). Bidirectional reflectance model validation and utilization. Technical report, Environmental research institute of michiganann arbor infrared and optics division. 33

Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2015). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *CoRR*, abs/1512.02134. 128

Mekni, M. and Lemieux, A. (2014). Augmented reality. applications, challenges and future trends. In *International Conference on Applied Computer and Computation Sciences*, pages 205–215. 2, 175

MetaVision (2017). Meta vision headset. https://www.metavision.com/. 2

Microsoft (2016). Hololens headset. https://www.microsoft.com/en-us/hololens. 2

Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77(12):1321–1329. 1, 175

Morgand, A., Tamaazousti, M., and Bartoli, A. (2017). A multiple-view geometric model of specularities on non-planar shapes with application to dynamic retexturing. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2485–2493. 52

Morgand, A., Tamaazousti, M., and Bartoli, A. (2018). A geometric model for specularity prediction on planar surfaces with multiple light sources. *IEEE Transactions on Visualization and Computer Graphics*, 24(5):1691–1704. 52

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006). Real time localization and 3d reconstruction. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 363–370, Washington, USA. IEEE Computer Society. 23

Nayar, S. K. and Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831. 16

Neverova, N., Muselet, D., and Trémeau, A. (2012). Lighting estimation in indoor environments from low-quality images. In Fusiello, A., Murino, V., and Cucchiara, R., editors, *Computer Vision – ECCV 2012. Workshops and Demonstrations*, pages 380–389, Berlin, Heidelberg. Springer Berlin Heidelberg. 47

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352. 144

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011a). Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA. IEEE Computer Society. 23, 24

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011b). Dtam: Dense tracking and mapping in real-time. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2320–2327, Washington, DC, USA. IEEE Computer Society. 23

Nguyen, V., Vicente, T. F. Y., Zhao, M., Hoai, M., and Samaras, D. (2017). Shadow detection with conditional generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4520–4528. 126, 137

Nishino, K., Zhang, Z., and Ikeuchi, K. (2001). Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 599–606 vol.1. 44, 52, 63, 159

OrEl, R., Rosman, G., Wetzler, A., Kimmel, R., and Bruckstein, A. M. (2015). Rgbd-fusion: Real-time high precision depth recovery. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5407–5416. 142

Ortiz, F. and Torres, F. (2006). Automatic detection and elimination of specular reflectance in color images by means of ms diagram and vector connected filters. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36:681–687. 43, 53, 106

Panagopoulos, A., Samaras, D., and Paragios, N. (2009). Robust shadow and illumination estimation using a mixture model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 651–658. 46, 79, 159

Panagopoulos, A., Wang, C., Samaras, D., and Paragios, N. (2011). Illumination estimation and cast shadow detection through a higher-order graphical model. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 673–680, Washington, DC, USA. IEEE Computer Society. 46, 79

Park, Y., Lepetit, V., and Woo, W. (2012). Handling motion-blur in 3d tracking and rendering for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 18:1449–1459. 24

Pattanaik, S. N. and Bouatouch, K. (1994). Fast Wavelet Radiosity Method. *Computer Graphics Forum*. 35

PCL (2013). `https://www.http://pointclouds.org`. 64, 69, 85

Phong, B. T. (1975). Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317. 33, 34, 35, 55, 62, 65, 67, 83, 84, 101, 104, 110, 114, 116, 144

Plopski, A., Mashita, T., Kiyokawa, K., and Takemura, H. (2014). Reflectance and light source estimation for indoor ar applications. In *2014 IEEE Virtual Reality (VR)*, pages 103–104. 45

Polygon (2018). https://www.polygon.com/2018/6/28/17515430/niantic-pokemon-go-pikachu-occlusion-ar-demo. 3

PyTorch (2018). https://pytorch.org/. 133

Ritschel, T., Dachsbacher, C., Grosch, T., and Kautz, J. (2012). The state of the art in interactive global illumination. *Computer Graphics Forum*, pages 160–188. 5

Robertson, M. A., Borman, S., and Stevenson, R. L. (1999). Dynamic range improvement through multiple exposures. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 3, pages 159–163 vol.3. 70

Rohmer, K., Büschel, W., Dachselt, R., and Grosch, T. (2014). Interactive near-field illumination for photorealistic augmented reality on mobile devices. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 29–38. 43

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597. 131, 132

Rosenberg, L. B. (1993). Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 76–82. 1, 2, 175

