TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Informatik
der Technischen
Universität München

Dissertation

Leveraging the User's Face as a Known Object in Handheld Augmented Reality

Sebastian Bernhard Knorr

Institut für Informatik
der Technischen
Universität München

Leveraging the User's Face as a Known Object in Handheld Augmented Reality

Sebastian Bernhard Knorr

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Thomas Huckle
Prüfer der Dissertation:	1.	Prof. Gudrun Johanna Klinker, Ph.D.
	2.	Prof. Dr. Daniel Cremers

Die Dissertation wurde am 15.11.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 03.01.2017 angenommen.

Zusammenfassung

Um virtuelle Gegenstände in erweiterter Realität, englisch Augmented Reality (AR), wirklichkeitsgetreu in die Ansicht der realen Umgebung einzublenden, werden Kenntnisse über diese Umgebung benötigt: zum Beispiel die aktuelle Beleuchtungssituation, um virtuelle Gegenstände dementsprechend zu beleuchten, und die Abmessungen der Umgebung in absoluten Einheiten wie Meter, um virtuelle Gegenstände korrekt zu skalieren. Wir schlagen vor, das Gesicht des Benutzers als ein bekanntes Objekt zu verwenden und von Bildern des Gesichts die Beleuchtung und Skalierung der realen Welt abzuleiten.

Aktuelle Methoden sind entweder auf zusätzlich in die Szene gestellte bekannte Objekte angewiesen, benötigen Spezialausstattungen wie Tiefenkameras oder Fischaugenlinsen oder beinhalten umständliche Aufnahmeschritte. Für den normalen Benutzer von tragbaren Geräten, wie z.B. Smartphones, stellen diese Methoden Hürden dar. Das Gesicht des Benutzers kann dagegen jederzeit bequem mit der Frontkamera aktueller tragbarer Geräte aufgenommen werden.

Form und Reflexionsvermögen von Gesichtern unterscheiden sich nur begrenzt zwischen verschiedenen Menschen. Sie lassen sich daher mittels Durchschnittsmodellen approximieren. Diese Modelle können im Voraus erstellt werden und ermöglichen Methoden, die in Echtzeit und mit einfachen monokularen Kameras aktueller tragbarer Geräte arbeiten.

Als erstes verwenden wir das Gesicht zur Lichtschätzung. Wir lernen unter Verwendung einer Bilddatenbank mit verschiedenen Gesichtern unter verschiedenen bekannten Beleuchtungsituationen, wie das menschliche Gesicht einfallendes Licht aus einer bestimmten Richtung zur Kamera hin reflektiert. Dies ermöglicht es uns, anschließend auf Basis eines Bildes des Gesichts des Benutzers in Echtzeit das gegenwärtig einfallende Licht zu schätzen. Wir belegen die Leistungsfähigkeit unserer Methode zur Lichtschätzung sowohl durch Vergleiche der Schätzung mit der tatsächlichen Beleuchtung als auch anschaulich durch Bilder und Videosequenzen, in denen die geschätzte Beleuchtung auf die eingeblendeten virtuellen Gegenstände angewendet wird.

Als zweites verwenden wir die bekannte Größe des Gesichts, um die absolute Skalierung der Rekonstruktion der realen Welt, die aufgrund einer monokularen Kamera nicht bekannt ist, zu bestimmen. Indem der Augenabstand des Benutzers, entweder als statistischer Mittelwert oder durch einmalige Vermessung, in absoluten Einheiten wie Millimetern angegeben wird, kann auch die Pose (Position und Orientierung) der Frontkamera bezüglich des Gesichts in absoluten Einheiten bestimmt werden. Die Bewegung der Frontkamera kann auf die Bewegung der fest verbundenen Hauptkamera übertragen werden, welche die reale Welt vor dem Benutzer verfolgt. Unter der Annahme, dass sich das Gesicht bezüglich der Umgebung nicht bewegt, kann so die Skalierung der Rekonstruktion der Umgebung bestimmt werden. Die bekannte Skalierung erlaubt neben der Darstellung virtueller Gegenstände in richtiger Größe auch die Vermessung von Strecken in der realen Umgebung. Wir werten die Leistungsfähigkeit unserer Methode zur Skalierungsschätzung in Hinblick auf Genauigkeit und Präzision aus und zeigen, dass die erzielten Ergebnisse viele Anwendungsfälle ermöglichen.

Die beiden vorgestellten Ansätze zur Bestimmung von Beleuchtung und Skalierung der realen Welt stellen zum einen zwei neue Methoden dar, die auf den nicht professionellen Markt tragbarer erweiterter Realität zugeschnitten sind und durch ihre einfache Ausführbarkeit ohne zusätzliche Anforderungen an Hardware bisherige Einschränkungen existierender Ansätze beseitigen. Zum anderen verdeutlichen die beiden Ansätze auch das generelle von dieser Dissertation aufgezeigte Konzept, welches als Grundlage für zukünftige Forschung dienen kann: die Verwendung des Gesichts des Benutzers als bekannten Gegenstand zur Gewinnung von Informationen über die reale Umgebung.

Abstract

In Augmented Reality (AR) knowledge about the real environment is crucial to realistically embed renderings of virtual objects in the view of the real world. Knowledge comprises e.g. the illumination present in the real world to light virtual objects coherently. It also comprises the dimensions of the real environment in absolute units like meters to overlay virtual objects at proper size. We propose to leverage the face of the user as a known object in order to deduce information about the illumination and scale of the real environment based on images of the face.

State-of-the-art approaches either rely on additional known objects that have to be put into the scene, they rely on specialized hardware like depth cameras or fisheye lenses, or they require cumbersome capture procedures. All of that poses barriers for nonprofessional users of handheld AR applications. The face of the user in contrast can be conveniently captured by the user-facing camera of current handheld devices at any time.

As the shape and reflectance of faces vary only moderately among different humans, the user's face can be approximated by average models of shape and reflectance. These models can be built in advance and enable approaches that run in real time on images from simple monocular cameras of current handheld devices.

Firstly we leverage the face as a light probe. Using an image dataset with different faces under different known illuminations we learn how the average face reflects light incident out of a certain direction towards the camera. This enables us to subsequently estimate the present real-world illumination incident on the user's face from an image of the face in real time. We demonstrate the effectiveness of our light estimation method by comparing the estimated illumination against ground truth as well as perceptually by presenting augmented image and video footage where virtual objects are lit with the estimated illumination.

Secondly we leverage the face as an object of known size, which resolves the ambiguity in scale stemming from the reconstruction of the real world by a monocular camera. By specifying the distance between the user's eyes in absolute units – either by simply using the statistical mean value or by once measuring that distance – the pose (position and orientation) of the user-facing camera with respect to the face can be determined in absolute units like meters. In a handheld AR scenario, the motion of the user-facing camera can be transferred to the motion of a rigidly connected world-facing camera that is used to track the real world in front of the user. Under the assumption that the face remains stationary in the real world, the absolute scale of the real-world reconstruction can be determined. Knowing the scale of the reconstruction does not only allow rendering virtual objects at correct size but also performing distance measurements within the real world. We evaluate the performance of our scale estimation method in terms of accuracy and precision and demonstrate that the results are sufficient for many use cases.

The two presented solutions for estimating the illumination and absolute scale of the real world do not only enable new methods that are tailored to the nonprofessional handheld AR market, because they overcome limitations of state-of-the-art approaches and are simple to perform without additional hardware requirements. The two methods also showcase the much broader idea presented in this thesis which may serve as a foundation for future research: to leverage the user's face as a known object to deduce information about the real world.

Acknowledgements

First of all, I would like to thank Prof. Gudrun Klinker, Ph.D., for supervising my thesis as well as for kindly welcoming me at her research group FAR in Garching where she offered me a workplace for the four months I spent there in person. In this connection I would also like to thank Prof. Dr. Daniel Cremers for agreeing to serve as second examiner of this thesis.
I am especially thankful to Dr. Daniel Kurz who is not only the advisor for this thesis but also the second author of all my publications. I learned a lot from him about effective research as well as about successful publishing. Being my manager during the past years at Metaio, he also made it possible for me to work on research topics relevant for this thesis.
Of course I also want to say thank you to Dr. Thomas Alt and Peter Meier who believed in my research proposal and gave me the opportunity to pursue my doctoral degree on the job at Metaio. In particular I want to express my gratitude to them for supporting me in my ambition and persistently working on making this thesis possible even when the circumstances became more complicated. In the same way I want to say thank you to my former colleagues at Metaio. This thesis would not have been possible in its existing form without their outstanding work, assistance and support.
I am also very grateful to Dr. Holger Dammertz who sparked my interest in research and computer graphics during my studies at Ulm University and who also advised me to apply at Metaio.
Most importantly I want to say thank you to my parents Dr. Klaus Knorr and Dr. Antje Knorr for always encouraging me in my studies, and above all to my patient wife Anja Knorr who always supports me in whatever I do and – even more – lovingly pushes me towards my goals.
Finally I appreciate the support by the German Federal Ministry of Education and Research (BMBF, reference number 01IM13001L, ARVIDA).

Contents

List of Figures

List of Tables

1 Introduction

In Augmented Reality virtual content that is related to the real world is added to our view of the real-world surroundings. For seamlessly integrating this virtual content into the view of the real world a good understanding of the real environment, e.g. in terms of geometry and illumination, is essential. If the environment, which also is referred to as scene, is not already known in advance, a comprehensive on-the-fly reconstruction of the environment is needed. When the scene is reconstructed based only on images from a simple monocular RGB camera as it is available in current handheld devices, some parts of the reconstruction problem however are under-determined, e.g. appraising the absolute scale of the geometric reconstruction or estimating the illumination present in the scene. Most existing approaches that address these tasks in the context of Augmented Reality thus demand either additional specialized hardware like for example a depth camera or they rely on known objects that need to be captured by the camera. The latter thereby assume that the special objects are at hand and often involve laborious extra set-up and capture steps. We instead propose to leverage the face of the user as a known object, which is part of the scene anyway. Many available set-ups in video see-through Augmented Reality already feature a user-facing camera, so that taking an image of the user's face is straight forward. Additionally the human face is limited in its variations, so that average properties can be used as approximation for a particular user. With this idea we overcome common problems like the need for known special objects and laborious extra steps while still keeping the hardware requirements low.

The goal of this dissertation is to explore the idea of employing the face of the user as a known object in the context of scene reconstruction for Augmented Reality (AR) applications. Relying on the face supersedes specialized hardware and additional objects that are required by current approaches. By

that it enables new, light-weight, and easy-to-use methods for reconstructing the real world that are especially well suited for non-professional users.

In this introductory chapter we will start in section 1.1 by giving the context for the conducted research. We outline the idea of AR, to display virtual content, that is related with the real world, directly within our view of the world. In section 1.2, we then will document the rationale why a reconstruction of the real world is important for a seamless integration of virtual content into the view of the real world and will bring up existing challenges. In this thesis we will in particular pitch on two problems in scene reconstruction using a monocular camera: estimating the present illumination, and estimating the absolute scale of the real-world reconstruction. In section 1.3 we will convey our idea to leverage the face of the user as a known object to support the reconstruction of the scene and will already provide a short preview on how we address the two problems of scale and illumination estimation by using the face of the user as a known object. We sum up the main contributions of this thesis in section 1.4, bring up involved collaboration partners in section 1.5, and enumerate accompanying publications and filed patents in section 1.6. This introductory chapter then concludes with section 1.7, which describes the organizational structure of this thesis.

1.1 Augmented Reality

In Augmented Reality (AR) our perception of the real-world surroundings is enriched by additional digital content – ranging from purely textual information to complex 3-dimensional virtual objects. Which digital content is augmented where depends on what is visible in the real world as virtual content is spatially registered with the real world.

Technologically AR often is implemented as video see-through AR, where the view of the real-world environment is provided in form of a live-video stream that is captured by a camera and that is presented to the user on a display. The digital content is embedded by overlaying computer-generated renderings of the content on top of the video stream.

The content often comprises virtual 3-dimensional objects which act as surrogates for real objects. This can for example be used for product previews to give the user the impression how an actual object would look like in the surroundings. Popular examples comprise placing furniture in your room or virtually trying on jewelry or glasses. Other common applications range from games and advertising to work-flow assistance and visualizations in the industrial and medical domain.

A platform that is well suited for enabling video see-through AR applications for the mass market are smart phones. These devices have become very popular over the last decade. In 2015 for example more than 90% of the population in Germany at the age between 14 and 24 used a smartphone on a regular basis for mobile Internet access [Stat 16]. Smart phones thereby have become companions of our daily life, that we tend to always carry with us, so that these devices are ubiquitously available.

Being mobile computers equipped with camera, display, and mobile Internet, smart phones feature all the basic hardware requirements for video see-trough AR.

Beside smart phones also mobile computers like notebooks or tablets have become more and more popular. Most of these devices also at least feature a user-facing camera (web cam). This makes them applicable for video see-through AR applications on the image of that camera for example in the domain of virtual try-on or video chat.

While all these devices feature what is basically needed for AR, they are originally designed for general purpose and not specifically targeted at AR applications. The majority of devices for example only feature monocular intensity cameras and lack specialized sensors like depth or stereo cameras that would simplify 3-dimensional data acquisition.

In order to support an as broad as possible audience of users for an AR application it is thus important to pay attention to the hardware components and processing power that are available to the consumers and to tailor to that the particular approaches and techniques.

While with the predicted raise of AR in the future also the mass market devices will more and more focus on AR and hence these restrictions will diminish, the raise of AR itself will depend on the adoption by the market at present.

1.2 Scene Reconstruction in Augmented Reality

In traditional computer graphics all the definitions that are needed for generating an image are already given. The definitions may comprise the camera pose (i.e. the position and orientation of the camera), all the geometries (e.g. 3-dimensional triangle meshes) and materials (like textures and other rendering parameters like specularity) of the virtual objects, as well as all the light source specifications. These definitions thus can be directly used as input for the rendering stage.

In AR often, however, only the digital content is known in advance. The real-world part depends on where and when the AR application is executed and all the information that is known about the real world is provided by the captured images thereof. These images then shall be merged with renderings of the digital content. Especially for applications where the virtual 3-dimensional objects in the augmented view act as surrogates for real objects it is clearly beneficial when the renderings of the virtual objects *seamlessly* integrate in the view of the real world. Thereby the augmented image shall strongly resemble a fictitious real image, so that the user has the impression that the synthetic objects are actually placed within the real world. To coherently and seamlessly compose renderings of virtual objects with the view of the real world, knowledge about the surrounding environment is crucial. As this information is not known in advance, it thus must be first acquired and reconstructed.

1.2.1 Camera Pose Determination at Absolute Scale

The current camera pose in relation to the real world must for example be determined in order to display the virtual objects under an adequate perspective that matches the background image of the real world. Even when the real world, e.g. the room of a user, is unknown to the application at start up, it still shall be possible to determine the camera motion with respect to this previously unknown environment, in order to allow placing virtual objects into this environment. For that the geometry of the real-world environment must be reconstructed during run time.

A famous method that allows for building a (sparse) 3-dimensional model of an unknown scene while simultaneously delivering the current camera pose in relation to the scene is monocular SLAM [Davi 07]. SLAM stands for Simultaneous Localization And Mapping. Monocular SLAM runs on the image stream of a single intensity camera and thus is applicable even on standard smart phones. The resulting reconstruction from monocular SLAM however is created at an arbitrary scale factor with an unknown relation to real-world metrics like meters. This ambiguity in scale results from the fact that images captured of the scene only measure a projection of the scene. A camera pose that is determined from this reconstruction inherits the arbitrary scale factor of the reconstruction. Therefore, when this pose is used for rendering virtual objects, also the augmentations will appear at arbitrary size.

Very popular AR use cases include virtually placing a piece of furniture in the living room to test if it would spatially fit in the room and how it would visually match its surroundings. Here it is indispensable that the virtual objects are presented at correct scale. To display a virtual object in real physical dimensions, the camera pose must be determined at absolute scale.

Different approaches exist to bring the reconstruction and thus also the augmentations to correct size. Some rely on objects or markers of known size [Davi 07] that have to be put into the scene and captured by the camera to determine the correct scale. Others employ additional sensors, e.g. the accelerometer sensor of the smart phone [Tans 13] that is able to measure the linear acceleration of the device in absolute units. This then allows to transfer the knowledge about the camera motion to the reconstruction of the scene. Even other approaches rely on special cameras [Lieb 11, Kerl 13] that deliver depth values at correct scale. All these approaches limit the convenience for the user by adding additional requirements. Either in terms of available items like known objects, markers or special hardware like depths cameras. Or in terms of requiring the user to perform specific tasks like capturing the known objects or moving the device in certain unintuitive ways to e.g. exploit the readings from the accelerometer sensors.

1.2.2 Augmentations Lit with Real-World Illumination

If the goal is not only to display the virtual objects at the correct spot, orientation, and size in the real world but to *photo-realistically* embed the objects into the camera stream, also the illumination present

in the real world must be considered and applied for the renderings, so that the virtual objects are illuminated with the same lighting conditions visible in the real-world environment. If the scene is for example lit by sun light from one specific direction, also the virtual objects should be lit accordingly. Inconsistent illumination manifests for example in improper coloring as well as wrong positions of highlights and cast shadows and thereby disrupts the realistic impression. The illumination present in the real world however is unknown and thus first must be determined – preferably in real time at the time of augmentation.

Some methods therefore acquire omni-directional images of the surroundings to capture the incident light. They either rely on mirror spheres [Debe 98] that need to be put into the scene and need to be captured by the camera. Or they rely on an additional fish-eye camera [Sato 99] capturing the surroundings, which however requires the user to either have a mirror sphere or an additional fish-eye camera available. Other approaches stitch together multiple images [DiVe 08] taken of the surroundings. This however is a quite tedious and cumbersome process.

An alternative for determining the present illumination is to estimate the incident light from the video image of the real world used for the augmented view. While this poses less duties on the user, this problem in general is ill-posed and underdetermined. There is an ambiguity between the unknown material and geometry of the scene and the unknown illumination. Some approaches try to reduce the ambiguity by either using depth cameras [Grub 12] or by relying on known objects [Arie 12] that have to be additionally put into the image of the camera. Like in the case of scale estimation, this however adds additional requirements for the user either in terms of available hardware or in terms of additionally required tasks.

1.2.3 Targeting the Mass Market

This thesis targets AR applications for non-professional users.

The proposed approaches for reconstructing the real world thus need to work on hardware which is available at the mass market. In terms of cameras, we will only require simple monocular intensity cameras without any depth sensor. Our developed algorithms also need to feature a low impact on power consumption, which is especially important in the field of mobile AR.

Beside low hardware requirements, our focus lies on easy use. Unlike many existing methods, we do not expect the user to carry with them any additional markers or objects. Also we strive to avoid cumbersome and tedious set-up or calibration steps. We aim for fast and intuitive methods running in real time.

1.3 The Face of the User As a Known Object

As explained above, some problems in scene reconstruction are ill-posed and under-determined when working only on the images of a monocular intensity camera. One popular way to tackle this hurdle is the addition of known objects to the scene. The existing knowledge about these objects then can be used to make the reconstruction problem solvable for example in the field of illumination estimation or scale determination. We however do not want to add additional objects. We thus in section 1.3.1 have a look at what is already available in video see-through AR and identify the face of the user as well suited known object.

In this thesis we will in particular leverage the user's face as a known object to address two specific problems in the domain of scene reconstruction: estimating the present illumination and estimating the absolute scale of the real-world reconstruction. We provide short previews on our approaches in section 1.3.2 and section 1.3.3 respectively. The two examples illustrate the broad field of possible applications of our idea.

1.3.1 The Face of the User in Video See-Through AR

AR is a technology for human users. In video see-through AR we always have the user positioned in front of a screen observing the presented augmented view. The user thus is always part of the scene and is facing the screen. Many hardware set-ups like for example smart phones also comprise a user-facing camera, so that an image of the user's face can be acquired at any time.

Based on this awareness we will in this dissertation thesis aim at leveraging the user's face as a known object and we will in particular elaborate how images of the user's face captured by the user-facing camera can be exploited to deduce information about the real world in terms of present illumination and absolute scale.

Two facts thereby allow us to treat the user's face as a known object.

Firstly, even though the appearance of the human face varies significantly amongst different people, which for example enables us to recognize people from an image of their face, all the faces of different humans still have a lot in common. Human faces exhibit properties that vary only within a limited range, e.g. the spatial dimensions of facial fiducials and their appearance or the reflectance properties of human skin. This limited range over different humans allows generating models for the respective properties which then can be applied for a multitude of humans. The human face thereby becomes an (at least approximately) *known object*. Well-established applications thereof are face detection and face tracking, where an algorithm locates a face within an image and even determines the pose of the face with regard to the camera although the particular face is potentially unknown to the algorithm.