Rosten, E., Porter, R., and Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):105–119. 25

Roussou, M. and Drettakis, G. (2003). Photorealism and non-photorealism in virtual heritage representation. In *Proceedings of the 4th International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, VAST'03, pages 51–60, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 3, 176

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571. 25

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Li, F. (2014). Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575. 128

Sato, I., Sato, Y., and Ikeuchi, K. (1999). Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. *Proceedings of the ICCV*, 2. 45, 79

Sato, I., Sato, Y., and Ikeuchi, K. (2003). Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):290–300. 45, 159

Schmalstieg, D. and Hollerer, T. (2016). *Augmented reality: principles and practice*. Addison-Wesley Professional. 2

Schmitt, A., Leister, W., and Müller, H. (1988). Ray tracing algorithms-theory and practice. pages 997–1030. 35

Shafer, S. A. (1992). Using color to separate reflection components. *Healey, Glenn E. and Shafer, Steven A. and Wolff, Lawrence B.*, pages 43–51. 39

Sharma, G. (2002). *Digital Color Imaging Handbook*. CRC Press, Inc., Boca Raton, FL, USA. 86

Shen, H.-L. and Cai, Q.-Y. (2009). Simple and efficient method for specularity removal in an image. *Applied optics*, 48 14:2711–9. 40, 45, 47, 54, 68, 159, 160

Shen, L., Chua, T. W., and Leman, K. (2015). Shadow optimization from structured deep edge detection. *CoRR*, abs/1505.01589. 126

Sheng, Y., Shi, Y., Wang, L., and Narasimhan, S. G. (2014). Translucent radiosity: Efficiently combiningdiffuse inter-reflection andsubsurface scattering. *IEEE Transactions on Visualization and Computer Graphics*, 20(7):1009–1021. 35

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556. 133

Smith, S. M. and Brady, J. M. (1997). Susan—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78. 25

Sutherland, I. E. (1968). A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. 2

Tan, R. T. and Ikeuchi, K. (2003). Separating reflection components of textured surfaces using a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):178–193. 40, 54

Tang, A., Owen, C., Biocca, F., and Mou, W. (2003). Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80. 2, 175

Tang, Y., Salakhutdinov, R., and Hinton, G. (2012). Deep lambertian networks. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1419–1426, USA. Omnipress. 41

Teh, C. H. and Chin, R. T. (1989). On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859–872. 107

Thai, B., Deng, G., and Ross, R. (2017). A fast white balance algorithm based on pixel greyness. *Signal, Image and Video Processing*, 11(3):525–532. 105

Thomas, D. (2016). Augmented reality in surgery. the computer-aided medicine revolution. *International Journal of Surgery*, pages 47–68. 2, 175

Tillon, A. B., Marchand, E., Laneurit, J., Servant, F., Marchal, I., and Houlier, P. (2010). A day at the museum: An augmented fine-art exhibit. In *2010 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities*, pages 69–70. 24

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (1999). Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, London, UK, UK. Springer-Verlag. 23

Unity (2018). https://unity3d.com/. 116, 129, 149

Vicente, T. F. Y., Hoai, M., and Samaras, D. (2017). Leave-one-out kernel optimization for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):682–695. 126

Vicente, T. F. Y., Hou, L., Yu, C.-P., Hoai, M., and Samaras, D. (2016). Large-scale training of shadow detectors with noisily-annotated shadow examples. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 816–832, Cham. Springer International Publishing. 127, 128, 129, 137, 138, 155

Vicente, T. F. Y., Yu, C.-P., and Samaras, D. (2013). Single image shadow detection using multiple cues in a supermodular mrf. In *British Machine Vision Conference*. 126

Vorba, J. and Karlik, O. (2012). A survey of data driven methods in realistic image synthesis. In *Annual Conference of Doctoral Students*, pages 82–87. 30

Vuforia (2018). https://www.vuforia.com/. 150

Ward, G. J. (1992). Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272. 89

Wood, D. N., Azuma, D. I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D. H., and Stuetzle, W. (2000). Surface light fields for 3d photography. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 287–296, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. 62

Yu, C., Yao, W., and Bai, X. (2014). Robust linear regression: A review and comparison. *Communications in Statistics - Simulation and Computation*, 46. 90

Zhu, J., Samuel, K. G. G., Masood, S. Z., and Tappen, M. F. (2010). Learning to recognize shadows in monochromatic natural images. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 223–230. 79, 127, 128, 129, 137, 138, 155

Résumé

Vers la fin des années soixante, la création d'images de synthèse réalistes avait fait l'objet de plusieurs projets de recherche. Les travaux réalisés ont abouti à divers algorithmes permettant de générer des rendus incroyablement réalistes de scènes virtuelles entièrement modélisées. Ainsi, la possibilité de créer un monde numérique photoréaliste a contribué, 30 ans plus tard, à l'émergence du premier système de Réalité Mixte (RM) [Rosenberg, 1993]. En 1992, Louis Rosenberg introduit le concept de *montages virtuels*, une superposition d'informations virtuelles sur un espace réel dans le contexte de tâches militaires. L'idée centrale de la superposition ou du mélange d'informations réelles et virtuelles a d'abord été inventée par Tom Caudell [Caudell and Mizell, 1992], puis appelée réalité mixte dans le cadre du continuum réalité-virtualité [Milgram and Kishino, 1994].

Le réalité-virtualité continuum [Milgram and Kishino, 1994] introduit la notion de plusieurs types de réalités. Il va du monde physique réel où aucune donnée virtuelle n'existe à un monde de réalité virtuelle (RV) où tout est virtuel et modélisé. Dans ce continuum, la réalité mixte (RM) a été définie comme "*...tout segment entre les deux extrémités du continuum...*". Elle comprend donc toutes les configurations allant de la réalité augmentée (RA) où le virtuel augmente le réel à la virtualité augmentée (VA), où le réel augmente le virtuel. Dans cette thèse, nous abordons un segment de la réalité mixte permettant d'insérer des objets 3D virtuels dans le monde réel de l'utilisateur tout en se focalisant sur le côté interactif, temps réel et *réaliste* de l'application.

La réalité mixte va sans aucun doute jouer un rôle majeur dans le façonnement de notre avenir proche, non seulement en raison de ses divers cas d'utilisation mais aussi en raison du gain de temps, de productivité et de croissance économique qu'elle apporte. Au fur et à mesure que la technologie impliquée mûrisse, les systèmes basés sur la RM deviennent des produits abordables et conviviaux. Par conséquent, le nombre de scénarios dans lesquels cette technologie peut être utile est *sans limite* [Costanza et al., 2009][Mekni and Lemieux, 2014]. Historiquement, l'objectif primaire de l'information véhiculée par les objets virtuels est de rendre une tâche plus facile à accomplir pour un humain [Brooks, 1996]. Le premier domaine à accueillir ce concept a été le domaine militaire grâce aux travaux de Rosenberg [Rosenberg, 1993] permettant d'améliorer l'expérience télérobotique de l'utilisateur. Depuis, les systèmes de RM ont été utilisés dans divers domaines tels que le médical [Botella et al., 2010][Thomas, 2016], l'éducation [Herpich et al., 2017] et la formation industrielle [Tang et al., 2003]. Cette thèse appartient au domaine du divertissement et s'inscrit dans le cadre d'un partenariat industriel

CIFRE[1] avec Technicolor, une multinationale qui fournit des services et des produits aux industries du divertissement, et l'Institut de Recherche en Informatique et Systèmes Aléatoire (IRISA).

Dans ce contexte, bien que plusieurs acteurs industriels aient déjà proposé des solutions (ARKit, Vuforia, Wikitude) capables d'aligner géométriquement le monde réel avec le monde numérique, aucun d'entre eux n'a abordé le problème d'un mélange homogène et *réaliste*. En effet, lors de l'utilisation des systèmes existants, les objets virtuels se distinguent souvent facilement des objets réels à cause de leurs apparences incohérentes. Comme l'immersion de l'utilisateur représente un aspect important de ces systèmes [Roussou and Drettakis, 2003], cela pose un problème majeur. Les repères visuels humains sont sensibles à la cohérence globale d'une image. L'absence des ombres virtuelles ou la perception confuse des couleurs causée par un objet virtuel excessivement lumineux sont des éléments pouvant empêcher l'utilisateur d'interagir et de s'engager dans une application cible.