Secondly, the appearance of a particular user's face can also be once calibrated and then reused whenever this user is again running the AR application. The calibration can for example be stored on

the user's device. Also multiple calibrations of different faces can be stored either locally or online and the correct calibration can be selected using a face detection for the current user. A calibration for a specific face and its properties may increase the accuracy of the methods building upon.

In this thesis we will leverage the user's face as a known object to address two particular problems in the domain of scene reconstruction. First we will exploit limited variations in terms of shape and reflectance properties of human faces to *estimate the incident illumination* based on a single image of the user's face which for AR enables a coherent illumination of the virtual objects. A preview on our approach is given in section 1.3.2. Afterwards we will exploit limited variations in terms of spatial physical dimensions between different human faces to *bring a geometric reconstruction* of the real world from arbitrary scale *to absolute scale* using images of the user's face. This enables augmentations of virtual objects at correct size. A preview on our approach is given in section 1.3.3.

1.3.2 Light Estimation from the User's Face

A highly realistic augmented view requires that the illumination of the virtual objects matches the illumination of the real world. It thus is desirable to determine the present real-world illumination. State of the art approaches for acquiring the real-world lighting conditions often either use extra objects like markers or mirror spheres that need to be added to the scene as light probes or they require special hardware like depth or fish eye cameras.

We in here present an alternative approach. We employ the user's face, which is already located in the scene, as a light probe and estimate the real-world lighting conditions in real time from a single monocular image of the user's face. This allows a coherent illumination of virtual objects in AR. We thereby neither require that the user has available any additional markers, objects, or extra hardware nor do we expect the user to perform any cumbersome and tedious set-up or calibration steps.

Our approach falls in the area of supervised machine learning and regression analysis. The limited range in variations between different human faces allows to offline analyze their appearance under lighting and thereby their reflection properties beforehand and to then apply the findings to new faces. Our light estimation approach thereby is separated into two steps.

In a first step – a one-time offline training process – we learn radiance transfer functions for different positions on the face, based on a dataset of images of faces captured under different *known* illuminations from *The Extended Yale Face Database B* [Geor 01, Lee 05]. The radiance transfer functions describe how incident light on the face coming from different directions is reflected at the particular position on the face towards the camera. These functions enable us in a second step to estimate in real time the real-world lighting conditions from measured reflections in a face.

Mathematically we encode these functions as well as the incident light using a Spherical Harmonics basis (see section 2.4). In this thesis we provide and explain the needed mathematical fundamentals like equations that specify the propagation of light or Spherical Harmonics that we use for modeling

different kinds of functions. We describe the scenery in which we want to estimate the illumination and we derive our particular method that estimates the illumination from a single image of a face.

Additionally we quantitatively evaluate our method by comparing estimated illuminations against ground truth. We provide a way to select sub sets of the sample positions to reduce their number and analyze the correlation between the employed number of sample positions and the achieved accuracy.

We furthermore extend our method to prevent solutions containing non-negligible amounts of negative intensities of light. In addition to that we present a way to make our method more robust against deviations of a particular face from the learned average of faces. We also discuss what amount of information about the illumination is contained in a single image of the face and indicate how to combine the information from multiple images.

We show for a variety of lighting conditions, that by applying the lighting conditions estimated by our method to the virtual content, the augmented scene is shaded coherently in the real and virtual parts of the scene and thus demonstrate that our approach provides plausible results considerably enhancing the visual realism in real-time AR applications.

1.3.3 Scale Estimation from the User's Face

For a realistic preview of objects using AR it is also indispensable that the virtual objects are presented at correct size. As described before the commonly used method of monocular visual SLAM however delivers a camera pose with respect to the scene at an arbitrary scale factor. To determine the absolute scale of the scene and the camera motion, state of the art methods rely either on known objects that have to be added to the scene, on special hardware like depth cameras, or they demand the user to perform longer calibration procedures and movements. All of that represents a barrier to a natural AR experience for a mass-market audience. It assumes that the user either has the particular marker of known size or a specialized hardware at hand or that the user is familiar with non-intuitive extra set-up or calibration steps.

In this thesis, we present a method for estimating scale that requires no additional objects or special hardware. Our developed approach works non-intrusive and allows estimating the absolute scale in handheld monocular SLAM.

While a world-facing camera captures the scene and monocular SLAM maps the scene and estimates the pose of the camera relative to the reconstruction, we detect the user's face in the image of the user-facing camera. Knowing the dimensions of the face in real-world metrics like meters allows to determine the camera position relative to the face at absolute scale.

The dimensions of human faces, e.g. the interpupillary distances [Dodg 04], only vary moderately between different people and are well described by statistics. We thus can employ – comparable to the light estimation approach – the average over all human faces, e.g. the mean value of interpupillary

distances. For an improved accuracy we however also support calibrating the interpupillary distance for a particular user.

With face tracking at absolute scale, two images of the same face taken from two different viewpoints hence enable estimating the translational distance between the two viewpoints in absolute units, such as meters. The motion of the user-facing camera can be transferred to the rigidly connected world-facing camera. Under the assumption that the face itself has not moved relative to the scene between taking the two images, this then allows determining also the motion of the world-facing camera relative to the scene in absolute units, and in consequence also reconstructing and tracking the scene at absolute scale.

In this thesis we will present a proof-of-concept implementation of this idea, which we will quantitatively evaluate against ground truth data with regard to the estimated scale. These evaluations confirm that our approach works and that it provides absolute scale for the reconstruction by monocular SLAM at an accuracy that is sufficient for many AR applications.

We will show for different scenarios how our approach enables reconstruction and tracking at absolute scale. Particularly, we show how our method enables various use cases in handheld AR, from applications that rely on superimposing virtual objects at true size to interactive distance measurements in the environment.

1.4 Contributions of this Thesis

This dissertation thesis proposes to leverage the user's face as a known object in the context of AR in order to deduce knowledge about the real world. The thereby gained knowledge enables a coherent integration of the virtual content into the view of the real world in terms of illumination and scale.

One focus of this thesis lies on pointing out the fact that the face of the user is well-suited to be leveraged as a known object. Firstly the face can be easily captured in common video see-through AR scenarios. Secondly the limited range in variations of human faces allows to rely on average models and pre-processed data. Finally the face of a particular user can also be calibrated once and this calibration can be reused. This calibration is especially reasonable due to two facts. Most of the times a particular user will employ one and the same device, so that the calibration could be stored on the device. Additionally facial recognition allows to recognize a particular user which allows to select the appropriate calibration out of many, that either are stored locally or distributed.

All of this makes the face of the user well-suited to be leveraged as a known object for deducing information about the real world. We hope that this awareness will inspire further research building upon these findings.

Beside the general idea, this thesis makes a series of contributions in the context of video see-through AR. We introduce novel algorithms for determining absolute scale in monocular SLAM as

well as for estimating and reapplying the real-world illumination to virtual objects in real time. A key aspect of the developed methods thereby is the elimination of limitations of current state of the art approaches that either require the user to add additional known objects or markers to the scene or demand specialized hardware like depths sensors. Concept implementations of our methods showcase the effectiveness of the presented algorithms together with evaluations of the quantitative results against ground truth.

1.5 Collaborations

This thesis would not have been possible in its existing form without the outstanding work, assistance and support from my colleagues at Metaio.

Especially the existing code base of the Metaio SDK [Meta 15] has proven to be a valuable tool for many of the prototype implementations, experiments, and evaluations and has been used either as state-of-the-art black box, e.g. for camera calibration, marker tracking, or monocular SLAM, or as starting point for modifications, e.g. by adding Spherical Harmonics rendering on top of the existing rendering pipeline.

In addition to leveraging the existing code base from Metaio, the ideas and implementations presented in here benefited from active discussions with multiple colleagues at Metaio, first of all to be mentioned with Daniel Kurz, who is also the coauthor of all my involved publications. Being my manager over the last three years, he rendered it possible for me to focus my work on topics relevant for this thesis.

This work was also supported in part by BMBF grant ARVIDA under reference number 01IM13001L.

1.6 Publications

All major contributions that are presented in this thesis have either been published in the proceedings of an international conference or are currently planned for submission in form of a journal article. The following gives an overview of the existing publications.

SEBASTIAN B. KNORR AND DANIEL KURZ. "**Real-Time Illumination Estimation from Faces for Coherent Rendering**". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2014. [Knor 14]

This publication has been selected as nominee for the best long paper award of the *IEEE conference ISMAR* 2014. An extended journal version is planned for submission for publication in a special issue of the journal *IEEE Transactions on Visualization and Computer Graphics (TVCG)*.

SEBASTIAN B. KNORR AND DANIEL KURZ. "**Leveraging the User's Face for Absolute Scale Estimation in Handheld Monocular SLAM**". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2016. [Knor 16]

Besides the two peer-reviewed academic publications listed above, two patent applications have been filed in the scope of this work.

1.7 Thesis Overview

This chapter 1 provided an introduction to the thesis, presenting the fundamental idea to analyze images of the user's face in order to overcome existing shortcomings in the reconstruction of the real world for seamless Augmented Reality (AR).

In the next two chapters 2 and 3 we will zoom in on two particular realizations of this general idea. Each of the two approaches and chapters is self-contained and begins with an introduction to the respective topic as well as with a summary of the current state of art. In each case we show how our idea of using the face of the user fits into this context. From that we derive a particular method including a working implementation which we afterwards evaluate in terms of results and performance. We conclude each topic with a discussion about the suitability, potential as well as the shortcomings of our presented approach.

In chapter 2, we will employ the user's face as a light probe for estimating real-world lighting conditions. Estimating the incident light allows us to subsequently use this illumination for matching the lighting of the virtual objects in an AR view.

In chapter 3, we will employ the user's face as an object of known size, which will allow us to bring a monocular SLAM reconstruction of the real-world surroundings performed on images of the back-facing camera from arbitrary scale to absolute scale. By that also virtual objects can be embedded in the augmented view at correct size.

Chapter 4 finally summarizes this thesis with a conclusion of the key findings.

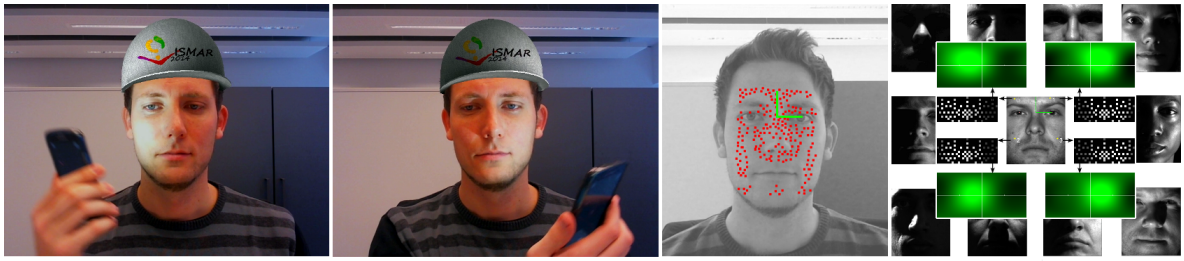
2 Coherent Illumination - The User's Face as a Light Probe

In this chapter we will employ *the user's face as a light probe* for estimating the lighting conditions present in the real world. We learn in an offline process, based on an image dataset with different faces under different known illuminations, how the average face reflects incident light. This knowledge then allows us to estimate in real time the present real-world illumination incident on the user's face from a single image of the face. The estimated incident light is subsequently used for accordingly illuminating virtual objects in an Augmented Reality view. We refer to illuminating the virtual objects according to the illumination of the real world as *coherent illumination*.

We present a method that achieves coherent illumination for virtual objects in Augmented Reality (see figure 2.1) by employing the user's face as a light probe for estimating real-world lighting conditions in real time. This part of the thesis is structured as follows.

We start by giving an introduction to the topics of coherent illumination and illumination estimation in the context of Augmented Reality in section 2.1. First we explain the motivation behind coherent illumination (section 2.1.1) and also indicate existing challenges (section 2.1.2). We then give a brief summary of our particular approach for light estimation in section 2.1.3. This short wrap-up allows us to subsequently compare our method in section 2.2 more easily to other state-of-the-art approaches that acquire the real-world illumination for coherent illumination in Augmented Reality. We will also establish the relation of our particular method to existing work from other domains.

The approach we present in this thesis explicitly focuses on estimating incident light from a single image of a human face captured by a simple monocular camera. We discuss benefits and drawbacks of this specialization on the human face in section 2.3.



(a) (b) (c) (d)

Figure 2.1: Our method enables coherent illumination of virtual augmentations (a, b) by estimating the illumination present in the real world based on pixel intensity values at sample positions in the image of a face (c) in combination with the corresponding radiance transfer functions at the sample positions learned beforehand from a dataset comprising images of faces with known illuminations (d).

After providing mathematical fundamentals relevant for our light estimation approach in section 2.4, we elaborate our particular method in detail in section 2.5. We start by describing the scenery in which we want to estimate the illumination and introduce the involved entities like the camera, the face of the user and the light sources. From that we build up a model for the light interactions and from that derive a simple method that estimates the illumination. Subsequently we identify shortcomings of the simple method and propose improvements to overcome them.

In section 2.6 we evaluate the results of our method. We make a qualitative evaluation by presenting the visual results achieved by our method in terms of the augmented view where the estimated lighting conditions are applied to the virtual content as well as a quantitative evaluation by numerically comparing the estimated illumination to the ground truth illumination.

Finally we complete the light estimation part of this thesis with a discussion in section 2.7 and a conclusion about our presented approach in section 2.8 .

Note on Publication All major contributions of this chapter have either already been published by Knorr and Kurz [Knor 14] in the proceedings of the IEEE *International Symposium on Mixed and Augmented Reality (ISMAR)* 2014 or are currently planned for submission by Knorr and Kurz as an article to the IEEE journal *Transactions on Visualization and Computer Graphics (TVCG)*. In both cases, the fundamental research was conducted by the first author, Sebastian Knorr, under the technical and project-administration guidance of the second author, Daniel Kurz. In particular, the theory of the approach as well as the implementation was developed by the first author.

For the avoidance of doubt The term *coherent illumination* in this thesis does not refer to the physics terminology *coherent light*, which implies a constant phase difference between two light sources. Instead it refers to an illumination of the virtual objects in an augmented view that matches appearance-wise the illumination present in the real world.

2.1 Introduction to Coherent Illumination in Augmented Reality

In this section we will introduce the concept of coherent illumination in the domain of Augmented Reality. We will explain why it is beneficial to strive for coherent illumination (section 2.1.1) as well as what kinds of difficulties exist in achieving it (section 2.1.2). We then will give a short introduction in section 2.1.3 onto how we approach the goal of coherent illumination with our method presented in this thesis.

2.1.1 The Motivation

Augmented Reality combines our natural view of the real world with an overlay of computer-generated content. The view of the real world in *video see-through* Augmented Reality is represented by a live video stream that is captured by a camera. Digital contents, e.g. 3-dimensional virtual objects, are rendered by means of computer graphics and their images are merged with the video stream.

This composite stream then is presented to the user on a display. By looking at the augmented video, the user shall get the impression that the virtual objects are actually placed within the real world.

To generate this illusion, the virtual objects are rendered using a virtual camera pose which conforms to the determined pose of the real video camera with respect to the real world. By that, the perspective of the rendering of the virtual objects is changing along with the perspective of the real world, when the real video camera is moved or rotated within the real world. This already generates the rough impression that the virtual objects are placed within the scene.

For many Augmented Reality applications the goal is to make this illusion as realistic as possible, such that a user who sees the augmented image cannot differentiate between the captured real-world part and the rendered virtual part. Use cases that clearly benefit from realistic augmentations are, for example, virtual try-ons of glasses, jewelry or clothes, as well as place-in-your-augmented-room applications that let you preview e.g. a desired furniture in your home. Here the augmented view shall offer the user a preview of how some object would look in reality. It thus is crucial to mimic how the objects would look in reality as close as possible.

The whole composition of the real and the virtual world must look plausible. It is thus important that the appearance of both the parts, virtual and real, match each other. The perceived realism however is not only induced by the correct position, scale, and orientation of the rendering of a virtual object. Also lighting has a big impact on the appearance of an object and plays a crucial role in how we perceive an object living in the space.

The light incident on an object influences how bright different parts of the object appear. The human vision system uses these differences in brightness, e.g. gradients in shading and reflections, as cues to determine the orientation of surfaces and the overall shape of the object [Rama 88]. Furthermore cast

shadows, which means absence of light, indicate that some other geometry is occluding the light. Cast shadows from one object onto another thereby additionally support our visual system in determining distances and spatial relations between multiple objects [Mama 98].

Simultaneously with determining the shape of the visible objects and their spatial constellation, our mind thereby also builds up a model of the illumination present in the scene e.g. the main directions and intensities of incident light.

For a plausible and convincing augmented image, it consequently is not enough to just render virtual objects photo-realistically with some generic illumination. The lighting on the virtual objects must match the real world. If there is for example strong lighting from the left in the environment, the illumination of a virtual object should match this.

(a) (b) (c) (d)

Figure 2.2: An image of the real world (a) is augmented by a virtual object, e.g. an orange (b, c, d). A coherent illumination (b) of the virtual object is needed so that the augmented image looks plausible. If the illumination of the virtual object is dissonant (c, d), the shading, directions of cast shadows, and positions of highlights are inconsistent with the real world, which makes the augmented image look bogus.

A dissonant illumination of the virtual part and the real-world part manifests for example in inconsistent positions of highlights and cast shadows and disrupts the credibility of the presented augmented view. This fact is illustrated in figure 2.2. A captured photograph showing a real table with a real apple on top (a) is augmented with a virtual orange. A coherent illumination, where the orange is lit coherently to the apple, results in a quite credible and realistic augmented image (b), while a dissonant illumination clearly breaks this illusion (c, d).

2.1.2 Challenges in Augmented Reality

Matching the illumination of virtual objects to the real world poses a challenge in Augmented Reality. The shape, size, and material of a virtual object like, for example, the orange in figure 2.2 usually are known in advance and therefore can be predefined e.g. by a modeler. The lighting, however, if it shall match the real world, heavily depends on the place and time where and when the Augmented Reality application is finally executed. One user for example executes the application in a room where there is a lamp at the ceiling so that more light is incident from above. Another one uses the application in a room, where more light is coming from one side, because there are several windows in the wall. A third user executes the application in an outdoor scenario, such that direct sunlight dominates,

combined with less strong ambient lighting from the sky. Furthermore even for a fixed location, the illumination may change over time. For example when lights are switched on or off, or when the sun is setting or is occluded by clouds.

In order to match the illumination of virtual objects to the real world the question thus remains: What kind of illumination is present in the real world at the time the application is executed?

Currently Augmented Reality applications often simply ignore the issue of coherent illumination and just pick one particular lighting condition for the virtual content. The intent is to choose a nondisruptive illumination. This can for example be a simple uniform illumination, meaning the same light intensity is coming from all directions. Or an illumination from the top is chosen such that the virtual objects feature a cast shadow underneath. While this allows for a quite realistic look of the rendering of the virtual objects in separation, it does not account for the real illumination present in the real world and thereby does not provide a seamless augmentation. Especially when there is a strong dominant light source in the real world, the lack of coherent illumination becomes evident.

We want to do better. Our goal is to enhance the realism of the augmented view by really considering the real-world illumination for the rendering of the virtual objects. The acquisition of the real-world illumination as well as the rendering of the virtual objects with the acquired real-world illumination should be as fast as possible. Firstly, in order not to let the user wait until the illumination is acquired and secondly to support a dynamically changing illumination in real time. In addition, attention should be paid to the fact that the end-user is involved. The method should impose as few challenges and requirements on the user as possible. A tedious set-up or the requirement of particular markers thus should be avoided.

2.1.3 Preview on Our Approach

The method we propose estimates the lighting conditions present in the real world from a single monocular image of a human face in real time. Augmented Reality (AR) applications always involve a user and a lot of hardware setups feature a user-facing camera. In these cases an image of the user's face can be acquired at any time. Based on the visual appearance of this face we estimate the incident light that led to the observed appearance. In an offline process we can learn beforehand with the aid of a dataset containing images of different faces under different illumination how bright certain parts of the average human face appear under certain illumination.

Our approach thereby is particularly beneficial in use cases where virtual objects are augmented directly on the image of the user-facing camera close to the face of the user. Common examples include virtual try-on of glasses, jewelry, or hats, as in the example shown in figure 2.1 (a, b). Different setups for AR experiences comprising a user-facing camera may take advantage of our method, from stationary ones like web-based shopping applications or AR kiosks to mobile ones running on handheld devices such as smart phones and tablet PCs.

The information gathered about the illumination from the user-facing camera however can also be used in other scenarios, e.g. the dual camera set-up of a smart phone with one user-facing camera and one world-facing camera. Knowing the transformation between the two cameras, the illumination estimated on the user-facing camera can be transformed into the coordinate system of the world-facing camera. Limitations and arising problems in this scenario are discussed in section 2.7.

Our method estimates the real-world illumination incident at the location of the user's face in terms of *primary light directions, light colors, and light intensities* encoded as Spherical Harmonics (section 2.4). Running in real time the method is able to *dynamically adapt* to changes in the illumination present in the real world, for example when a light is switched on or off, or when the user is walking through a hallway.

2.2 State of the Art and Related Work

In the following sections we will have a look at existing work regarding acquiring the real-world illumination. We start by naming some fundamental work on merging real and computer-generated images that also consider matching the illumination of the computer-generated part to the real image (see section 2.2.1).

In section 2.2.2 we then give an overview of different existing techniques to acquire the illumination present in the real world. We divide the techniques into two main categories. Firstly, approaches that directly measure the incident illumination by capturing images of the surroundings (see section 2.2.2.1). And secondly, approaches that estimate the incident light at a certain location from the appearance of that location itself (see section 2.2.2.2). Our approach thereby falls into the latter category, as we estimate the incident light from the appearance of the user's face.

After a generic overview on the topic of light estimation we pay additional attention to existing work in the domain of estimating illumination from the image of a human face in section 2.2.3.

2.2.1 Fundamental Work in Merging Real and Computer-Generated Images Considering the Real-World Illumination

In 1986, Nakamae *et al.* [Naka 86] present one of the pioneer works about combining virtual renderings with images of the real world where also the illumination of the computer-generated content is semi-automatically matched to the illumination visible in the image of the real world. In their paper, they demonstrate a method for architectural simulation, where a 3-dimensional computer-generated image of buildings is superimposed on top of a digitized photograph of an outdoor environment. The lighting of the virtual buildings is adapted to the real world by using information from the real background picture. Nakamae *et al.* use an illumination model consisting of two components, the sun and the sky. They calculate the position of the sun based on the capture time and date, the longitude and latitude coordinates of the camera position as well as the viewing direction of the camera. Once the sun position is determined, the unknown parameter is the *ratio between the intensity of the sun and the sky light*, which depends on the present illumination situation. They determine this ratio by comparing pixel intensities and surface normals of two manually selected walls in the real image, where one wall is facing the direct sun light while the other one is in shadow. If two such walls are not contained in the image, they propose to position an additional white pilot box in the scene as a light probe. Other approaches that also use information from the shading of real objects to deduce the light situation are covered in section 2.2.2.2.

In 1993, Fournier *et al.* [Four 93] simulate indoor global illumination for a combined scene of a real video image and a computer-generated image. The simulation is based on radiosity computations [Cohe 93], which approximate the solution of the rendering equation using *finite elements* (surface triangles), and calculate how much one element directly influences another one in terms of illumination.

In their publication the real scene geometry is approximated by a virtual model built out of simple geometry like boxes. For the manual creation of that model, photographs with orthographic views of the objects are taken and used to manually define textures with transparency for parts not covered by the object. The material of the real scene is assumed to be diffuse and reflectance values are estimated based on average intensities in comparison to neighbouring image regions.

The authors demonstrate that when the real light positions are already known but the intensities of the single light sources are unknown, the relative intensities of the light sources can be estimated from radiosity solutions on the reconstructed real scene. Therefore they compute separate radiosity solutions, one for each light source using unit intensity. The final radiosity value of an element is the sum over the scaled solutions from the single lights. The intensity values for the different lights thus can be optimized so that the final computed radiosity solution with all the lights best fits to the "measured" radiosity values of the different finite basis elements.

Fournier *et al.* do not only estimate the illumination, but also take into consideration how an added virtual object changes the illumination in the real world. With the known positions and estimated intensities of all light sources, a second radiosity solution is calculated now containing the reconstructed real scene *and* the virtual objects. The ratio between the old and new radiance per element is used to modify the intensity of each pixel in the real video image belonging to the element. Thereby effects like cast shadows from a virtual objects onto a real surface as well as light scattered from virtual objects to the real surface are covered.

If light positions are also assumed to be unknown, every element of the reconstructed scene may be considered a potential light source and the average image intensity per element from the real-world image is used as initial surface radiosity. To calculate the influence of the virtual object on the real environment, "negative" radiosity is emitted between the elements and subtracted, if the path is blocked by the virtual content.

Fournier *et al.* [Four 93] note that the rough quantization in the global illumination computation leads to artifacts and does not allow for modeling finely localized shadow. Still it shows, that a rough approximation of the real scene already allows for satisfactory results concerning a matching illumination including mutual influence between the real and virtual objects.

In 1996, Hirota *et al.* [Hiro 96] present a hybrid tracking method combining vision-based tracking and magnetic tracking. In one of their sample applications they put a mirror sphere into the view of the video camera and grab the image of the sphere, which shows the reflections of the real environment and thereby the light incident at the location of the mirror sphere. They use the image as a reflection map for a virtual teapot. They also let the virtual objects cast shadows onto the real scene, which has been acquired beforehand.

In 1998, also Debevec [Debe 98] places a mirror sphere as light probe in the scene at the target position of the virtual content for capturing high-dynamic-range panoramic measurements of the scene radiance. His goal is to improve the mutual interactions of light between real and virtual objects in

a composite scene. Similar to Fournier *et al.* [Four 93], Debevec calculates two global illumination solutions, a first image with only a proxy model of the real scene and a second image with the proxy model together with the additional virtual content. Debevec however then employs the *difference* between the two solutions to calculate the differential effect that the added virtual objects have on the appearance of the real scene and vice versa. He uses this differential effect to correct the final augmented image. This incremental update is commonly known as *Differential Rendering*. The scene is partitioned into three components: the (real) *distant scene*, the (real) *local scene*, and the *synthetic objects*. Similar to our approach, Debevec uses the assumption that the distant scene only emits light and is not influenced by the addition of synthetic objects. The local scene and the synthetic objects in contrast influence each other in terms of light interactions and thus need to be modeled including geometry and material. By only estimating the change in illumination, flaws in the reconstruction of geometry and material of the real local scene parts have a much smaller impact.

2.2.2 Related Work for Determining the Real-World Illumination

Information about the illumination that is present in a real scene can be gained in multiple ways. Often existing work and also the method we propose reduces the problem by assuming that the illumination only needs to be estimated incident *at a certain location* in the scene. Preferably at the location where the virtual objects will be added.

We will distinguish two fundamental ideas:

The most straight forward way to determine the light incident from the environment at a certain location is to directly measure it by capturing an image of the environment from that location. While our method itself does not directly capture the surrounding environment to determine the incident light, this is a very popular approach. We will enumerate some work following this principle in section 2.2.2.1.

An alternative to taking direct measurements of the surroundings is to deduce the incident light from the appearance of the lit scene. Cues like highlights, shading, or cast shadows visible in an image of the real world can be used to reconstruct the incident light. Applying various assumptions and restrictions, the most plausible explanation for the observed effects can be found. The method we propose falls into this category, as we deduce the incident light from the appearance of the user's face. We summarize various approaches that reconstruct the illumination from the appearance of the lit scene in section 2.2.2.2.

2.2.2.1 Directly Measuring the Incident Light

A common approach for directly measuring the incident light at a location is the acquisition of an omni-directional image which shows the real-world surroundings from that specific location. Such an image is also referred to as *environment map*. Each pixel of the image corresponds to a certain

direction of incident light. The intensity of a pixel describes the intensity of light incident out of that particular direction. Due to non-linear effects in the capturing pipeline, e.g. the camera response curve, also the mapping from light intensities to pixel intensities often is non-linear. A captured environment map can either be directly used for simulating the incident illumination in the rendering process or it can be further processed by, for example, extracting the main sources of illumination. Different approaches exist to capture this kind of image.

Mirror Sphere One already mentioned approach to measure incident illumination is to capture an image of a mirror sphere [Hiro 96, Debe 98, Gibs 00, Gibs 03, Supa 06, Pess 10], that is positioned within the scene where the synthetic objects shall be placed at afterwards. The chrome sphere reflects the illumination incident on the sphere from the surrounding scene towards the camera so that again each pixel corresponds to a particular direction of incident light. As described in [Rein 10], special care may be taken for the attenuation of light by the mirror itself by calibrating the reflectivity of the sphere, as well as for blind spots in the captured directions by combining two images of the sphere captured under two different orientations.

A drawback of this method is the need for a mirror sphere and the additional set-up step, as the mirror sphere has to be actively added to the scene. By that also the original scene is altered. The method, however, supports dynamic scenes and changes in illumination by continuously capturing images of the mirror sphere. Also the resulting environment maps capture the illumination very well at a high resolution.

By taking multiple low-dynamic-range images of the sphere with different exposure, Debevec [Debe 97, Debe 98] reconstruct a *high-dynamic-range* image capturing the full dynamic range of the scene illumination. The author points out, that in order to produce photo-realistic lighting on the objects, high-dynamic-range measurements of scene radiance are necessary.

The use of mirror spheres to capture the incident light at a specific spot nowadays is still state of the art in many movie productions. In this professional domain the additional time and effort is clearly outweighed by the exceptional quality delivered by this method. In case that lighting is dynamic, multiple exposures that are needed to reconstruct the high dynamic range however are impractical. Recently LeGendre *et al.* [LeGe 16] presented a set-up with two mirror spheres for multispectral capturing of incident light. Additionally to a first chrome sphere, the set-up comprises – besides color checker charts and markers – a second black acrylic sphere. This second black specular sphere increases the dynamic range captured by a single shot of the set-up. It allows to take a single image at an exposure level that captures both the light from the environment in the first chrome sphere as well as the bright light sources in the black sphere. Both reflections then can be combined into a measurement with high dynamic range.

Fish-eye Lens Another way to create an environment map is used by Sato *et al.* [Sato 99]. They employ a camera setup with a fish-eye lens to capture hemispherical images of the surroundings. This so called *inside-out method* is used frequently [Frah 05, Supa 06, Gros 07, Knec 10, Knec 12, Kan 12, Son 12, Kan 13, Niko 13, Fran 13] for determining the real-world illumination in AR.

The camera capturing the illumination is additional to the camera which is responsible for the augmentation part. This additional requirement for an extra camera with a fish-eye lens as well as the extra capture procedure are disadvantages of the method. For non high-dynamic-range cameras, a wider dynamic range can again be recovered using multiple images following [Debe 97].

Combining Multiple Images Captured with Narrow Field of View In general, these kinds of panoramic, hemispherical, or omni-directional images can also be created without a mirror sphere or fish-eye lens by stitching together multiple images captured with narrow fields of view (FOV). Problems arise in practice because the manual process of capturing all directions is quite tedious and time consuming without proper equipment like a tripod (at the least). A varying focal point between the different images results in parallax errors, which are especially bad for closer surroundings. Additionally the captured frames need to be stitched, and thus need to have enough overlap and distinctive features.

DiVerdi *et al.* [DiVe 08] present an approach that constructs environment maps for Mixed Reality. They apply vision based tracking on video streams to estimate the camera pose and optionally support the pose estimation using the gryoscope / compass. Frame by frame they project the video image into a cubemap. Additionally they provide feedback to the user about where there are still gaps in the environment map. The finally remaining gaps are filled using texture diffusion.

To simplify the tedious capture process, Jung *et al.* [Jung 13] do not capture and reconstruct the full environment map but only take images of the surrounding main light sources. They rely on a smart phone offering two cameras - one facing front and one facing back. They capture pictures of the light sources at the ceiling using the front-side camera while the back-camera and orientation sensor is used for tracking. The captured images are mapped to the hemisphere and light sources in the image are detected by binarizing and contour detection. This gives them intensity, color information, as well as the dimension of the light sources.

A general disadvantage of the approaches that combine multiple images are the separate manual capture process and the lack of support for dynamic scenes.

Processing the Environment Map A hemispherical or omni-directional image - whether created by a mirror sphere or fish-eye lens - contains pixel-wise measurements of incident light at a high angular resolution. For a real-time rendering process a fast approximation is needed. Therefore the acquired environment map image is often further processed.

For the shading of the virtual objects, pre-filtered versions of the environment map image can be generated [Kaut 00a, Kaut 00b, Gibs 00, Supa 06, Pess 10], which range from fully diffuse irradiance maps over glossy ones to fully specular ones. The shading of a virtual object then is performed by simply sampling values from these maps. The specularity of the surface determines from which maps to sample from. The surface normal of the virtual object determines the pixel position within the map.

Sometimes simpler light representations like the main light directions are preferred for the rendering process especially for computing cast shadows. Different approaches exist for extracting light sources from an environment map image.

Gibson and Murta [Gibs 00] for example determine up to 8 directional light sources in an offline pre-processing step. They therefore first use the environment map to render a target reference image showing the shadow cast by a simple virtual sphere onto a plane using Monte-Carlo ray-tracing. Monte-Carlo ray-tracing uses Monte-Carlo integration [Hamm 64], a method for numerical integration using random numbers, for calculating integrals over light paths. Using the rendered target reference image, Gibson and Murta [Gibs 00] then use an optimization procedure to find the set of up to 8 directional light sources that create a cast shadow which fits best to the shadow visible in the target reference image.

Supan *et al.* [Supa 06] down sample the cube map of the environment map to a low resolution (i.e. to 4x4 pixels per face). For each of the remaining pixels, they create one light source according to the position, intensity, and color of the pixel.

A more common approach to extract light sources from an environment map is applying pure image processing techniques. The main directional light sources for example can be found by generating samples on the environment map with probabilities according to intensity [Gros 07], by warping samples initially uniformly distributed over the hemisphere hierarchically based on relative luminance within the mipmap levels [Knec 12], or by adaptively subdividing regions and minimizing the variances in the generated regions [Fran 13]. Alternatively, the main directional light sources can be found by determining the center of gravity of image parts that are saturated in all channels [Frah 05], or by applying thresholding on the image and detecting blobs on the resulting binary image using connected component analysis with contour tracing [Kan 12, Kan 13].

Grosch *et al.* [Gros 07] also present a more complex approach for simulating the indirect illumination caused by direct incident illumination. For distinct regions of a hemispherical camera image, which they use to capture the direct incident illumination, they precompute the resulting indirect illumination. They compute so-called *basis irradiance volumes* using a radiosity simulation on the manually created 3-dimensional model of the scene. This allows them to then represent an arbitrary daylight situation by the linear combination of the distinct regions. The indirect illumination resulting from the daylight situation then can be calculated as a linear combination of the basis irradiance volumes of the distinct regions.

Relaxing the Distant Scene Assumption The environment map captured with a mirror sphere or fish-eye camera is only valid for that particular location where the probe was taken, i.e. where the mirror sphere or fish-eye camera was placed. A common assumption however is, that as long as the surroundings are sufficiently far away, the environment map is still sufficiently correct in the close neighborhood of that location.

This also assumes, that the incident illumination does not vary with location. In order to overcome this limitation, some methods try to recover the 3-dimensional location of the light sources. The acquired radiance information from the environment can for example be projected onto an approximate geometrical model of the scene [Debe 98, Gibs 03].

Sato *et al.* [Sato 99] take a pair of omni-directional HDR images at two different locations using fish-eye lenses and apply an omni-directional stereo algorithm to reconstruct a geometric model of the scene from the two images. Feature points with high contrast in the images are considered as direct light sources and their 3-dimensional coordinates are determined by the intersections of the 3-dimensional reprojection lines.

Frahm *et al.* [Frah 05] capture a sequence of hemispherical images with a moving fish-eye camera. They recover the 3-dimensional position of the light sources detected in the images by tracking the camera motion and triangulating the light source directions between the frames.

Pessoa *et al.* [Pess 10] use an omni-directional HDR image of the environment as a simple skybox. The image has been pre-captured using a chrome sphere. For nearby real objects they employ pre-modeled geometrical representations. From the center of each virtual object they then create a separate environment map for that object, which shows the surroundings including the other virtual objects, the pre-modeled representations of the nearby real objects, as well as the skybox in the background. For each of these environment maps they also create a diffuse and two glossy versions for lighting the virtual object.

Nikodým *et al.* [Niko 13] investigate the use of multiple live video environment maps, which are captured simultaneously by smartphone cameras with fish-eye lenses at different locations within the scene. They do not reconstruct the real 3-dimensional positions of light sources, but generate directional light sources by merging the data from the multiple displaced cameras depending on the position of the virtual object.

Meilland *et al.* [Meil 13] illuminate virtual objects by a 3-dimensional map of the scene created by dense visual SLAM with a low-dynamic-range RGB-D camera. The 3-dimensional map of the scene is extended over time by high-dynamic-range values recovered from low-dynamic-range images taken with different exposure times. The image-based high-dynamic-range model of the scene allows generating dense virtual images for a specific camera pose by blending nearby key-frames. By that also virtual high-dynamic-range environment maps can be generated at the position of virtual objects. The scene and illumination in this approach however is restricted to be static.

2.2.2.2 Estimating the Incident Light

Directly measuring the incident light at a location by acquiring images of the surrounding environment as described above comes with the burden of additional required equipment such as a mirror sphere or fish-eye lens. It also involves an extra set-up step, e.g. putting the mirror sphere in the scene or acquiring the fish-eye lens image. For professional installations this poses no big problem. The benefit is a reliable and robust acquisition of the incident illumination at a high resolution.

For Augmented Reality applications that are used by non-experts, e.g. in the field of handheld Augmented Reality or webcam-based virtual try-on, these methods however are too cumbersome and thus not feasible. A less disruptive way to acquire the illumination, which preferably happens under the hood, is favored in these cases.

An alternative approach for acquiring the incident illumination is to *estimate* the lighting conditions from the appearance of the illuminated parts of the scene in the view of the camera. The method we propose in this thesis also falls into this category, as we deduce the incident light from the appearance of the user's face. Instead of *measuring* the incident light at a specific location by directly capturing the real-world surroundings with an omni-directional image taken at that location, information about the incident light is derived from visible effects like highlights, shading or shadows cast. Applying various assumptions and restrictions, the most plausible illumination for the observed effects is found.

Surface Reflectance Properties What is captured as pixel intensities by an image of the scene is loosely speaking the intensity of light from the scene that is reflected towards the camera. Amongst others the amount of reflected light also depends on the reflectance properties of the scene at that surface position, for example whether the surface is smooth or rough. We also refer to these properties as material. Approaches that estimate the lighting conditions from the appearance of the illuminated parts of the scene often make assumptions about the reflectance properties of the scene. While reflectance properties can be quite complex in reality, they are commonly approximated in computer graphics and computer vision using simpler basic reflectance types.

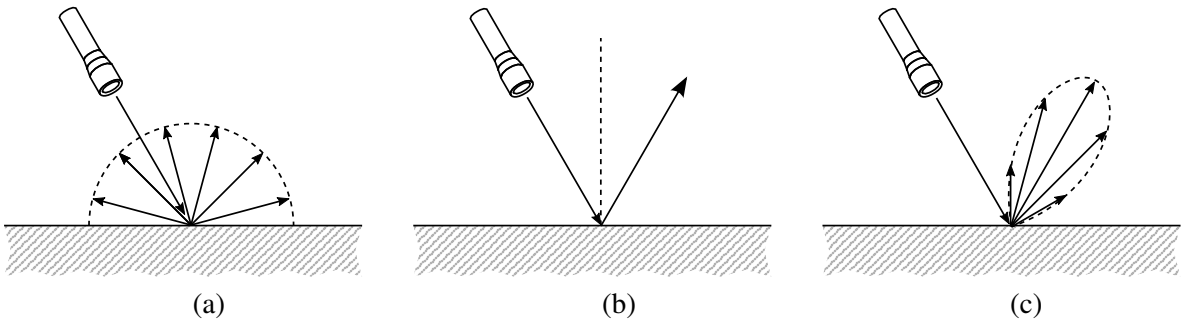


Figure 2.3: Basic reflectance types comprise diffuse (a), specular (b), and glossy (c) reflectance.

Often surfaces are assumed to be so-called *Lambertian surfaces*, which means that their reflectance property is purely *diffuse*. A diffuse surface reflects incident light equally into all directions like depicted in figure 2.3 (a). In consequence a diffuse surface looks equally bright from all viewing directions. The diffuse reflectance property for a specific wavelength of light can be specified by a single value – the reflection coefficient – also known as *albedo*. How bright the surface appears under illumination then depends on the albedo as well as the irradiance incident at the surface position, which is the effective power of incident light. This power of incident light follows the Lambert's cosine law [Lamb 92], which states that the irradiance is directly proportional to the cosine of the angle between the surface normal and the direction where the light is incident from.

On the other side the appearance of a surface with *specular reflectance*, which we know for example from mirrors, heavily depends on the viewing direction. Light incident on a specular surface is not equally spread into all directions. Instead the light incident out of one direction is also reflected in one direction only. The angle of incidence thereby equals the angle of reflectance as depicted in figure 2.3 (b).

Figure 2.3 (c) also shows a third type of reflectance, referred to as *glossy reflectance*. Similar to the specular reflectance here again the main reflection direction depends on the incident light direction. However, light in this case is also spread in a range of directions around to the main reflection direction. The appearance of surfaces with glossy reflectance thus also depends on the viewing direction. A material in computer graphics often is modeled by a linear combination of diffuse, glossy, and specular reflection.