Afin d'apporter une solution à cette problématique, le scénario considéré dans le cadre de ce travail est le suivant: à l'aide d'une caméra RVB-P produisant des images couleurs et des cartes de profondeur, nous parcourons une scène réelle d'intérieur pour en acquérir la géométrie. L'introduction de sondes lumineuses ou la necessité d'une interaction excessive de l'utilisateur ne sont guère envisagées. En analysant uniquement les images couleur et le modèle acquis de la scène, les approches proposées permettent d'estimer les propriétés de réflectance de surface et les caractéristiques d'illumination (position 3D, intensité, couleur). Notre but étant de gérer une grande variété de scènes d'intérieur réelles, nous considérons très peu de contraintes relatives au contenu de la scène. Plus précisément, nous ne supposons guère que la scène est composée d'objets 3D de couleur uniforme ni que l'éclairage est éloigné de la scène ou réduit à une seule source lumineuse. Enfin, les sources lumineuses peuvent être dynamiques et nos approches doivent tenir compte de ces changements dans l'éclairage et le rendu final des objets virtuels. Dans ce travail, nous sommes intéressés par le réalisme fonctionnel plutôt qu'un réalisme physiquement simulé. Notre but est de produire un mélange convaincant et esthétique entre le réel et le numérique.

Dans la littérature, les approches existantes introduisent souvent soit des dispositifs supplémentaires (une sphère chromée et/ou une caméra équipée d'un objectif fish-eye) qui peuvent être encombrants dans l'expérience-utilisateur, soit contraignent le contenu de la scène (une scène réduite à un seul objet, un éclairage représenté par une seule source lumineuse, des surfaces Lambertiennes). De plus, elles supposent souvent que l'éclairage de la scène est statique. Cela introduit une contrainte forte étant donné que l'utilisateur peut avoir besoin de changer l'éclairage de son environnement. Dans cette thèse, nous avons abordé ces limites en tirant parti de l'information RVB-P acquise.

Pour réaliser le recalage photométrique, nous avons considéré quatre principaux axes de recherche: (1) Recalage photométrique utilisant des réflexions spéculaires. (2) Re-

[1]Convention Industrielle de Formation par la Recherche

calage photométrique utilisant des ombres portées. (3) Recalage photométrique utilisant à la fois des réflexions spéculaires et des ombres portées. (4) Détection de réflexions spéculaires et d'ombres portées de scènes réelles intérieur à l'aide d'une approche d'apprentissage profond. Ces quatre axes de recherche ont donné lieu à quatre contributions principales détaillées ci-après.

(1) Recalage photométrique utilisant des réflexions spéculaires.

Dans une première contribution, nous considérons des scènes intérieures réelles où la géométrie et l'éclairage sont statiques. Au fur et à mesure que le capteur parcourt la scène, des réflexions spéculaires peuvent être observées à travers une séquence d'images RVB-P. Ces repères visuels sont très instructifs sur l'éclairage et la réflectance de la scène et ont longtemps été pris en compte dans les approches de recalage photométrique. Dans ce contexte, les techniques existantes estiment souvent les spécularités comme des régions saturées dans l'image. Par conséquent, les surfaces brillantes et blanches peuvent être considérées à tort. De plus, on suppose souvent que les sources lumineuses sont éloignées et seules leurs directions sont estimées. Cependant, dans les scènes réelles d'intérieur, cette hypothèse n'est pas toujours valable. Notre première contribution aborde ces limites. Plus précisément, nous considérons des scènes réelles arbitraires composées d'un ou plusieurs objets aux textures variées. Nous estimons les propriétés de réflectance diffuse et spéculaire à l'aide d'une analyse spatio-temporelle robuste de la séquence RVB-P acquise. Cette analyse repose sur la construction de profils de luminance qui conservent l'évolution de la luminance d'un pixel donné lors de mouvement de la caméra. De plus, nous estimons la position 3D de plusieurs sources de lumière sans aucune intervention de l'utilisateur. Nos estimations ont été intégrées dans un pipeline de rendu présentant des scénarios réalistes de RM tels que des objets virtuels occultant correctement des spécularités réelles (reconstruction précise de la région de spécularité en utilisant la réflectance diffuse estimée) ainsi que des ombres virtuelles réalistes en termes de forme et d'intensité.

(2) Recalage photométrique utilisant des ombres portées.