Inverse Lighting Deriving information about the incident light from visible effects like highlights, shading, or shadows cast in an image is also known as *Inverse Lighting* and was introduced by Marschner and Greenberg [Mars 97]. They reconstruct lighting from a photograph and a 3-dimensional model of the pictured object. In their approach incident light is modeled by uniformly distributed directional basis lights. The incident light is determined by the linear combination of the corresponding basis images (generated using the 3-dimensional model) that best matches the photograph. Marschner and Greenberg demonstrate re-lighting, i.e. modifying the image according to a new user-specified lighting configuration, for a diffuse rigid object as well as for a human face. In contrast to their approach, we do not rely on an explicit 3-dimensional geometry model of a face but learn our model from captured images of human faces. Furthermore, we have a sparse sampling approach while Marschner and Greenberg use a dense projection of the geometry. Our approach is also different in the employed illumination model and the method of resolution.

Note that there is a continuous spectrum between directly measuring the incident illumination by capturing the surrounding environment and estimating the incident illumination from the appearance of unknown arbitrary objects in the local scene. In some way, taking an image of a mirror sphere in order to capture the surroundings by the reflections can already be seen as some kind of Inverse

Lighting as we deduce the incident light from the appearance of the mirror sphere. The specular reflectivity and simple shape of the sphere just make the problem of Inverse Lighting trivial to solve.

Knowledge about the objects that are visible in the camera image in terms of geometry and material, i.e. reflectance properties, is needed for Inverse Lighting in order to derive the illumination reliably. A theoretical framework for the general problem of Inverse Rendering, that estimates multiple kinds of rendering attributes like lighting and reflectance properties of objects from images, is introduced by Ramamoorthi and Hanrahan [Rama 01]. They analyze the mathematical foundation of the reflected light field. By employing Spherical Harmonics representations (section 2.4), they demonstrate that for a curved, convex, and homogeneous surface under distant illumination the reflected light field can be described as a convolution of lighting and reflectance properties. Inverse Rendering accordingly can be seen as deconvolution. Ramamoorthi and Hanrahan also explain why light estimation from diffuse surfaces is ill-conditioned in contrast to light estimation from mirror-like surfaces. In a nutshell, the illumination that can be recovered will be a filtered version of the real illumination. High frequencies in the illumination can only be reconstructed from shading, if also the surface reflectance properties contain high-frequency components, e.g. sharp specularities. Similar insights are presented by Basri and Jacobs [Basr 03]. They show, that the set of reflectance functions for diffuse objects lies close to a 9-dimensional subspace. Images of a diffuse (convex) object under variable lighting thus can be represented using only nine basis functions. We also use Spherical Harmonics in our approach and make use of some of the insights from Ramamoorthi and Hanrahan [Rama 01]. We however have an object, i.e. the face, that is neither fully convex, nor homogeneous and that exhibits cast shadows, multiple reflections and subsurface scattering of the light.

Also in the context of Augmented Reality (AR) multiple methods exist that apply Inverse Lighting to recover information about the illumination. These methods often focus on a specific lighting cue like cast shadows, specular or diffuse reflection and rely on *known objects with predefined geometry and reflectance properties* that have to be placed additionally into the scene.

Specular Reflection Kanbara and Yokoya [Kanb 04] focus on specular reflections to estimate the incident illumination by Inverse Lighting. They attach a small black mirror ball to a conventional 2-dimensional square marker. The 2-dimensional marker is used for camera tracking. The reflections of the 8 brightest spots on the black ball are used to estimate directions, colors and intensities of the light sources. This approach is very close to capturing an environment map using a mirror sphere (section 2.2.2.1). The black color of the sphere however resolves the dynamic range problem and simplifies the detection of the 8 brightest spots. On the other side the small size of the attached black sphere ball results in a small pixel footprint in the captured image. This may cause discretization artifacts such as discontinuities in shadows calculated from the extracted light sources.

The black mirror ball from Kanbara and Yokoya [Kanb 04] is intentionally designed to feature only a specular reflection. Ordinary objects however often feature a combination of diffuse and specular reflection. Klinker *et al.* [Klin 88] demonstrate that for many dielectric materials, like porcelain,

paper, or plastic, the reflected light can be described as a *linear* combination of an object color and a highlight color. They further show that by analyzing color pixels and their corresponding clusters on a dichromatic plane in the color space, a single color image can be separated into two intrinsic reflection images: one image showing the highlight reflection and the other image showing the matte diffuse object reflection. The highlights then can for example be used to determine the color of the illumination.

The separation of a light field into a diffuse and a specular component is used by Jachnik *et al.* [Jach 12] who conduct inverse lighting from a planar surface which exhibits amongst others a specular reflection component. In contrast to Klinker *et al.* [Klin 88] they however rely on a set of multiple images and separate the light field into the diffuse and specular component by comparing multiple measurements for a single surface point. For that they track the surface and take observations of the surface from different angles. From these observations they create a surface light field, which is a 4-dimensional function specifying the observed color depending on the location of the surface and the viewing angle. In order to separate this light field into a diffuse and a specular component they first set the diffuse component for a surface location to the median color of all the observations of that location from different angles. The specular component then is the remaining part after removing the diffuse component. By applying the distant illumination assumption, i.e. assuming that incoming light is not varying with location but only with direction, the remaining specular component is collapsed into a 2-dimensional function, corresponding to an environment map.

Uranishi *et al.* [Uran 16] propose some kind of exotic marker for estimating the direction of incident light, which they name *The Rainbow Marker*. Their marker is composed of a conventional planar marker for determining the camera pose with respect to the marker and an additional microscopically structured surface. This surface produces structural color, i.e. the appearance of the surface varies depending on viewpoint and direction of incident light. In a pre-process the authors collect referential patterns by observing the surface under different known angles while it is lit from different known directions. Afterwards the authors estimate the incident direction by finding the referential pattern that best matches the present appearance of the marker for the current camera pose.

Diffuse Reflection As already described above, diffuse surfaces reflect light equally in all directions. The brightness of a surface depends on the orientation of the surface towards the incident light, according to Lambert's cosine law [Lamb 92]. Aittala [Aitt 10] assumes a fully diffuse reflection for the surfaces he uses, which are either a ping pong ball or a planar marker that is rotated in front of the camera. Both objects are easy to track and have a known diffuse reflectance. The ping pong ball exhibits different values of brightness for different surface orientations according to Lambert's cosine law. The planar marker changes his brightness accordingly when it is rotated. For a set of linearly dependent basis lights, Aittala estimates the light intensities by minimizing the differences between the image brightness observed for a surface orientation and the brightness resulting from the diffuse reflection of the linear combined basis lights. Additionally he employs regularization and thereby

favors sparse solutions, i.e. solutions where many coefficients for the basis lights are equal to zero.

Also Calian *et al.* [Cali 13] assume fully diffuse surfaces. Instead of first recovering the incident illumination and afterwards using the recovered illumination to light the virtual objects accordingly, Calian *et al.* [Cali 13] however use so-called shading probes. These 3D-printed objects consist of a white diffuse kernel that is partitioned into different spherical sections by black walls. The shading of the 3D-printed objects thus is parameterized by surface orientation and visibility of the hemisphere. During rendering they directly use these captured shading values for shading the virtual objects. In order to capture all the different kernel parts of the shading probe the user must however rotate the camera around the probe.

Shadows Specular reflection and diffuse reflection depend on the orientation of a surface towards the incident light. But also the absence of incident light at a specific location can be used as a cue. If a surface point lies in shadow with respect to a light source, the direct connection line between the surface point and the light source is occluded by other scene geometry. Shadows can thus be used together with knowledge about the scene geometry to infer information about the position of a light source.

Arief *et al.* [Arie 12] estimate the direction of one dominant light source based on the shadow contour cast by a cuboid shaped *3-dimensional AR marker*. They simultaneously use this marker for tracking. The fix simple geometry of the cuboid marker and the assumption of a planar surface beneath the marker make the shadow contour analysis faster. Approaches that rely on detecting shadow contours however require hard shadow borders to simplify the detection.

Some methods that consider shadows to derive information about the illumination assume that the position or direction of the light source is already known. They then try to recover either only the intensity and color of the primary light source or they even only investigate the visual *effect* of a shadow cast by an object in order to reproduce this effect for a consistent shadow cast by virtual objects.

Jacobs *et al.* [Jaco 05] present a real-time algorithm which allows for color-consistent virtual shadow considering a single light source. The geometry of the visible real objects and the position of the light source must be approximately known. Based on the geometry and the position of the light source they determine the shadow regions and refine them with an edge detection procedure. The average color *within* the shadow region divided by the average color *nearby* but outside the shadow region gives them a *scaling factor* between the shadowed region and the lit region. Virtual shadows then can be applied to a region that in reality is directly lit by scaling the colors in that region appropriately. For different materials and orientations towards the light source different scaling factors would need to be used, and thereby different shadow regions would need to be investigated.

RGB-D Cameras for Arbitrary Geometry All the above mentioned approaches of inverse lighting assume that the geometry of the captured object which is used for estimating the illumination is already known. Recently, approaches employing an RGB-D camera like the Microsoft Kinect have become more and more popular in the field of light estimation for Augmented Reality. Along with the usual RGB color image, these RGB-D cameras deliver an additional depth image containing a per pixel distance measurement. This means, these cameras provide by themselves already a partial reconstruction of the scene geometry. Combined with an algorithm like Kinect Fusion [Izad 11], which fuses information from multiple depth images taken at different view points, the complete scene can be reconstructed geometry-wise. This scene reconstruction is a clear benefit, as it eliminates the need of relying on predefined known objects.

Gruber *et al.* [Grub 12] present an approach of Inverse Lighting for Augmented Reality that supports *arbitrary* scene geometry by using an RGB-D camera for simultaneous geometry reconstruction and camera pose estimation. They do not recover the color of the light but assume a white light color and a diffuse reflectance model for the entire scene. Similar to our approach they make use of Radiance Transfer Functions (RTFs). While we capture the RTFs in an offline process based on images with known illumination and in doing so also take non diffuse reflections, subsurface scattering, and indirect illumination into consideration, Gruber *et al.* calculate the RTFs for visible surface points at run-time purely based on the occlusions by nearby scene geometry using raycasting on a voxel representation of the scene. Similar to us they encode both the illumination as well as RTFs using a low order Spherical Harmonics basis (see section 2.4). The illumination is specified by direction only, meaning it does not depend on the location. Each visible surface point contributes an equation between the observed pixel intensity, the RTF, and the unknown illumination. The unknown illumination is estimated by solving the system of linear equations of multiple observations by least squares minimization. Calculating the RTFs creates a performance bottleneck, still Gruber *et al.* [Grub 12] achieve interactive rates for the light estimation. Our light estimation method employs pre-learned RTFs for a sparse set of sample positions, which significantly speeds up the process.

Boom *et al.* [Boom 13] also use an RGB-D camera for scene reconstruction and assume a diffuse reflectance model. However, instead of only estimating the directional light distribution, they also estimate the 3-dimensional position of one single point light source. For a captured image they build regions of constant albedo based on the assumption that contiguous segments in the RGB image with similar color have the same albedo. The surface normals for a pixel can be extracted from the reconstructed scene. They then run an optimization procedure regarding the position of the light source. In each step they use the current estimate for the position and intensity of the light source to compute the albedo for each segment considering the surface normals and the observed brightness. Afterwards they compute a reconstructed image based on the light position and albedo terms. Iteratively they search for the light source position and intensity that minimize the error between the original image and the reconstructed image. This search does not run in real time.

Similar to the work of Boom *et al.* is an approach presented by Neverova *et al.* [Neve 12], which

uses specular and diffuse reflectance in combination to estimate the illumination from a single image. Exploiting the surface normals delivered by the depth camera, the image is separated in an iterative process into a specular image and a diffuse (i.e. specular-free) image. Simultaneously the light color is estimated from the specular highlights. Initial light positions are estimated from the directions of the specular highlights via the intersections from different reflections. Afterwards an optimization framework refines the estimates for the light positions by minimizing differences between the original diffuse and specular shading and the rendered solution. The authors show that accounting for the specular reflection part improves the results. Yet their approach does not run in real time.

When considering specular reflection, multiple images of the scene captured from different viewpoints can be helpful, as the specular reflection in contrast to the diffuse reflection depends on the viewing direction.

Buteao and Saito [Bute 15] also make use of an RGB-D camera for light estimation. They are able to estimate the position of *multiple* point lights. They run a plane separation algorithm ontop of the 3-dimensional reconstruction of the scene from Kinect Fusion [Izad 11]. Using multiple images of the scene with different camera poses, they separate the brightness of the walls into diffuse and specular components. They identify the brightest spots in the diffuse component of a wall (which they assume is uniformly colored), and claim that the location of the illuminating point light is located along the ray starting at this spot in direction of the normal of the wall. Afterwards they use the specular components to refine the position of the light sources.

Richter-Trummer *et al.* [Rich 16] use an RGB-D camera to estimate the incident lighting on objects as well as to recover their surface materials. By that they both support relighting of the scanned objects as well as lighting virtual objects coherently to the real world. Their approach requires multiple images of the objects that are captured from different viewpoints. From these images the geometry of the objects as well as their lit texture is reconstructed. The lit texture is generated from the average color of a surface point from multiple observations. Based on this lit texture the mesh is segmented into parts sharing the same material. Although multiple separate materials for the objects are supported, each material needs to span a larger connected part. The radiance transfer functions (RTFs) for the reconstructed geometry are computed in terms of Spherical Harmonics (see section 2.4) and each material is initialized with a grey albedo. Starting from this, they estimate the incident lighting (including light colors). They then calculate the error per material patch based on the estimated lighting and adequately adjust the albedo of that segment. They alternately estimate the illumination and adjust the albedos. By that they are able to recover the diffuse albedo color for each material as well as the lighting including light color. Additionally they then recover also the specularity of the materials based on the original observations of the surface, the reconstructed geometry, and the estimated illumination. While they demonstrate that they are able to separate light color and material color, they thereby rely on a scene with objects of different large uniformly colored regions.

All the mentioned approaches that work with an RGB-D camera levitate the need to have a known

object available. They show good results in estimating the illumination. On the other side they require the user to have an RGB-D camera available and involve a quite high computational complexity. Additionally the approaches are unable to reliably solve the ambiguity between light and material for arbitrary scenes.

An approach that similar to ours also focuses on a particular body part of the user is presented by Yao *et al.* [Yao 13]. They use an RGB-D camera to capture the hand of the user as a shading probe. While we however learn radiance transfer functions beforehand from a dataset of images, they calculate the shading in terms of Spherical Harmonics based on the surface normal derived from the depth images. In their process they neither consider occlusions nor non-diffuse reflections.

Intrinsic Images Aside from the above mentioned approaches that primarily target at estimating the illumination in real time for a coherent illumination of virtual objects, a lot of research on the topic of light estimation and simultaneous scene reconstruction from images has been done in the name of recovering so called *intrinsic images* [Barr 78]. These approaches – e.g. the one by Barron and Malik [Barr 13] – try to separate an input image into multiple parts like illumination, object geometry, and object materials by defining additional constraints for the particular components. Often these approaches employ non real-time optimization procedures. Recently, Meka *et al.* [Meka 16] however presented a method which performs the intrinsic decomposition for a video sequence in real time. Transferring models, restrictions, and findings from the domain of intrinsic images to the topic of light estimation in AR seems promising.

Outdoor Scenarios with Sky and Sun Model A variety of methods for light estimation exist which explicitly focus on the outdoor scenario. These methods try to determine the ratio between sunlight and skylight from captured images based on regions in direct sun light and regions in the shade.

Madsen and Nielsen [Mads 08] present a method to estimate the radiance values for the sky and the sun in outdoor scenarios with changing illumination conditions. Their outdoor daylight illumination model consists of a sky dome covering the entire hemisphere and a distant disk light source representing the sun. The position of the sun is determined using the position on earth together with compass, date and time. Shadows are detected based on pixel statistics in the chromaticity plane combined with a graph cut algorithm that estimates RGB values which can be blended on pixels in direct light to mimic the effect of shadow. Thereby pixels are rated on a range from being directly lit by the sun to being fully in shadow. This shadow detection method is based only on information from cast shadows present in the image without any knowledge about the shadow casting geometry. For a diffuse surface the reflected radiance is proportional to the irradiance. Appling the daylight illumination model, this irradiance consists of two parts, irradiance from the sky and from the sun. The ratio of pixel values between shadow and non shadow is independent of camera constants and surface albedo. It only depends on the shadow to sun irradiance ratio. From the ratio between the irradiance within areas in

shadow and the irradiance within direct sunlit areas the unknown radiance values of sky and sun can be determined.

In [Mads 10] Madsen and Lal continue working on radiance recovery from cast shadow in outdoor scenarios under daylight conditions. This time an image stream is acquired using a stereo camera, additionally delivering dense 3-dimensional information. The camera position and orientation is fixed. The shadow cast by moving objects is detected as those pixels where the color is changing without a change in depth. Additionally these shadow candidates are verified using the fact, that sky light contains a larger portion of blue light than direct sun light. Using per pixel information like the surface normal, ambient occlusion value, and color information from verified shadow pixels they estimate the unknown sky and sun irradiance values.

Liu *et al.* [Liu 09] focus on AR in a static outdoor scenario with a static point of view. They derive illumination under different weather conditions (e.g. clouded) without knowledge of the scene geometry, material, or texture and estimate the intensity of sunlight and skylight. In an offline learning stage, they employ a set of sample images of the scene all taken from the same point of view under the same sun position but different light conditions (like full sun and cloudy sky). From these images they learn two basis images, where one image contains the appearance of the scene under sunlight, while the other one contains the appearance under skylight. Different illuminations of the scene under this sun position can then be represented as a linear combination of these two basis images. Within the online registration stage, light parameters can be estimated for an input frame, by finding the appropriate linear weight. The learning and estimation procedure is further extended to account for different sun positions. The learning process requires at least two images per sun position which need to be captured under two different weather conditions. The overall collection process however lasted a whole year, so that all possible sun positions were sampled.

Lalonde and Matthews [Lalo 14] apply a model for sun and sky light too, however not in the context of AR. They estimate the lighting conditions for images from outdoor image collections. Based on image collections, structure-from-motion pipelines are able to reconstruct 3-dimensional models of the captured buildings. Lalonde and Matthews build upon these 3-dimensional reconstructions to additionally recover the outdoor illumination conditions for each of the original images using inverse rendering. They use a model of sun position and intensity as well as sky color and intensity and recover high-dynamic-range lighting environments ranging from overcast sky to full direct sunlight. For their inverse rendering, they first create a database containing collections of images of 22 different landmarks. For each image in these collections they at the same time also capture high-dynamic-range images of the entire sky hemisphere using a fish eye lens and multiple exposures. They fit their model for sun and sky light to the captured light probes. Additionally they compute priors for lighting and reflectance from the captured database. Their inverse rendering then starts by estimating reflectance for the vertices of a reconstructed 3D model based on automatically-detected overcast images from the corresponding image collection for the 3D model. Additionally they pre-compute the visibility of the hemisphere for each vertex of the mesh. Starting from there, they use an alternating

approach to estimate and refine lighting and reflectance taking into consideration occlusions and cast shadows. While their approach can only precisely recover the lighting conditions when an image contains strongly visible effects like cast shadows or strong differences in shading due to orientations, their estimated sky probes exhibit a high similarity to the ground truth light probes. For images with bright sunlight they report a median error in estimated sun direction of 17°.

2.2.3 Related Work on Illumination of the Face

Our method estimates the lighting conditions present in the real world from a single monocular image of the user's face. A publication which takes a similar direction in the context of Augmented Reality as our approach is the one presented by Koc and Balcisoy [Koc 13]. They also capture an image of the face to estimate the illumination. In contrast to our work their approach works offline and is limited to a single dominant light direction as they focus on outdoor use and direct sunlight. For a captured image of a face, they first align a geometrical face model to the image in an offline procedure and then estimate the direction of sun light based on the reflection direction at the brightest spot. Our approach in contrast does not have an explicit geometrical face model but learns in advance the whole radiance transfer for a set of sparse sample positions based on a dataset of images. Additionally our approach runs in real time and is able to estimate the whole directional distribution of incident light.

2.2.3.1 Face Relighting

Illumination of the human face is also studied in different domains. Beyond the area of AR, active research regarding illumination of the human face has been performed e.g. in computer graphics, particularly in the field of *relighting*, i.e. rendering of faces under new illumination. Debevec *et al.* [Debe 00] acquire the light reflected from a human face by capturing images of the same face under dense sampling of incident illumination directions using a so called *Light Stage*. From the captured images they construct a reflectance function for each image pixel. These reflectance functions correspond to our Radiance Transfer Functions and capture the overall reflected light at a surface position for light incident out of a particular direction. By employing these functions Debevec *et al.* are able to directly create new images of the face under any form of illumination.

In contrast to their work, we use the Radiance Transfer Functions to recover the incident light for a particular image of a face. While Debevec *et al.* work with reflectance functions stored as images, we project our Radiance Transfer Functions into Spherical Harmonics. As we do not aim at creating new images of the same face under different illumination, we do not recover the functions for each image pixel of a face but for a sparse distribution of appropriate locations on the face.

Fuchs *et al.* [Fuch 05] analyze spatially varying reflectance properties of a particular human face by taking photos in calibrated environments under different poses and up to seven point-light conditions. They estimate the geometry of the particular face using a 3-dimensional Morphable Model [Blan 99].

Additionally they fit parameters of an analytic BRDF (Bidirectional Reflectance Distribution Function [Nico 77]) model to the measured reflectance for different regions in the face together with a fine-grained locally varying diffuse term. This allows rendering the face under new poses and changed complex lighting conditions. The authors also map the acquired reflectance properties of one face onto another face based on facial features.

Nishino and Nayar [Nish 04] compute the environment map of the scene from the reflections of the surrounding world visible in the image of an eye and use the result for light estimation, face relighting, as well as for reconstruction of facial geometry. While they demonstrate good results for their light estimation, they require a quite high camera resolution so that at least a clear image of the eyes is captured. As we distribute samples sparsely over the face, our approach will work under less optimal conditions. Also illumination incident from above the user is partly blocked by the frontal bone and thus is not visible in the reflections on the eye balls.

2.2.3.2 Face Recognition

The illumination of faces is also highly relevant in the area of *face recognition*, as lighting has a large impact on appearance and interferes with the goal to determine a person's identity. Work in this domain mostly targets at making a face recognition method invariant to changes in illumination.

In order to still recognize a face when the illumination has changed Georghiades *et al.* [Geor 01] build the illumination cone, i.e. the set of all images of a face in a fixed pose, but under all possible illumination conditions, by reconstructing shape and albedo for a particular face from seven images of the same face and pose under different lighting directions.

Also Sim and Kanade [Sim 01] create new images of a face under changed illumination for better face recognition. Their method requires only a single image of an unknown face under unknown illumination. Their shading model is based on the diffuse Lambertian equation [Lamb 92] which they extend by an additive per pixel error term. By that error term they account for cast shadows and specular reflections which are not modeled by the Lambertian equation. From a set of images of people under different known illumination directions, Sim and Kanade [Sim 01] learn a statistical model for the normals as well as for the error term depending on the location on the face. The incident light direction for a new image is simply estimated based on the difference between the image and each training image using a Gaussian weighted sum over the corresponding known light directions. For the images out of the training set itself, they demonstrate high accuracy on the recovered light direction. While we, in our approach, also learn properties of the human face from a set of images of people under different known illumination directions, we neither explicitly recover surface normals nor assume diffuse reflection. Rather, we directly capture the overall radiance transfer. Furthermore we also do not estimate the illumination by image comparison but by solving a system of equations based on intensities and radiance transfer functions.

Zhang and Samaras [Zhan 03] also create a statistical model for the illumination of the human face. They however employ a collection of 3-dimensional face scans. Considering only the surface orientations of the 3-dimensional scans for the illumination, and thereby assuming a convex diffuse object, they compute per pixel means and covariances of Gaussian distributions for the influence of different Spherical Harmonics (SH) basis functions on image brightness. Afterwards they estimate an additional error term for the statistical model based on images of faces under known lighting. This error term comprises deviations from the diffuse as well as the convex assumption. Based on this model, Zhang and Samaras estimate the SH coefficients of the unknown illumination for a given face image using kernel regression [Atke 97]. Our method also models the influence of the light using SH basis functions and estimates SH coefficients of the unknown illumination. We however do not rely on 3-dimensional models of human faces, but directly learn from images. We also do not first assume convex diffuse objects and compensate for it later by an error term but directly capture the real Radiance Transfer Functions.

Also Qing *et al.* [Qing 04] use 3-dimensional models of human faces and create a multitude of images that show the influence of different SH basis functions, used to model the illumination, on the brightness of the faces. Again only the surface orientations of the 3-dimensional models are considered. Average images for the influence of a particular SH basis function, obtained by a principal component analysis, are used to estimate the unknown illumination for a given unknown face image. In our approach, we also analyze the influence of different SH basis functions modeling the illumination on the appearance of the face. In contrast to Qing *et al.* we however use real images of illuminated faces and thereby capture additional effects that are ignored by Qing *et al.* like cast shadow, specular reflection, or subsurface scattering.

All the presented approaches above are interested in the entire area of the face. As we are only interested in the lighting conditions and not in re-lighting or recognizing the whole image of the face itself, we focus on identifying sparse sample positions in the region of the human face that are well suited for estimating the illumination instead of using the whole image area of the face.

2.2.4 Related Work for Photo-Realistic Rendering in Augmented Reality

Once the real-world illumination has been acquired, we want to use it in order to create a plausible photo-realistic augmented image. The focus in this thesis however lies on the acquisition of the illumination present in the real world. Our particular simple implementation for rendering is described in section 2.5.7.

Creating photo-realistic augmented images involves many steps. The virtual part and the real part must fully interact in terms of light transport. Obvious examples are shadows cast from virtual objects onto the real ground plane, but also effects like the visiblity of virtual objects in mirrors of the real world or light reflected from virtual objects onto the real world are part of the so called *global illumination* problem.

Special attention must be paid in Augmented Reality to the fact that the video image showing the real world is already given and must be *adjusted* to mimic the impact of the added virtual objects, while the part of the final image showing the virtual objects must be *fully computer-generated* in such a way that it visually fits into the real scene. A common approach for combining the image of the real world with computer-generated content is called *differential rendering* [Debe 97]. Two global illumination solutions are calculated: one, where the scene only contains proxy objects for the real world; and another one, where the scene contains both the proxy objects for the real world as well as the to be added virtual objects. The difference between the two global illumination solutions then is used to adjust the video image of the real world. Ideally, the real world has to be known in terms of geometry and material properties, to provide realistic proxy objects. Differential rendering however forgives some degree of inaccuracy.

The challenge in differential rendering for AR is to approximate the global illumination solution as good as possible while still achieving real-time rendering frame rates. Knecht *et al.* [Knec 12] for example present a method that combines differential rendering with instant radiosity by Keller [Kell 97] to approximate the global illumination. Mehta *et al.* [Meht 15] use a two-mode path tracing approach for calculating the mutual illumination between the real and virtual objects. Their approach operates partially on GPU for the virtual geometry as well as in screen space for the real geometry which is reconstructed using a Kinect camera. By filtering the image they eliminate the noise resulting from the sparsely-sampled Monte-Carlo integration [Hamm 64], which is a method for numerical integration using random numbers. For capturing the environment lighting, Mehta *et al.* [Meht 15] use images of a mirror sphere.

For a photo-realistic augmented image and a coherent appearance of the virtual parts and the real parts it is also important to simulate existing camera effects for the virtual content, as shown by Klein and Murray [Klei 10], who model artifacts arising during the imaging process, such as distortions, chromatic aberrations, blur, and noise.

Sophisticated rendering methods like these are outside the scope of this thesis, which focuses on estimating the real-world lighting conditions.

2.2.5 Summary

Different approaches exist for acquiring the real-world illumination for AR applications – from direct light source capturing to the reconstruction of the illumination from reflections or shadows visible in the scene. A common approximation in most approaches is to model the light sources in terms of directional light coming from a distant scene, although some approaches also aim at estimating the position of point light sources.

While direct light source capturing is popular as it is straight forward and delivers a high resolution of the light distributions at a location, it requires additional hardware such as a fish-eye camera or a mirror ball, as well as an extra capturing or setup step.

Reconstructing the illumination from shading or cast shadow, on the other hand, requires knowledge about the scene geometry, so that existing approaches either rely on known objects or a reconstruction of the scene. If a method is working with special known objects like markers, these objects need to be available to the user and must be positioned in front of the camera, which both limits the convenience for the user. With the increasing distribution of depth cameras, the requirement for pre-modeled geometry will diminish. At the moment however most widely-used webcams and smartphones still do not feature depth cameras. In addition current approaches with RGB-D cameras primarily focus on texture-less materials because of the ambiguity between light and material. Another problem of these approaches is the high computational cost which limits the real-time feasibility of the algorithms especially on mobile devices.

2.2.6 Our Approach

Our approach, that we will present in the following, does not require a specialized depth camera but works with a simple monocular intensity camera which nowadays is already an integral component of current smart phones, tablets, and notebooks. We employ the face of the user as kind of a known object. Therefore we learn in a pre-processing step how faces reflect incident light and then use this knowledge to estimate the present illumination from the appearance of the user's face. By focusing on the face of the user we also overcome the need for additional special known objects like markers, that have to be available and actively added to the scene. The user's face can be conveniently captured by a user-facing camera at any time.

2.3 Benefits and Drawbacks of Using the Face as Light Probe

For consumer-targeted AR scenarios, employing the user's face as a *light probe* in order to estimate the present illumination has a number of benefits compared to other approaches. It however also brings along some drawbacks. We want to discuss both in the following section.

In contrast to other light estimation methods that rely on special known objects which have to be explicitly placed into the scene, our approach releases the user from both the duties to firstly have this special object at hand and to secondly actively place the object into the scene. It also releases the user of having to actively point the camera towards the particular known object, that is used for the light estimation. When the user faces the display of a hand held device, the user's face can be conveniently captured by the user-facing camera located next to the display. Thus, as long as the user observes the screen our method is able to immediately and continuously estimate the illumination. This estimation procedure thereby can be performed unnoticed and becomes less disruptive for the AR experience.

Our method is a perfect fit for AR applications that present the user an augmented camera stream of the user-facing camera, e.g. virtual fitting of clothes, jewelry, or glasses, or video chat applications featuring augmentations. The closeness of the augmented virtual object to the face here loosens the distant scene assumption, as we estimate the incident light near the virtual object's location.

Another reason why the illumination estimated from the image of the user's face is particularly suited for illuminating virtual objects on the camera stream of the *user-facing* camera goes along with a weakness of this approach. In the image of the user-facing camera we only see those surfaces of the user's face that are oriented towards the camera. In other words, we cannot see surfaces facing away from the camera. Due to this limited visibility of surface orientations, the image of the user's face especially exhibits lighting effects caused by light incident out of directions in front or beside of the user. Consequentially it is also this part of the incident illumination that can be estimated most reliably from the image. When we render virtual objects onto the camera stream of the user-facing camera, luckily for these objects the same applies as for the face. We again only see those surfaces of the virtual objects that are oriented towards the camera. It is thus the same part of the incident illumination that has the biggest impact on the appearance of the virtual objects.

The limited information contained in the image of the user's face about the light that is coming out of directions behind the user leads to an increased uncertainty in the estimation for light out of these directions. This impacts the suitability of our light estimation for lighting virtual objects augmented on the camera stream of the *world-facing* camera. We will get back to this problem in section 2.5.6.2, where we will show that this problem can be partially addressed by capturing multiple images of the face under different orientations and fusing the information contained in the different images. Note, that light incident out of directions behind the user still has some influence on the appearance of the face. Subtly in form of subsurface scattering but also more strikingly in form of diffuse and glossy reflections. We plan to further address this limitation of our method in future work by extracting

additional information about the light coming from behind the user from the background image region around the user's face. This information then could be fused together with our current estimation.

While at the moment directly using the estimated illumination one-to-one for renderings on the back-facing camera is somehow limited, it is still possible to deduce information from the estimation, as for example demonstrated in an on-stage presentation at the InsideAR 2014 conference, where we used the illumination estimated from the user's face captured by a user-facing camera to select from a predefined set of illumination configurations for rendering augmentations for a world-facing camera.

Restricting our method to inverse lighting of the human face has some benefits compared to state-of-the-art approaches that estimate the illumination by inverse lighting on *arbitrary environments*.

Except for some corner cases like fully tattooed faces or faces with a full beard, the suitability of the human face for estimating the illumination is known in advance. Our algorithm is tailored to the human face and we deduce information by exploiting the different surface orientations and concavities present in all the human faces. For light estimation methods working on arbitrary scenes [Grub 12, Bute 15], there is no guarantee that a particular scene is suited at all for the particular light estimation method. Approaches that rely on diffuse geometry and exploit different surface orientations (e.g. [Grub 12]) would fail for a very glossy scene or for a scene where the camera only sees the top of a planar table. Approaches that rely on planar uniform scenes and point lights (e.g. [Bute 15]) on the other hand would fail for scenes that do not feature these planar uniform surfaces.

While we have the guarantee that the human face is fairly suited to estimate the incident illumination, the face on the other hand has less ideal properties than objects that are *specifically* targeted for light estimation like mirror spheres [Debe 98] or 3D-printed shading probes [Cali 13]. The face is neither fully specular like a mirror sphere nor does it contain all these uniformly distributed concavities of the 3D-printed shading probe. The specularities as well as the concavities both cut out and separate the lighting effects of different parts of the illuminating hemisphere. The appearance of most locations on the face however is strongly influenced by all of the light incident out of the upper hemisphere of directions. As derived by Ramamoorthi and Hanrahan [Rama 01] this makes the problem of inverse lighting, which can be seen as a de-convolution of the radiance leaving a surface into incident illumination and radiance transfer properties of the surface, harder to solve. As a consequence our method is only able to recover a low frequency approximation of the incident illumination.

A key benefit of our method is the low hardware requirement. Methods that estimate the illumination based on arbitrary scenes need to acquire and process the geometry of the scene on-the-fly, e.g. using RGB-D cameras [Grub 12, Bute 15]. This imposes additional requirements with regard to available hardware as well as with regard to processing time.

As human faces have a limited range of variations in geometry and reflectance properties between different individuals, these properties can be determined beforehand in an offline pre-process. In this offline process the properties can either be modeled manually or learned automatically e.g. from a multitude of different example faces. Constraining the scene reconstruction problem on faces would

also make it possible to fit a generic 3-dimensional face model [Blan 99] using a single image captured by a standard RGB camera. With the face of the user staying the same over time, this fitting would only have to be done once, compared to arbitrary parts of the scene geometry, which change while the user is moving through the scene.

Splitting up the light estimation method into a pre-processing step and a live estimation step enables algorithms that are optimized for faces based on valid assumptions and restrictions.

The live estimation part then can already be provided with information about the geometry and reflectance, so that no special sensors like RGB-D cameras are needed to acquire the face. This makes it feasible to run the estimation on images captured by a conventional monocular intensity camera, e.g. a webcam or the camera of a smartphone. Removing the expensive scene reconstruction part from the live estimation part also makes the estimation run more efficiently than generic approaches. This is beneficial for mobile devices, which have limited processing power and where a low power consumption is essential. Both these points, i.e. lower requirements in hardware as well as a lower power consumption, are especially useful for AR applications for the consumer market.

The restriction to human faces also mitigates another problem that exists for approaches working with arbitrary and unknown scenes, namely an ambiguity between light and material. While reconstructing the geometry of a scene can be enabled by depth cameras, acquiring material properties on-the-fly under unknown lighting conditions is difficult and typically hardly possible from a single image. Therefore methods try to tackle this issue by making certain assumptions about the materials in the scene, which however may be invalid. As we already know that we are working with a human face, we are able to make stronger assumptions that hold true, like a specific model for skin reflectance which constrains the physical problem of ambiguity between the surface material and the light in terms of intensity and color. We even can predefine or pre-learn those regions of the human face that are particularly suited for estimating the illumination and discard other regions that are unreliable.

In summary, focusing on the human face lets us tailor our algorithm on that specific object. It allows us to make valid assumptions and to take pre-processing steps, which lower the requirements in terms of hardware and processing power during run time. Selecting the face of the user as known object also decreases the tasks the user has to perform, as it eliminates the need for an additional known object as well as an additional set-up and capture step. While observing the face as light probe gives a certain guarantee for the suitability for estimating the illumination it also limits the performance in recovered lighting to low frequencies. It particularly is suited for estimating light incident from in front of the user, and by that for enabling a coherent illumination of augmentations on the image of the user-facing camera.

2.4 Basic Knowledge of Spherical Harmonics

This section provides a short introduction to *Spherical Harmonics (SH)*, an orthonormal function basis that is defined over the surface of the unit sphere, which is equivalent to the domain of all directions in $\mathbb{R}^3$. We will employ this basis throughout our light estimation approach, for modeling the incident illumination as well as for modeling the reflection properties of the human face.

We provide mathematical definitions in section 2.4.1 and emphasize important properties that we will rely on later. We then explain in section 2.4.2 how we align the SH functions to the coordinate system of the human face. Finally, we show how to visualize functions that are defined over the surface of the unit sphere in section 2.4.3.

Please refer to [Gree 03] and [Sloa 08] for a deeper insight on SH in the domain of computer graphics and lighting.

2.4.1 Mathematical Definition

Spherical Harmonics (SH) are functions that are defined on the surface of the unit sphere. They form a complete set of orthonormal functions, which makes them effective as basis for representing functions on the sphere as linear combination.

2.4.1.1 Notations

Let S^2 describe the set of points $(x,y,z)^\top \in \mathbb{R}^3$ with $x^2+y^2+z^2=1$ [Fenn 01]. This set of points can either be interpreted as the surface of the unit sphere or equivalently as the set of all directions in $\mathbb{R}^3$. We thus synonymously also say that SH are defined over all directions. As each basis function is parameterized by direction, it can be both formulated in spherical and Cartesian coordinates. Depending on the use case we will prefer either the former or the latter coordinate system.

We will only make use of the real part of the SH, while they are in fact functions in the complex numbers. A particular SH function for us is defined as

$$Y_\ell^m(\theta,\phi)\colon S^2 \to \mathbb{R}.$$

The radius of the spherical coordinates thereby is set fix to unity, $\theta \in [-\pi/2,\pi/2]$ corresponds to the elevation angle or latitude, and $\phi \in (-\pi,\pi]$ to the azimuth angle or longitude.

The index $\ell \in \{0,...,\infty\}$ specifies the degree or band ℓ of the particular SH function, while the index $m \in \{-\ell,...,\ell\}$ indicates the order m within this band.

Parameterized in spherical coordinates, SH functions can be written as a tensor product between so called Legendre polynomials and the Fourier basis. This formula however is out of scope for this

work. What is important for us from that definition is that with increasing degree ℓ the SH functions contain higher frequencies in their variation over the unit sphere.

Instead of the two indices ℓ and m we from time to time will also make use of a linearized single index notation [Sloa 08], which lets us write Y_ℓ^m as Y_n with:

$$n = \ell(\ell+1)+m \tag{2.1}$$

Like mentioned above, instead of using spherical coordinates, a direction can also be written in Cartesian coordinates: $\vec{\omega} = (x,y,z)^\top \in \mathbb{R}^3$ with $|\vec{\omega}| = 1$. The two representations can be easily transformed into each other:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\sin\phi\cos\theta \\ \sin\theta \\ \cos\phi\cos\theta \end{pmatrix} \tag{2.2}$$

$$\begin{pmatrix} \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \arcsin y \\ -\arctan 2(x,z) \end{pmatrix} \tag{2.3}$$

The function arctan2 thereby describes a variant of the arc tangent function that additionally considers the appropriate quadrant of the computed angle based on the signs of the two input variables.

Instead of $Y_n(\theta,\phi)$ we thus can equally write $Y_n(\vec{\omega})$ and we can also transform the corresponding formulas of the SH basis functions from spherical coordinates into Cartesian coordinates.

2.4.1.2 Spherical Harmonics as Function Basis

SH can be used to represent any real-valued square-integrable function $f\colon S^2 \to \mathbb{R}$ that is parameterized by direction as a linear combination of the basis functions:

$$f(\vec{\omega}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_{m,\ell} \cdot Y_\ell^m(\vec{\omega}) = \sum_{n=0}^{\infty} f_n \cdot Y_n(\vec{\omega}) \tag{2.4}$$

The coefficients f_n that scale the different SH basis functions describe the particular function f.

If we cut the infinite sum by introducing a maximum degree of L for the SH functions, we get a sum over $(L+1)^2$ elements. We thereby omit all the higher frequencies, so that the linear combination of the remaining basis functions approximates the function f :

$$f(\vec{\omega}) \approx \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} f_{m,\ell} \cdot Y_\ell^m(\vec{\omega}) = \sum_{n=0}^{(L+1)^2-1} f_n \cdot Y_n(\vec{\omega}) \tag{2.5}$$

The higher the maximum degree L, the higher frequencies can be represented. In this work, we will only use SH basis functions up to maximum degree $L = 2$. This low degree will limit our SH basis to very smooth functions that do not contain abrupt changes like edges.

2.4.1.3 The First Nine SH Functions

The maximum degree $L = 2$ gives us nine SH basis functions Y_n and correspondingly nine coefficients f_n for the approximation of a function f. These nine coefficients $f_0, f_1, ..., f_8$ that describe the function f can be written as a SH coefficient vector $\hat{f} \in \mathbb{R}^9$ with $\hat{f} = (f_0, f_1, \cdots, f_8)^\top$.