Dans cette contribution, l'analyse est basée sur les ombres portées observées dans la scène. Les ombres sont omniprésentes et résultent de l'occutation de la lumière par la géométrie existante. Elles représentent donc des indices intéressants pour reconstituer les propriétés photométriques de la scène. Lorsqu'il s'agit de scènes d'intérieur, les solutions existantes supposent souvent la présence de surfaces de couleur uniforme pour détecter les ombres. La présence de texture dans ce contexte est un scénario difficile. En effet, la séparation de la texture et des effets d'éclairage est souvent traitée par des approches qui nécessitent une grande interaction de l'utilisateur (l'indication de l'emplacement des ombres) ou qui ne répondent pas aux exigences de la réalité mixte (quelques minutes pour détecter les ombres dans une seule image). Dans cette contribution, nous présentons une méthode qui aborde ces contraintes. L'approche proposée est double: nous séparons d'abord la texture et l'éclairage en considérant des paires de points ayant la même propriété de réflectance mais soumis à des conditions d'éclairage différentes. Ensuite, à partir de l'illumination estimée, nous obtenons la position 3D

et l'intensité des sources lumineuses via un processus itératif. Notre méthode permet de gèrer également l'éclairage dynamique et fonctionne à une fréquence d'image interactive (4 images par seconde). Par conséquent, elle est adaptée aux scénarios RM où l'utilisateur peut librement allumer, éteindre et déplacer les sources lumineuses.

(3) Recalage photométrique utilisant à la fois des réflexions spéculaires et des ombres portées.

Dans cette contribution, nous abordons le problème de l'estimation de l'illumination et de la réflectance en analysant conjointement les réflexions spéculaires et les ombres portées. L'approche proposée tire parti de l'information apportée par les deux indices visuels pour traiter une grande variété de scènes. Par exemple, les ombres portées faibles sont difficiles à détecter en utilisant uniquement des approches basées sur les ombres; cependant, lorsque des réflexions spéculaires sont disponibles, il est possible de combiner efficacement les deux informations pour estimer l'éclairage. Dans cette contribution, nous proposons une méthode qui permet d'estimer la position et la couleur de sources lumineuses multiples dans la scène. Notre approche est capable de traiter n'importe quelle surface texturée et prend en compte à la fois les sources de lumière statiques et dynamiques. Son efficacité est démontrée par une gamme d'applications, y compris des scénarios de réalité mixte en temps réel où le rendu d'objets synthétiques est cohérent avec l'environnement réel (une occultation de spécularité réelle par un objet virtuel, des ombres virtuelles visuellement cohérentes) et la re-texturation où la texture de la scène est modifiée tandis que l'éclairage incident est conservé.

(4) Détection de réflexions spéculaires et d'ombres portées de scènes réelles intérieures par une approche d'apprentissage profond.

Dans les contributions mentionnées précédemment, nous avons exploré des approches permettant de détecter et modéliser efficacement les réflexions spéculaires et les ombres portées afin d'obtenir le recalage photométrique des scènes réelles. Une dernière contribution de cette thèse a été de proposer un cadre d'apprentissage en profondeur pour détecter conjointement les spécularités et les ombres portées dans les scènes. Dans le cadre de ce type d'approches, un facteur clé de la généralisation consiste à disposer d'un ensemble de données avec une grande variété de scénarios. En ce qui concerne notre tâche cible, les bases de données relatives à la détection de réflexions spéculaires ne sont pas disponibles et la majorité des bases de données relatives à la détection d'ombres prennent en compte les scènes extérieures où le soleil est la seule source lumineuse. Aussi, nous avons construit une base de données complète et exhaustive dans le but de traiter des scénarios de scènes réelles intérieures et extérieures. Notre technique a été testée sur une variété d'images et produit de bons résultats.

Titre : Recalage Photométrique de Scènes Réelles d'Intérieurs à l'aide d'une Caméra RGB-D avec Application à la Réalité Mixte

Mots clés : Illumination, réflectance, diffuse, spéculaire, texture, ombre, réalité mixte

Résumé :

L'objectif principale de la Réalité Mixte (RM) est de donner aux utilisateurs l'illusion que les objets virtuels et réels coexistent indistinctement dans le même espace. Une illusion efficace nécessite un recalage précis entre les deux mondes. Ce recalage doit être cohérent du point de vue géométrique et *photométrique*. Dans cette thèse, nous proposons de nouvelles méthodes de recalage photométrique pour estimer l'illumination et la réflectance de scènes réelles. Plus précisément, nous proposons des approches en nous attaquant à trois grands défis : (1) utilisation d'une seule caméra RGB-D. (2) estimation des propriétés de réflectance diffuse et spéculaire. (3) estimation de la position 3D et de la couleur de sources lumineuses dynamiques multiples.