Using Cartesian coordinates, the functions in the first three bands of the SH that are relevant for us can be compactly written as:

$$
\begin{aligned}
Y_0(\vec{\omega}) &= Y_0^0\ (\vec{\omega}) = \quad 0.5\ \cdot\sqrt{\frac{1}{\pi}} \\
\\
Y_1(\vec{\omega}) &= Y_1^{-1}(\vec{\omega}) = -0.5\ \cdot\sqrt{\frac{3}{\pi}}\cdot y \\
Y_2(\vec{\omega}) &= Y_1^0\ (\vec{\omega}) = \quad 0.5\ \cdot\sqrt{\frac{3}{\pi}}\cdot z \\
Y_3(\vec{\omega}) &= Y_1^1\ (\vec{\omega}) = -0.5\ \cdot\sqrt{\frac{3}{\pi}}\cdot x \\
\\
Y_4(\vec{\omega}) &= Y_2^{-2}(\vec{\omega}) = \quad 0.5\ \cdot\sqrt{\frac{15}{\pi}}\cdot yx \\
Y_5(\vec{\omega}) &= Y_2^{-1}(\vec{\omega}) = -0.5\ \cdot\sqrt{\frac{15}{\pi}}\cdot yz \\
Y_6(\vec{\omega}) &= Y_2^0\ (\vec{\omega}) = \quad 0.25\cdot\sqrt{\frac{5}{\pi}}\cdot(3z^2-1) \\
Y_7(\vec{\omega}) &= Y_2^1\ (\vec{\omega}) = -0.5\ \cdot\sqrt{\frac{15}{\pi}}\cdot xz \\
Y_8(\vec{\omega}) &= Y_2^2\ (\vec{\omega}) = \quad 0.25\cdot\sqrt{\frac{15}{\pi}}\cdot(x^2-y^2)
\end{aligned}
\tag{2.6}
$$

The first SH function Y_0 with $\ell = 0$ contributes the constant term, while the three SH functions Y_1, Y_2, Y_3 with $\ell = 1$ represent a linear dependence. The five SH functions $Y_4, ..., Y_8$ with $\ell = 2$ correspond to quadratic terms.

The constant scale factors of the functions take care of the normaliziation of each function, so that for all functions Y_n:

$$\int_{S^2} (Y_n(\vec{\omega}))^2 \, \mathrm{d}\vec{\omega} = 1 \tag{2.7}$$

2.4.1.4 Orthonormal Property

A big benefit of the SH basis is its orthonormal property. That means that the SH basis functions are orthonormal to each other. The integral of the product of two SH basis functions Y_a and Y_b over all directions thus is either 1 if the two functions are one and the same, or 0 otherwise.

$$\int_{S^2} Y_a(\vec{\omega}) \cdot Y_b(\vec{\omega}) \, \mathrm{d}\vec{\omega} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{else} \end{cases} \tag{2.8}$$

Thanks to the orthonormal property, we can determine the coefficients f_n that best approximate a function $f(\vec{\omega})$ by simply projecting the function f onto each particular basis function Y_n:

$$f_n = \int_{S^2} f(\vec{\omega}) \cdot Y_n(\vec{\omega}) \, \mathrm{d}\vec{\omega} \tag{2.9}$$

Another advantage of the orthonormal property becomes evident when we calculate the integral over all directions of the product of two functions f and g that are both specified in the SH basis approximation. By rearranging the terms and by applying equation (2.8) it becomes evident, that the integral can be simply calculated as the dot product of the two SH coefficient vectors $\hat{f}$ and $\hat{g}$.

$$\int_{S^2} f(\vec{\omega}) \cdot g(\vec{\omega}) \, \mathrm{d}\vec{\omega} = \int_{S^2} \sum_{i=0}^{8} \hat{f}_i \cdot Y_i(\vec{\omega}) \cdot \sum_{j=0}^{8} \hat{g}_j \cdot Y_j(\vec{\omega}) \, \mathrm{d}\vec{\omega} \tag{2.10}$$

$$= \sum_{i=0}^{8} \sum_{j=0}^{8} \hat{f}_i \hat{g}_j \int_{S^2} Y_i(\vec{\omega}) \cdot Y_j(\vec{\omega}) \, \mathrm{d}\vec{\omega} \tag{2.11}$$

$$= \sum_{n=0}^{8} \hat{f}_n \hat{g}_n \tag{2.12}$$

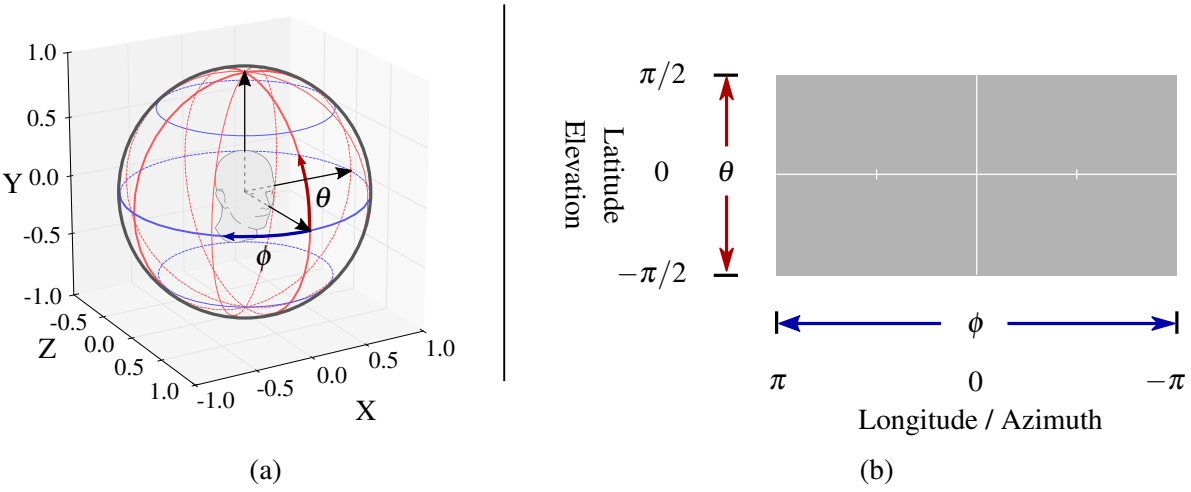
$$= \hat{f}^\top \cdot \hat{g} \tag{2.13}$$

2.4.2 Coordinate System with Respect to the Human Face

In this work we want to employ Spherical Harmonics to describe quantities that depend on directions with respect to the human face. Figure 2.4 (a) illustrates how we embed the orientation of the human face into the spherical coordinates system. The figure also shows the axes of the Cartesian coordinate system that we define with respect to the face.

The coordinates $(\theta,\phi)^\top = (0,0)^\top$, or $(x,y,z)^\top = (0,0,1)^\top$, correspond to the front-facing direction. The rotation around yaw – the y-axis – is represented by angle ϕ, while θ describes the elevation. The coordinates $(\theta,\phi)^\top = (\pi/2,0)^\top$, or $(x,y,z)^\top = (0,1,0)^\top$, correspond to the up-facing direction. Note that the representation in spherical coordinates is not always unique, as for example multiple coordinates $(\theta,\phi)^\top = (\pi/2,\cdot)^\top$ correspond to the same direction to the top. The same is true for the direction $(\theta,\phi)^\top = (-\pi/2,\cdot)^\top$ pointing down.

The presented embedding of the face into the coordinate systems will be used in the following both to define the incident illumination with respect to the face as well as to define how light is reflected by the face depending on the incident light direction.



(a) (b)

Figure 2.4: The human face is embedded into a spherical coordinate system (a), which enables us to define functions in Spherical Harmonics with respect to the human face. Functions defined over the set of unit directions can also be plotted in form of Latitude Longitude (Lat-Long) images (b).

2.4.3 Visualization

The order $m \in \{-\ell, ..., \ell\}$ of the SH basis establishes a pyramidal structure of the functions, as each band ℓ adds $2\ell + 1$ basis functions. We will use this structure when we plot the individual SH basis functions or quantities related to them and we will refer to this ordering of the subplots as *pyramidal SH structure*:

Table 2.1: Pyramidal SH structure induced by $m \in \{-\ell, ..., \ell\}$.

		m				
		-2	-1	0	1	2
ℓ	0			Y_0		
	1		Y_1	Y_2	Y_3	
	2	Y_4	Y_5	Y_6	Y_7	Y_8

Let's have a look at the different SH basis functions. Multiple ways exist to illustrate a function that is defined over the domain of directions. We will use two of them.

Firstly we visualize the functions geometrically in 3-dimensional space, like for example in figure 2.5. This figure shows the nine SH basis functions with maximum degree $L = 2$. In this geometrical visualization, the absolute value of the function for a direction $(\theta, \phi)^\top$ determines the extension of the displayed shape in this direction. For a SH basis function Y_n this results in surface points $(|Y_n(\theta, \phi)|, \theta, \phi)^\top$ (specified in spherical coordinates). This is equivalent to the point $|Y_n(\vec{\omega})| \cdot \vec{\omega}$ in Cartesian coordinates, which represents the unit vector in direction $\vec{\omega}$ scaled by the absolute value $|Y_n(\vec{\omega})|$. A higher absolute value of a function for a particular direction thus correlates in the figure with a higher distance from the origin in that direction. In order to distinguish between positive and negative values, positive values are encoded in green, negative ones in red.

While this illustration mode is quite demonstrative, it has the problem that not all directions can be visualized at once as some are occluded. Also the projection of the 3-dimensional figure onto two dimension is ambiguous.

We thus also employ a second way to plot a function defined over the domain of directions: *Latitude Longitude (Lat-Long) images*. Figure 2.4 (b) illustrates the mapping in our Lat-Long images. A coordinate $(\theta, \phi)^\top$ of a direction is simply interpreted as a 2-dimensional pixel coordinate. The center pixel of the image corresponds to $(\theta, \phi)^\top = (0, 0)^\top$. Note that the longitude axis is positive to the left for a consistent mapping from the spherical coordinates (figure 2.4 (a)). The projection from the sphere onto the rectangle implies severe distortions especially at the poles.

Figure 2.6 shows again the nine SH basis functions with $L = 2$, this time plotted as Lat-Long images. Like before positive values are encoded in green and negative ones in red. This time however the absolute value $|Y_n(\theta, \phi)|$ determines how bright the pixel is. A value of 0 results in a black pixel.

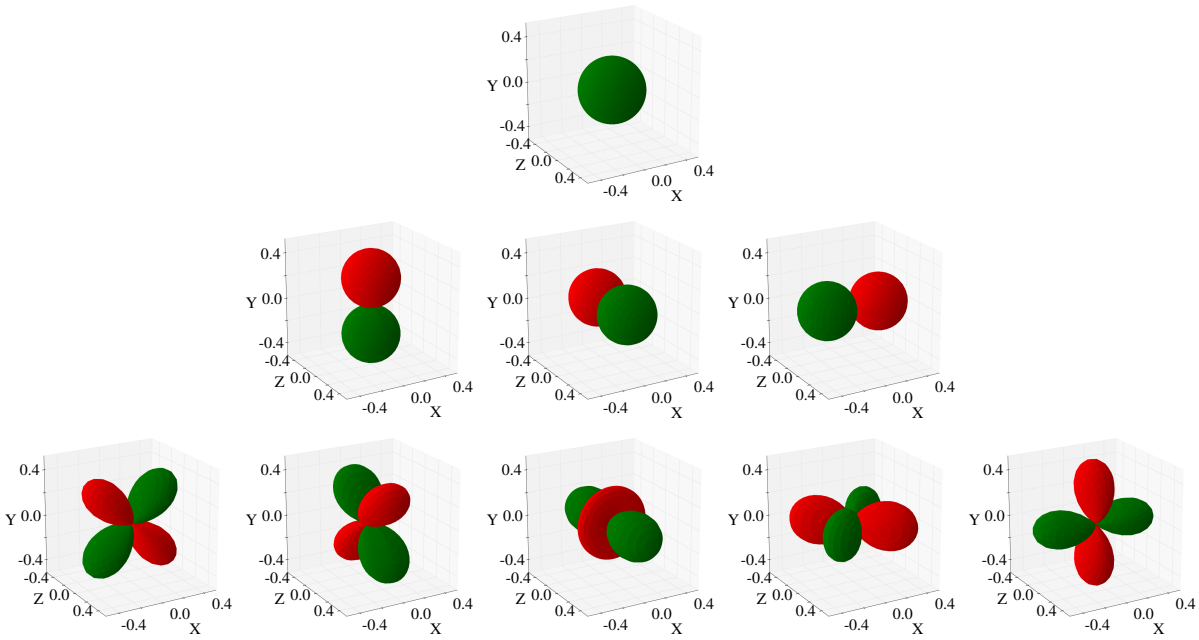


Figure 2.5: The first nine SH basis functions plotted geometrically in pyramidal SH structure (in accordance with table 2.1).

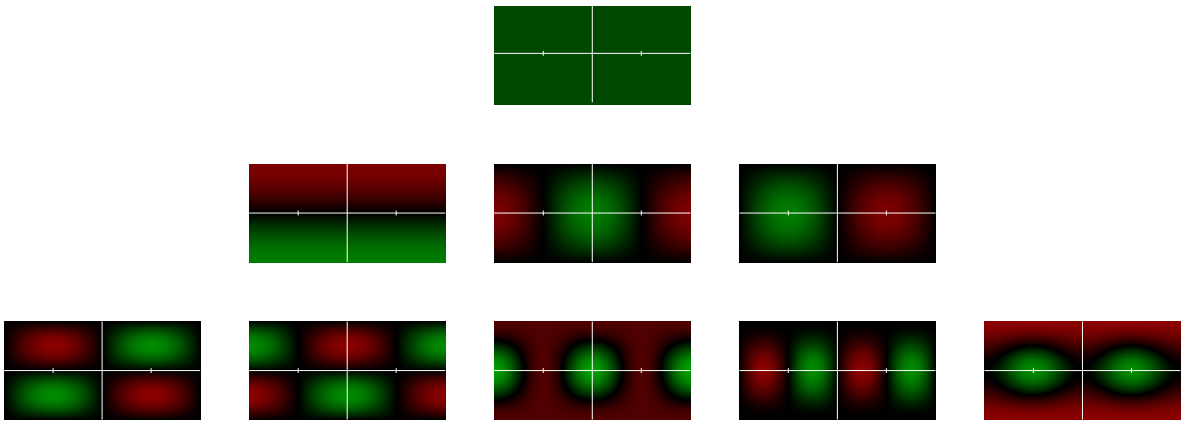


Figure 2.6: The first nine SH basis functions plotted as Lat-Long images in pyramidal SH structure (in accordance with table 2.1).

2.5 Method of Estimating Light from the Image of a Face

In this section we present our particular approach for estimating the incident light based on a single monocular image of a human face in real time.

We first outline in section 2.5.1 how to mathematically describe the distribution of light within a scene as well as the propagation of light, as it emerges from light sources and interacts with the objects in the scene.

In section 2.5.2 we then explain our specific setting in which we want to employ our method to estimate the illumination. This setting involves the face of the user and a user-facing camera capturing an image thereof. From this setting, we deduce a separation of the light in our scene into light incident from the distant environment and light leaving a particular position on the face.

These two parts of the light obviously are linked, the more light is incident from the distant environment onto the face, the more light is leaving a particular position on the face. The correlation between these two intensities can be described by *Radiance Transfer Functions (RTFs)*, which we explain in section 2.5.3. We hereby are especially interested in light leaving the face towards the camera. Loosely speaking in our case the RTFs encode the ratio between the intensity of light leaving the face towards the camera and the intensity of light incident from the distant environment. An image of the face captured by a camera will contain multiple measurements of the light leaving the face towards the camera. Based on these measurements we finally want to estimate the distribution of incident light.

Our light estimation method consists of two steps, firstly an offline learning process and secondly a real-time light estimation that relies on the previously acquired knowledge.

We make use of the limited range in variations between different human faces in terms of shape and reflection properties. We thereby assume that the knowledge about how the human face in *average* appears under a certain illumination can be also applied to the *particular unknown* face belonging to the user. The face of the user then can be treated as an at least approximately known object and used as a light probe for estimating the incident light.

The offline learning process, described in detail in section 2.5.4, only has to be executed once in advance. In that step we analyze the appearance of human faces in terms of image brightness under different *known* illuminations. For this purpose we employ *The Extended Yale Face Database B* [Geor 01, Lee 05], which is publicly available. It contains a set of images of faces of different humans under directional illumination out of different known directions. For these images we know the reflected light provided by the pixel intensities as well as the incident light specified in primary light direction by the dataset.

We select a set of sparsely distributed locations on the human face. Examining each of these sample locations in separation of the others, we *learn the average RTF* per location over different humans. In our method we model both RTFs as well as the incident illumination as 2-dimensional functions which

depend on the incident light direction. This allows us to encode both of them in terms of Spherical Harmonics basis approximations (see section 2.4).

After we learned the average RTFs in the offline learning process, we then apply the functions to new images of an *unknown* face in section 2.5.5 in order to estimate the incident illumination present in the image. In our use case, the unknown face more precisely is the face of the user of an Augmented Reality application. We receive an image of the user's face as input. We then align the set of sample positions from the offline process to the new image by means of image-based face tracking. This allows us to extract the intensity values at the sample positions which we subsequently use together with the corresponding RTFs identified in the offline learning stage to *estimate the most plausible real-world lighting conditions* in real time.

In section 2.5.5 we will start by presenting a straight forward unconstrained least-squares solution for the real-time light estimation. We then will further improve our estimation by additionally modeling the physical restriction to positive light intensities in section 2.5.6. Here we also discuss remaining limitations of our method and propose how to approach problems like ambiguity in the solution space or deviations of the user's face from the previously learned average face.

The fundamental goal why we want to estimate the illumination present in the real world is a coherent illumination of the virtual objects in an Augmented Reality application. We demonstrate how to use the estimated illumination for the rendering of virtual objects in section 2.5.7.

2.5.1 Foundations of Light Transport

Before we have a closer look at how our method estimates in detail the incident light from an image of the human face, we in this section first provide well-established mathematical foundations for describing light and the transport thereof.

Light is an electromagnetic radiation which can be specified by its wavelength and intensity. The spectrum of light that is visible for humans roughly spans wavelengths in between 400 nm and 700 nm. Depending on its wavelength, light appears to us in a certain – what we refer to as – *color*. Cones in the retina of the human eye respond to incident light and transform the incident radiation into nerve impulses. To distinguish between different wavelengths of incident light, the human eye possesses three different types of cones, also referred to as color receptors. These three color receptor types differ in how sensitive they are to particular wavelengths. The human brain interprets the difference in the received signals from the different types of cones as color.

The way we perceive color allows us to make a simplification for our method which is common in computer graphics and computer vision. We do not consider the continuous spectrum of wavelengths for the light but instead only light at three discrete wavelengths corresponding to red, green, and blue. We further assume that light does not change its wavelength on interactions with surfaces. By that we neglect effects like e.g. fluorescence, where light is absorbed by a material and subsequently reemitted

at a different wavelength. This simplification allows us to consider each of the three wavelengths in separation of the others. We thus can also carry out our light estimation separately for each particular wavelength. In the following, mathematical derivations and quantities will consequently consider light of one specific wavelength. In order to later represent *colored* light, we will have one quantity for each relevant wavelength.

Our method will employ inverse lighting, which means it will estimate the incident light on an object from its appearance. Therefore it is important to understand how the appearance of an object is influenced by light.

Light originates at light sources, which are objects or media that emit light with a specific spectrum of wavelengths, intensities and directions. The emitted radiation then propagates through empty space on straight lines, until it interacts with matter. For simplification we assume vacuum instead of a scene filled with air, as we then can neglect marginal scatter and absorption effects in the empty space.

When light hits a surface it is either reflected, refracted or absorbed. The part of the light that is not absorbed continues to propagate through space until it again interacts with matter. This propagation goes on and on, until all light is absorbed. While we refer to as *reflecting* in case that the light is bouncing back on the surface and thereby stays in the same medium, we refer to as *refracting*, when the light is passing through a surface from one medium into another medium.

Refraction for example is happening, when light hits a surface like human skin and passes through into the skin. Within the skin, light then interacts with the matter, is scattered once or multiple times and potentially leaves the skin again at some other location. This process is called *subsurface scattering* and substantially contributes to the characteristic soft look of human skin. The influence of subsurface scattering is for example addressed by Weyrich *et al.* [Weyr 06] who present a skin reflectance model whose parameters can be estimated from measurements, as well as by d'Eon and Luebke [dEon 07] in the field of realistic real-time skin rendering.

In the following mathematical derivation of light transport and radiance transfer in this section we will for simplicity however deliberately omit the effects of refraction and subsurface scattering. We will consider only reflection, as if the face would be a fully opaque object. This allows for a simpler explanation of the main principle. Finally, for learning our Radiance Transfer Functions, we will however take the full light transport into consideration by working on real captured images. Consequently also our implementation for estimating light will itself automatically take refraction and subsurface scattering into account. We will point this out again later when relevant.

As described above, emitted light propagates until it has been absorbed. At the same time light sources will continue to emit light over time. For now we assume that the light sources in our scene have a constant emission over time. The distribution of light in our scene, that arises from the continuous emission of light by the light sources and the succeeding propagation of the light, then almost instantly reaches an equilibrium state due to the high speed of light.

This equilibrium solution can be mathematically formulated as an integral equation called *Rendering Equation* [Kaji 86]:

$$\begin{aligned} &L\colon \mathbb{R}^3 \times S^2 \to \mathbb{R}_0^+ \\ &L(x,\vec{\omega}) = L_e(x,\vec{\omega}) + L_r(x,\vec{\omega}) \end{aligned} \tag{2.14}$$

For one specific wavelength at one specific point in time, $L(x,\vec{\omega})$ specifies the radiance[1], loosely speaking power of light, at a surface point $x \in \mathbb{R}^3$ *into* direction $\vec{\omega} \in S^2$. This radiance itself is composed out of two parts: $L_e(x,\vec{\omega})$, radiance *emitted* at location x into direction $\vec{\omega}$ and $L_r(x,\vec{\omega})$, radiance *reflected* at location x into direction $\vec{\omega}$. L_e thereby corresponds to light directly emitted by light sources, and is defined by characteristics of the particular light source at x. The radiance L_r reflected at x on the other hand can be further disassembled, referred to as *Reflection Equation*:

$$L_r(x,\vec{\omega}) = \int_{\Omega(x)} f_r(x,\vec{\omega}_i,\vec{\omega}) L_i(x,\vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}(x))\,\mathrm{d}\vec{\omega}_i \tag{2.15}$$

The radiance L_r reflected at a location x into direction $\vec{\omega}$ originates from radiance L_i incident at x from all possible directions $\vec{\omega}_i \in \Omega(x)$, with $\Omega(x)$ specifying the upper unit hemisphere with respect to the surface orientation at position x. The surface orientation is represented by the outward-pointing unit-length vector $\vec{n}(x) \in \mathbb{R}^3$. By integrating over $\Omega(x)$ we collect all the light that is incident at x from above. Note that only considering the *upper* unit hemisphere at this point neglects refraction and thus subsurface scattering, as we do not collect light coming from below the surface.

The incident radiance L_i out of direction $\vec{\omega}_i$ is modulated by the cosine of the angle between $\vec{\omega}_i$ and $\vec{n}(x)$, which is calculated by the scalar product of the two unit length vectors. This modulation accounts only for the irradiance, i.e. the effective power incident on the unit area of the surface.

The resulting irradiance is multiplied by the so called *Bidirectional Reflectance Distribution Function (BRDF)* [Nico 77] $f_r(x,\vec{\omega}_i,\vec{\omega})$, which specifies the ratio of *locally* reflected radiance into outgoing direction $\vec{\omega}$ to *locally* incident irradiance out of direction $\vec{\omega}_i$. The BRDF captures the reflectance properties of the material at surface location x. It depends on both incident and outgoing direction. Compared to the Radiance Transfer Functions we will employ later, this function only models the *local* effect of reflection at this specific spot, i.e. how light that arrives at the surface position is reflected.

Radiance along a ray does not change as long as light travels through empty space. We thus can express the radiance L_i incident *out of* direction $\vec{\omega}_i$ at position x in equation (2.15) as outgoing radiance L *into* direction $(-\vec{\omega}_i)$ at the surface point that is visible from x in direction $\vec{\omega}_i$.

[1] radiance is measured in $W \cdot sr^{-1} \cdot m^{-2}$

Given a function $h(x, \vec{\omega}_i) \in \mathbb{R}^3$ which returns the surface point that is visible from point x in direction $\vec{\omega}_i$, we get:

$$L_i(x, \vec{\omega}_i) = L(h(x, \vec{\omega}_i), -\vec{\omega}_i) \tag{2.16}$$

The reflection equation – which is part of the rendering equation (2.14) – thus can be rewritten as:

$$L_r(x, \vec{\omega}) = \int_{\Omega(x)} f_r(x, \vec{\omega}_i, \vec{\omega}) L(h(x, \vec{\omega}_i), -\vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}(x))\, \mathrm{d}\vec{\omega}_i \tag{2.17}$$

Note that in the overall rendering equation (2.14) the unknown function L thereby occurs twice: firstly on the left and secondly inside of the integral of the reflection equation (2.17) on the right side, which makes the rendering equation a so called Fredholm equation of the second type (see for example [Poly 08]) which in general is hard to solve analytically.

A useful property of light worth noting lies in its linear nature - see the *rules of superposition* [Nime 95]. The radiance in a scene that is generated by two light sources is simply the sum of the radiance that each particular light source would cause individually. Analogously scaling the intensities of all the light sources by a certain factor causes the same scale in the radiance in the scene.

Given this basic knowledge about how to describe light mathematically we will now have a closer look on the specific scenario, in which we want to estimate the illumination.

2.5.2 The Illuminated Face – Local and Distant Scene

Our method is specialized on estimating the illumination that is incident on a human face based on an image thereof. In our target scenario, the face belongs to a person that is using an Augmented Reality application within some environment, either outdoors or indoors. Light sources in the surroundings, e.g. the sun, lanterns, or lamps, thereby illuminate the whole scene. While the user is looking at a screen which displays the augmented view of the Augmented Reality application, his face is captured by a user-facing camera.

We are interested in the overall incident light from the surroundings at the position of the user – more specific at the position of the user's face – in order to apply a coherent illumination for the virtual content. The term *overall incident light* refers to the fact that at this point we do not care whether the incident light at the position of the user is directly coming from a light source or whether it has already been reflected in the surrounding environment towards the face. We however assume that the surrounding environment as well as the light sources are located distant from the user, as this will simplify our estimation.

Similarly to Debevec [Debe 98] we partition the scene for our light estimation method into two parts. The first part of the scene is called *distant scene* and comprises both the light sources as well as the reflecting surrounding surfaces in the environment around. All we will care about of this distant scene is the incident light.

The second part is the *local scene*, i.e. the face of the user. While the face does not emit light by itself, the light incident out of the distant scene manifests itself by illuminating the face of the user.

By assuming that the distant scene is located far away from the face, the parallax effect regarding the incident light from the distant scene can be neglected for locations on the face. Figure 2.7 illustrates the concept behind. As long as a light source is close to the face (see the face on the left of figure 2.7), the direction of incident light from the light source varies quite strong between different positions on the face. With increasing distance between the light source and the lit object (see the face on the right of figure 2.7), this variation diminishes and incident light rays become more and more parallel. We thus treat light incident from the distant scene as only depending on the incident direction and no longer depending on the particular location on the face.

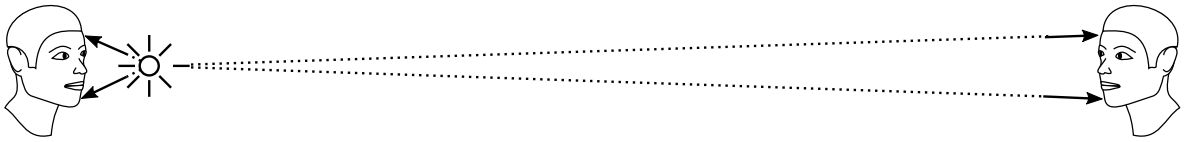


Figure 2.7: Incident light from a light source has a stronger variation in direction for the close face on the left than for the distant face on the right, where the light is incident nearly parallel.

The light incident from the distant scene can thus be specified as a 2-dimensional function $E(\vec{\omega}_i) \in \mathbb{R}_0^+$ which depends on incident direction only (see figure 2.8). In the following we will refer to E also as *directional distribution of incident light*. E is defined over the continuous range of directions – mapping from direction to light intensity incident from that direction. Note that $\vec{\omega}_i$ refers to the direction where light is incident *from* and not where it is heading in this case.

An obvious example where our assumption, that incident light does not vary with location, is not valid is e.g. the illumination of the face by a video projector. Here every region of the face can be lit individually. Another more natural example where the assumption does not hold is a shadow that is cast by the surroundings on only a part of the face.

The light E incident out of the distant scene illuminates the face. For the face, which corresponds to our local scene, we introduce $R(x, \vec{\omega}) \in \mathbb{R}_0^+$, which represents the light leaving at a surface point x of the local scene *into* the direction $\vec{\omega}$. As the face does not emit light by itself R depends on E. It also depends on the material and geometry properties of the local scene. For a particular point on the face some part of the distant environment may be occluded so that the surface point does not directly receive light from this part of the environment. This occlusion may manifest as a cast shadow. See for example figure 2.9 (a), where light incident on the face is occluded by the nose which casts a shadow onto the right cheek. Light incident from the distant environment may however not only be occluded

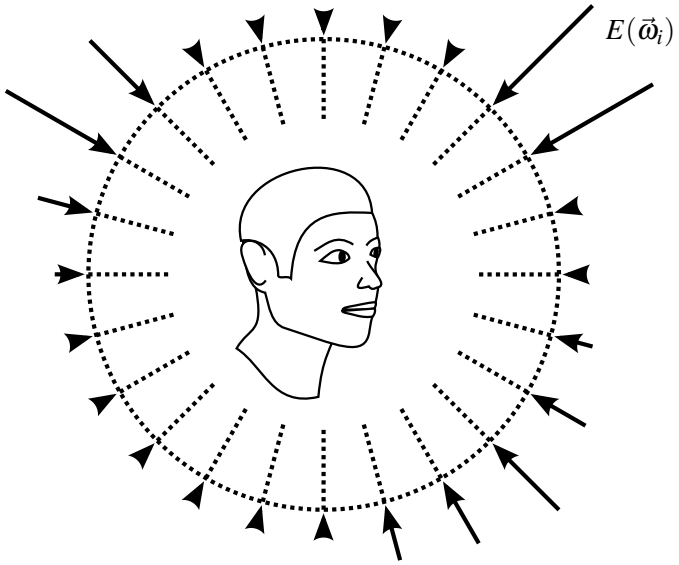


Figure 2.8: We model the light incident from the distant scene as a function $E(\vec{\omega}_i)$ defined over the range of directions.

by local geometry, but light from the distant environment incident on the local scene may also be reflected from one surface point towards another one.

As we already know from section 2.5.1, the overall *reflected* light at a surface point x into direction $\vec{\omega}$ can be specified according to the reflection equation (2.15).

Our separation of the scene into local and distant scene allows us to now rewrite this equation. We divide the domain of integration in equation (2.15), the upper hemisphere $\Omega(x)$, into two disjoint sets of directions. The first set $\Omega_E(x)$ contains the directions into which from surface point x the distant environment is visible – marked as green in figure 2.9 (b). The second set $\Omega_R(x)$ contains those directions into which the distant environment is occluded by the local scene – marked as red.

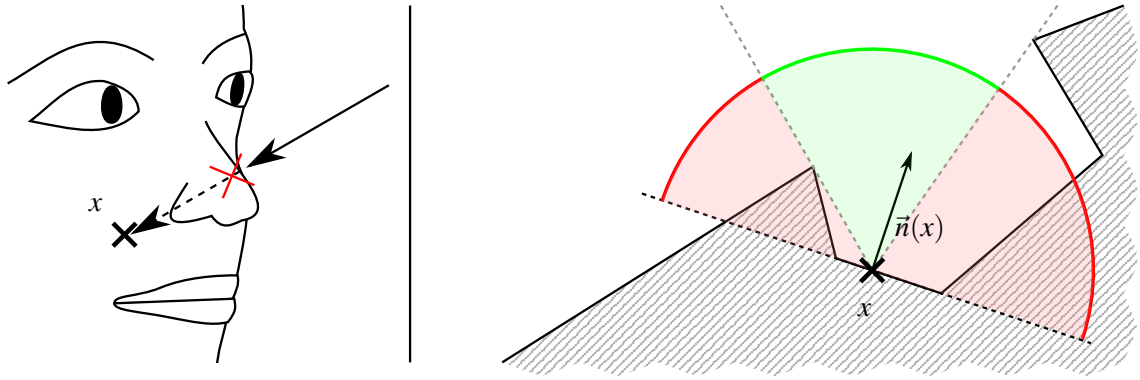
$$\Omega(x) = \Omega_E(x) \cup \Omega_R(x) \tag{2.18}$$

When sampling the incident light L_i at a surface point x in the reflection equation, we now can distinguish whether from direction $\vec{\omega}_i$ either light is coming directly from the distant scene or light is coming from the local scene itself.

We thus specify the incident light as:

$$L_i(x, \vec{\omega}_i) = \begin{cases} R(h(x, \vec{\omega}_i), -\vec{\omega}_i), & \text{if } \vec{\omega}_i \in \Omega_R(x) \\ E(\vec{\omega}_i), & \text{otherwise, i.e. if } \vec{\omega}_i \in \Omega_E(x) \end{cases} \tag{2.19}$$

For the overall *reflected* light R at the surface point x on the face into direction $\vec{\omega}$ we then have:



(a) (b)

Figure 2.9: At a surface position x, e.g. on the cheek, some of the incident light out of the distant environment may be blocked, e.g. by the nose (a). The upper hemisphere $\Omega(x)$ oriented along surface orientation $\vec{n}(x)$ at surface point x can be divided (b) into the set of directions $\Omega_E(x)$ (green) in which the distant environment is visible and the set of directions $\Omega_R(x)$ (red) in which the distant environment is occluded by local geometry.

$$\begin{aligned} R(x,\vec{\omega}) = &\int_{\Omega_E(x)} f_r(x,\vec{\omega}_i,\vec{\omega}) \cdot E(\vec{\omega}_i) \cdot (\vec{\omega}_i \cdot \vec{n}(x))\, \mathrm{d}\vec{\omega}_i \\ &+ \int_{\Omega_R(x)} f_r(x,\vec{\omega}_i,\vec{\omega}) \cdot R(h(x,\vec{\omega}_i), -\vec{\omega}_i) \cdot (\vec{\omega}_i \cdot \vec{n}(x))\, \mathrm{d}\vec{\omega}_i. \end{aligned} \tag{2.20}$$

An image captured of the face will contain partial measurements of R: measurements for surface points x that are visible for the camera with the direction $\vec{\omega}$ pointing towards the camera. Our goal is to find the most plausible directional distribution of incident light E for these measurements. For that we will in the following section have a closer look at the function modeling the correlation between E and R, which we refer to as *Radiance Transfer Function (RTF)*.

2.5.3 Radiance Transfer Function

By defining a linear operator **B** that represents the light transport by a *single* reflection step at a surface, we can simplify equation (2.20) for the light leaving the surface in the local scene to:

$$R = \mathbf{B}(E + R) \tag{2.21}$$

The equation still contains its recursive part, which we can rewrite as an infinite Neumann series:

$$\begin{aligned} R &= \mathbf{B}(E+R) = \mathbf{B}(E+\mathbf{B}(E+R)) = \cdots = \sum_{i=1}^{\infty} \mathbf{B}^i(E) \\ &= \mathbf{T}(E) \end{aligned} \tag{2.22}$$

The introduced new operator **T** includes *all* the light transport – from direct illumination of the local scene ($\mathbf{B}(E)$), up to an infinite number of interreflections ($\mathbf{B}^{\infty}(E)$) within the local scene. **T** maps the directional distribution of incident light E to radiance R leaving the surface of the local scene. Recovering the incident light would correspond to applying an inverse operator to R like $\mathbf{T}^{-1}(R) = E$.

Light incident from one direction We have modeled the incident light from the distant scene as a function E which depends on direction $\vec{\omega}_i$ and we have modeled the light leaving the local scene as a function R depending on location x and direction $\vec{\omega}$.

According to the linear nature of light (*rules of superposition* [Nime 95]) the radiance in the scene generated by multiple light sources is simply the sum of the radiance that each particular light source would cause individually. Let us thus start by considering that light would be incident out of the distant scene only from one particular direction $\vec{\omega}_i$ with intensity $E(\vec{\omega}_i)$, as depicted in figure 2.10. The light incident out of $\vec{\omega}_i$ with intensity $E(\vec{\omega}_i)$ distributes and propagates throughout the scene and creates a certain radiance in the scene, which we denote as $R_{\vec{\omega}_i}(x, \vec{\omega})$.

For every triple $(x, \vec{\omega}, \vec{\omega}_i)$ of location and directions we thus can calculate the ratio between $R_{\vec{\omega}_i}(x, \vec{\omega})$ and $E(\vec{\omega}_i)$. This gives us a non-negative real-valued function T which describes – analogically to the operator **T** – how radiance incident onto the local scene from the distant environment out of direction $\vec{\omega}_i$ is transferred into radiance leaving the local scene at position x into direction $\vec{\omega}$.

$$T(\vec{\omega}_i, x, \vec{\omega}) = \frac{R_{\vec{\omega}_i}(x, \vec{\omega})}{E(\vec{\omega}_i)} \tag{2.23}$$

On the one hand, this ratio depends on the surface orientation and the material properties at the surface location x itself, similar to the BRDF $f_r(x, \vec{\omega}_i, \vec{\omega})$ in equation (2.20). On the other hand – in contrast to that BRDF – this ratio however also incorporates the full light transport and not only a single reflection step. It thus also depends on the geometry and material properties of the whole local scene which can occlude light but also reflect light towards x.

Such kind of function has been already studied in the past. In accordance to terminology in precomputed radiance transfer in computer graphics (e.g. [Sloa 02]) we refer to that function as *Radiance Transfer Function (RTF)*. Debevec *et al.* [Debe 00] for example denote it as *reflectance function*. According to [Liu 04], an RTF tabulates the linear response of a surface point in terms of *exit radiance* (R) to *source lighting* (E).

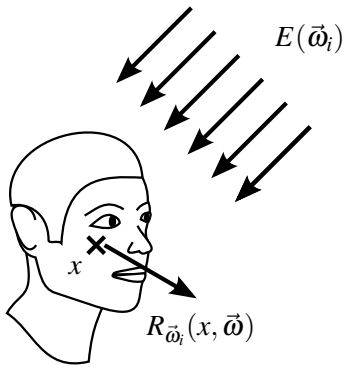


Figure 2.10: The light $E(\vec{\omega}_i)$ incident on the face from the distant scene out of direction $\vec{\omega}_i$ is transferred into radiance $R_{\vec{\omega}_i}(x,\vec{\omega})$ leaving the face at a point x in direction $\vec{\omega}$.

We can rearrange equation (2.23) to get an equation for $R_{\vec{\omega}_i}(x,\vec{\omega})$:

$$R_{\vec{\omega}_i}(x,\vec{\omega}) = T(\vec{\omega}_i,x,\vec{\omega}) \cdot E(\vec{\omega}_i) \tag{2.24}$$

Refraction and subsurface scattering effects In our previous mathematical derivations of light transport and RTFs, we deliberately assumed for simplicity that surfaces are opaque. We thereby neglected effects like refraction and subsurface scattering of the light and only considered reflection. While the involved math in sections 2.5.1 and 2.5.2 would become a little bit more verbose, the previous derivation of an RTF would still hold, when incorporating all these effects into the equations.

In the following we will employ captured images of *real* faces for measuring R, the light leaving the local scene. These measurements of R thus contain all the effects existing in the real world. As we will learn our RTF based on these measurements, we automatically account for all the effects like reflection, refraction, and subsurface scattering that occur on the face.

Figure 2.11 illustrates different examples of light paths that contribute to $R_{\vec{\omega}_i}(x,\vec{\omega})$. All these paths have in common that the light originally comes out of the distant scene from direction $\vec{\omega}_i$ and finally leaves the face at x into direction $\vec{\omega}$. The incident light from the distant environment may directly hit the surface point x and be reflected towards the camera (figure 2.11 (a)). The distant environment may however also be blocked at x in the direction $\vec{\omega}_i$ (figure 2.11 (b)). Additionally light that first hits another surface point may then be reflected to x either directly or after multiple reflections (figure 2.11 (c)). Likewise the incident light may be scattered to x below the skin after first hitting another surface point (figure 2.11 (d)). Obviously combinations of these effects contribute to $R_{\vec{\omega}_i}$ as well.