Dans notre première contribution, nous considérons des scènes réelles d'intérieurs où la géométrie et l'éclairage sont statiques. En observant la scène à partir d'une caméra mobile, des réflexions spéculaires peuvent être détectées tout au long de la séquence d'images RGB-D. Ces indices visuels sont très instructifs sur l'éclairage et la réflectance des surfaces des scènes. Par conséquent, nous les modélisons pour estimer à la fois les propriétés de réflectance diffuse et spéculaire ainsi que la position 3D de sources lumineuses multiples. Notre algorithme permet d'obtenir des résultats de RM convaincants tels que des ombres virtuelles réalistes ainsi qu'une suppression correcte de la spécularité réelle.

Les ombres sont omniprésentes et représentent l'occultation de la lumière par la géométrie existante. Elles représentent donc des indices intéressants pour reconstituer les propriétés photométriques de la scène. La présence de texture dans ce contexte est un scénario critique. En effet, la séparation de la texture et des effets d'éclairage est souvent gérée par des approches qui nécessitent l'intervention de l'utilisateur ou qui ne répondent pas aux exigences du temps de traitement de la réalité mixte. Nous abordons ces limitations et proposons une méthode d'estimation de la position et de l'intensité des sources lumineuses. L'approche proposée gère les lumières dynamiques et fonctionne en temps quasi-réel.

L'existence d'une source lumineuse est plus probable si elle est soutenue par plus d'un indice visuel. Nous abordons donc le problème de l'estimation des propriétés d'éclairage et de réflectance en analysant conjointement les réflexions spéculaires et les ombres projetées. L'approche proposée tire parti de l'information apportée par les deux indices pour traiter une grande variété de scènes. Notre approche est capable de traiter n'importe quelle surface texturée et tient compte à la fois des sources lumineuses statiques et dynamiques. Son efficacité est démontrée par une gamme d'applications, incluant la réalité mixte et la re-texturation.

La détection des ombres projetées et des réflexions spéculaires étant au cœur de cette thèse, nous proposons finalement une méthode d'apprentissage approfondi pour détecter conjointement les deux indices visuels dans des scènes réelles d'intérieurs.

Title: Photometric Registration of Indoor Real Scenes using an RGB-D Camera with Application to Mixed Reality

Keywords: Illumination, reflectance, diffuse, specular, texture, shadow, mixed reality

Abstract:

The overarching goal of Mixed Reality (MR) is to provide the users with the illusion that virtual and real objects coexist indistinguishably in the same space. An effective illusion requires an accurate registration between both worlds. This registration must be geometrically and *photometrically* coherent. In this thesis, we propose novel photometric registration methods to estimate the illumination and reflectance of real scenes. Specifically, we propose new approaches which address three main challenges: (1) use of a single RGB-D camera. (2) estimation of both diffuse and specular reflectance properties. (3) estimation of the 3D position and color of multiple dynamic light sources.

Within our first contribution, we consider indoor real scenes where both geometry and illumination are static. As the sensor browses the scene, specular reflections can be observed throughout a sequence of RGB-D images. These visual cues are very informative about the illumination and reflectance of scene surfaces. Hence, we model these cues to recover both diffuse and specular reflectance properties as well as the 3D position of multiple light sources. Our algorithm allows convincing MR results such as realistic virtual shadows and correct real specularity removal.

Shadows are omnipresent and result from the occlusion of light by existing geometry. They therefore represent interesting cues to reconstruct the photometric properties of the scene. Presence of texture in this context is a critical scenario. In fact, separating texture from illumination effects is often handled via approaches which require user interaction or do not satisfy mixed reality processing-time requirements. We address these limitations and propose a method which estimates the 3D position and intensity of light sources. The proposed approach handles dynamic light sources and runs at an interactive frame rate.

The existence of a light source is more likely if it is supported by more than one cue. We therefore address the problem of estimating illumination and reflectance properties by jointly analysing specular reflections and cast shadows. The proposed approach takes advantage of information brought by both cues to handle a large variety of scenes. Our approach is capable of handling any textured surface and considers both static and dynamic light sources. Its effectiveness is demonstrated through a range of applications including real-time mixed reality and retexturing.

Since the detection of cast shadows and specular reflctions are at the heart of this thesis, we further propose a deep-learning framework to jointly detect both cues in indoor real scenes.