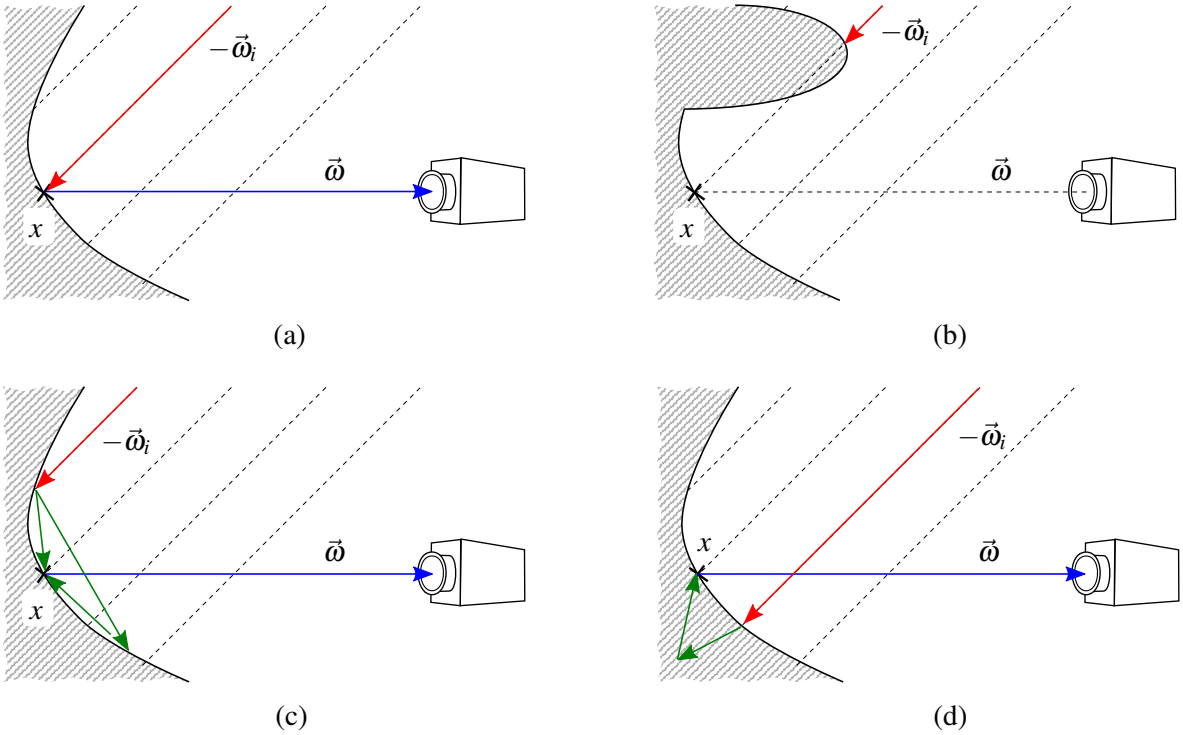


Figure 2.11: Possible light paths for light incident on the local scene out of direction $\vec{\omega}_i$ (red), which finally leaves at surface point x into the direction $\vec{\omega}$ towards the camera (blue). Beside direct reflection (a) and occlusion (b), the light in-between (green) potentially is reflected (c) or refracted and scattered (d) multiple times.

Light incident from multiple directions In the real environment, light usually does not come from only a single direction, but from all directions with varying intensities. The incident light then consists of a dense distribution of light intensities over the range of directions. Finally we want to solve for this distribution of incident light E, based on an image of a face. This image will however contain measurements of R, while our equation (2.24) at the moment is specified for $R_{\vec{\omega}_i}$.

Based on the linear nature of the light transport, the overall reflected light, resulting from light incident from the distant scene from multiple directions, is the *sum* over the reflected light intensities corresponding to each single incident light direction. For the continuous range of incident light directions the sum becomes an *integral* of the reflected light for incident light from the distant scene over all directions (specified as the unit sphere S^2). The integrand is the product of the RTF $T(\vec{\omega}_i, x, \vec{\omega})$ and the incoming light intensity $E(\vec{\omega}_i)$ from the distant scene, both evaluated for the particular direction $\vec{\omega}_i$.

$$R(x, \vec{\omega}) = \int_{S^2} T(\vec{\omega}_i, x, \vec{\omega}) \cdot E(\vec{\omega}_i) \, \mathrm{d}\vec{\omega}_i \tag{2.25}$$

How $R(x,\vec{\omega})$ relates to pixel intensity In our approach we employ real images of the human face for measuring the light R leaving from the local scene, i.e. the face. By capturing an image of the user's face, we take measurements of that part of R, that is leaving the face towards the camera. We use these measurements both for first learning the RTFs as well as for later estimating the illumination.

During image capturing, the radiance R coming from the local scene towards the camera first passes the lens system of the camera and is projected onto the sensor plane of the camera. The resulting irradiance incident on the sensor is proportional to radiance R for any particular position on the sensor, the factor however may vary between different positions. Debevec and Malik [Debe 97] argue that most modern camera lenses provide a nearly constant mapping factor over the sensor especially for smaller apertures. The camera response function of the particular camera electronics then determines how sensor irradiance is mapped to pixel intensities in the final image. For algorithms calculating with intensities of light, an important pre-processing step often is a radiometric calibration of the camera (see e.g. [Debe 97]), which determines the inverse mapping from pixel intensity to sensor irradiance and thus permits calculating with linear intensities.

Currently we however do not employ any kind of radiometric calibration, neither for offline learning nor for the online estimation. We however assume that images are encoded with a gamma correction of $\gamma = 1/2.2$, for which we compensate by decoding with $\gamma = 2.2$. Beside that, we assume an approximately linear mapping of the camera between sensor irradiance and pixel intensity. While this degrades the physical correctness of the estimation, it copes with the objective to run the algorithm out of the box on diverse consumer hardware.

Sparse Sampling We will not recover the full RTF $T(\vec{\omega}_i, x, \vec{\omega})$ but instead we will investigate this function at a discrete number of positions x and directions $\vec{\omega}$.

For the images of the faces that we use, we restrict ourselves to a fixed pose. Without loss of generality we pick the frontal head pose. Due to the fixed pose of the face in front of the camera a certain position $x = x_j$ on the face, for example on the right cheek, also implicates a fixed direction $\vec{\omega} = \vec{\omega}_j$ from x_j towards the camera in relation to the coordinate system of the face (see figure 2.12). The position x_j on the right cheek is projected along direction $\vec{\omega}_j$ onto a pixel position in the captured image. The brightness of the pixel at that position thus contains information about $R_j = R(x_j, \vec{\omega}_j)$, the radiance leaving at the surface point x_j into the direction $\vec{\omega}_j$, i.e. towards the camera.

Associated with x_j and $\vec{\omega}_j$ we define $T_j \colon S^2 \to \mathbb{R}_0^+$ with $T_j(\vec{\omega}_i) = T(x_j, \vec{\omega}_i, \vec{\omega}_j)$. This partial function of the RTF T describes the ratio of radiance leaving at position x_j into direction $\vec{\omega}_j$, i.e. towards the camera, to radiance incident out of the distant scene from $\vec{\omega}_i$. As j fixes x_j and $\vec{\omega}_j$, $T_j(\vec{\omega}_i)$ only depends on the direction of incident light from the distant environment. Note that we will again refer to T_j itself simply as a Radiance Transfer Function, although we now have a discrete set of these *partial* RTFs.

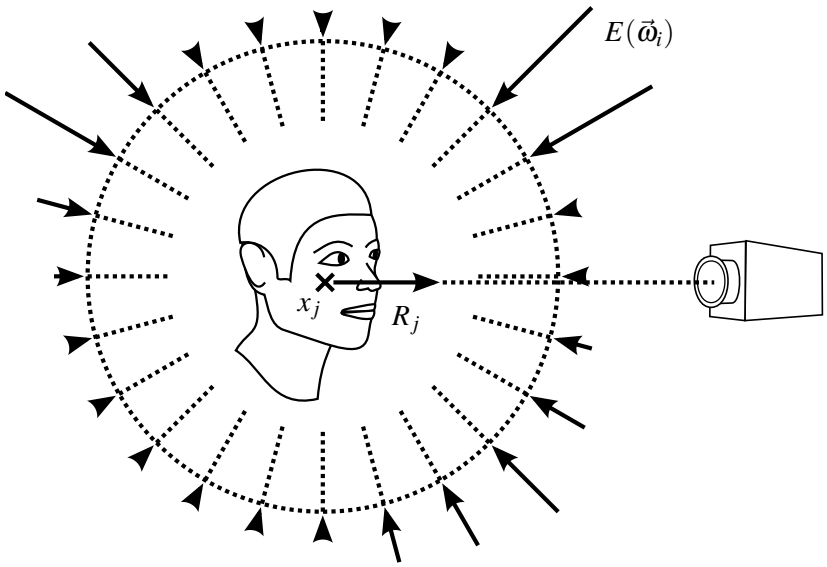


Figure 2.12: The light distribution E incident from the distant scene is transferred by the face into radiance R, where R_j is that part of R that is leaving at sample position x_j into direction $\vec{\omega}_j$ towards the camera.

Pinning $x = x_j$ and $\vec{\omega} = \vec{\omega}_j$ also reduces equation (2.25) to:

$$R_j = \int_{S^2} T_j(\vec{\omega}_i) \cdot E(\vec{\omega}_i)\, \mathrm{d}\vec{\omega}_i \tag{2.26}$$

Spherical Harmonics approximation Equation (2.26), which describes the radiance R_j leaving at x_j into direction ω_j, contains the two functions $E\colon S^2 \to \mathbb{R}_0^+$ and $T_j\colon S^2 \to \mathbb{R}_0^+$. Both map from the domain of unit directions to real (non-negative) numbers.

In order to cope with the learning of the RTFs and the estimation of the illumination, we reduce the dimensionality of the function space for E and T_j. We therefore model all RTFs T_j as well as the distribution of incident light E from the distant environment using a linear combination of real-valued Spherical Harmonics (SH) – orthonormal basis functions $Y_n(\vec{\omega})$ defined over the domain of directions (see section 2.4). We restrict the SH basis functions to maximum degree $L = 2$ resulting in nine SH coefficients describing a particular linear combination. The low maximum degree limits our function space to very smooth functions.

Let $\hat{T}_j \in \mathbb{R}^9$ be the SH coefficient vector that describes the RTF $T_j(\vec{\omega}_i)$ at location x_j. T_j then is approximated by:

$$T_j(\vec{\omega}_i) \approx \sum_{n=0}^{8} \hat{T}_{j,n} Y_n(\vec{\omega}_i) \tag{2.27}$$

Similarly, let $\hat{E} \in \mathbb{R}^9$ be the SH coefficient vector that describes $E(\vec{\omega}_i)$, a particular directional distribution of incident light. E then is approximated by:

$$E(\vec{\omega}_i) \approx \sum_{n=0}^{8} \hat{E}_n Y_n(\vec{\omega}_i) \tag{2.28}$$

Following equation (2.26) the reflected light R_j can be expressed as an integral of the product of RTF and particular distant illumination over all directions, where we now can plug-in our SH approximations.

$$\begin{aligned} R_j &= \int_{S^2} T_j(\vec{\omega}_i) \cdot E(\vec{\omega}_i) \, \mathrm{d}\vec{\omega}_i \\ &\approx \int_{S^2} \sum_{n=0}^{8} \hat{T}_{j,n} Y_n(\vec{\omega}_i) \cdot \sum_{n=0}^{8} \hat{E}_n Y_n(\vec{\omega}_i) \, \mathrm{d}\vec{\omega}_i \end{aligned} \tag{2.29}$$

With both the RTF as well as the distant illumination expressed in SHs, we can exploit the orthonormal properties (see equation (2.8)) of the SH basis functions. The integral of the product of RTF and incident illumination over all directions then becomes a simple dot product of the SH coefficient vectors $\hat{T}_j$ and $\hat{E}$ and thus can be evaluated much more efficiently.

$$R_j \approx \hat{T}_j^\top \cdot \hat{E} \tag{2.30}$$

This equation (2.30) is fundamental for all the following calculations, from learning the RTFs (section 2.5.4) to estimating the illumination (section 2.5.5).

In the following we loosely write $=$ instead of $\approx$ also when we refer to SH approximations.

2.5.4 Offline Learning of the Impact of Light on the Appearance of Faces

In this section we elaborate our training procedure to determine the Radiance Transfer Function (RTF) T_j for a particular position x_j on the human face. This offline learning process for an RTF only has to run once in advance.

We refer to a position x_j as sample position. Later we will use a whole set of sample positions distributed over the face. In this phase we however look at each particular sample position x_j separately and for each location x_j learn its own RTF T_j.

As already indicated we model our RTFs $T_j(\vec{\omega}_i)$ using Spherical Harmonics (SH) basis approximations, so that a specific function is described by the SH coefficient vector $\hat{T}_j \in \mathbb{R}^9$. With regard to this representation, learning an RTF T_j means determining the nine SH coefficients of $\hat{T}_j$.

We later want to use the learned RTFs to estimate the illumination from images of arbitrary human faces, without the need to do a separate learning step for every new person. Therefore we want to determine for a particular sample position x_j on the human face the *average* RTF $T_j(\vec{\omega}_i)$ that best approximates all the various RTFs from different faces at position x_j .

2.5.4.1 Input Training Data

In the offline learning stage, we learn how illumination impacts the appearance of a face. For that purpose we employ images from *The Extended Yale Face Database B* [Geor 01, Lee 05], which is publicly available. This database contains a set of grayscale images of faces with *frontal head pose* from 38 human subjects each under 64 different illumination conditions. We additionally include horizontally flipped versions of the images, which overall gives us 4864 images.

Our goal is to determine the RTF T_j for a sample position x_j based on the intensities of the pixels where x_j projects to in the images of different persons under different known directional illuminations.

Coordinate system and sample positions We loosely use the term x_j for both the position on a face in 3-dimensional space, as well as for the 2-dimensional projection x_j on the image of the face.

In order to specify the sample positions on an image of a face, we define a 2-dimensional coordinate system based on the two pixel positions of the eyes. For that we manually labeled the positions of the eyes for each person in the training images. The origin of the 2-dimensional coordinate system is set to the center between the two eye positions, the unit vector in x-direction pointing to the image-wise right eye, and the unit vector in y-direction pointing perpendicular in upwards direction.

Relative to this coordinate system, we specify a set of sparse sample positions x_j. We try to select the set so that positions are uniformly distributed over regions of the face which most likely correspond to skin e.g. cheeks, forehead, and nose. Details on the sample selection procedure are provided in

section 2.6.2.2. Let J be the set of selected samples with $j \in J$ specifying a particular sample with position x_j. The same set J of sample positions is used for all images.

Known directional illuminations The images of *The Extended Yale Face Database B* [Geor 01, Lee 05] are each taken under light from one particular direction only. This direction is specified in the database by azimuth and elevation angle, so that we know the illumination condition for each of the images.

The dataset thereby comprises light incident out of 64 different directions, which are marked in figure 2.13. The light directions range from frontal illumination, to illumination from the side and even partially from behind the face with a maximum longitude of 120°. Additionally light directions vary with latitude, some light is incident from below as well as some from above.

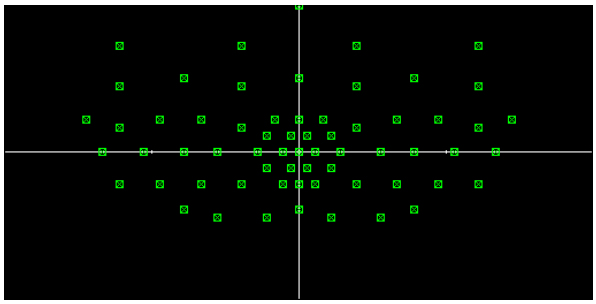


Figure 2.13: The employed dataset comprises 64 different directions of incident light which are marked in green in this Lat-Long image.

Let K be the set of different directional illuminations with $k \in K$ specifying a particular distant illumination E_k, that contains only directional light incident from direction $\vec{\omega}_k$. As a directional light only contains intensities for a single specific direction, it can be modeled as a Dirac delta function $\delta(\cdot)$, also known as Impulse symbol [Brac 99]. The directional distribution E_k then can be written as:

$$E_k(\vec{\omega}_i) = \delta(\vec{\omega}_i - \vec{\omega}_k) \tag{2.31}$$

This function equals to 0 for all directions $\vec{\omega}_i$ except for $\vec{\omega}_i = \vec{\omega}_k$. Integrating over this function results in 1. Being unaware of the real physical intensities, we assume some unit intensity for the light sources from the database.

We want to approximate E_k using our SH approximation. To determine the SH coefficient vector $\hat{E}_k$ we project E_k onto the SH basis functions (see equation (2.9)). Due to the properties of the Dirac delta function the integral in equation (2.9) to determine the coefficients becomes a simple evaluation of each basis function for $\vec{\omega}_k$.

$$\hat{E}_{k,n} = \int_{S^2} E_k(\vec{\omega}_i) \cdot Y_n(\vec{\omega}_i)\, \mathrm{d}\vec{\omega}_i = Y_n(\vec{\omega}_k) \tag{2.32}$$

Note that, albeit a directional light is locally defined in angular space, it contains all frequencies when defined in angular frequency space. An accurate representation by an SH expansion would need degree $L = \infty$. Our limitation to $L = 2$ hence involves a coarse approximation.

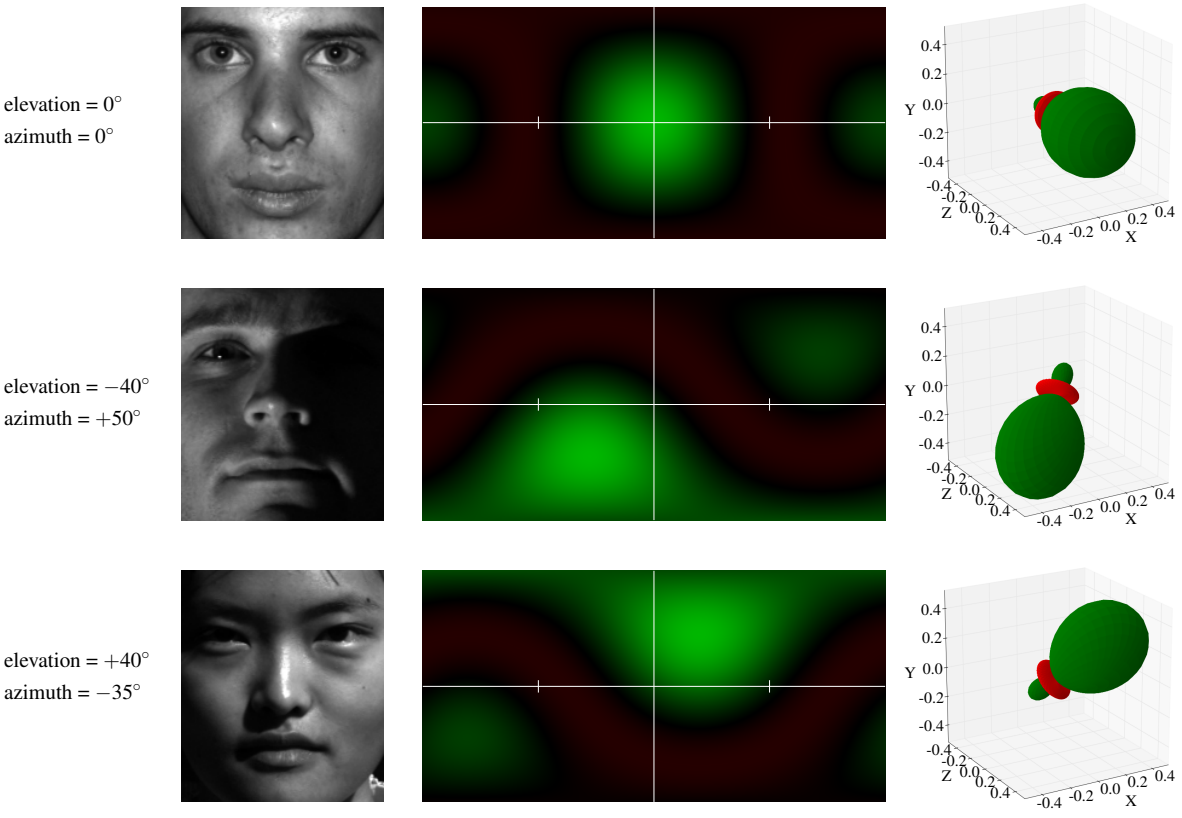


Figure 2.14: Each row illustrates one illumination condition: column (a) and (b) contain light specifications and images from *The Extended Yale Face Database B* [Geor 01, Lee 05], column (c) and (d) illustrate the corresponding SH approximation of the directional lights.

Figure 2.14 shows three images from the used dataset with corresponding directional illuminations, as well as our SH approximation for the directional light. The errors introduced by the SH approximation become apparent. While the original illumination only contains a peak in intensity for the particular direction $\vec{\omega}_k$, the Lat-Long illustrations (figure 2.14 (c)) of the SH approximations show a fair amount of blur around $\vec{\omega}_k$. The 3-dimensional plots in column (d) of figure 2.14 additionally show two other artifacts introduced by the approximation: some amount of intensity out of the opposite direction of $\vec{\omega}_k$, as well as small negative light intensities perpendicular to $\vec{\omega}_k$ – plotted in red.

2.5.4.2 Per Person Albedo Factor

We target at finding the RTF T_j for a sample position x_j that best approximates the radiance transfer for all different people. Different people however have different skin colors and thus obviously different RTFs. We make the assumption that the RTF for a particular sample position x_j mainly varies between different persons by a uniform scale. The scale can be considered as a per person albedo term that corresponds to the difference in the BRDF of the persons' skin. As skin reflectance is not fully diffuse, but also exhibits gloss, this assumption is not fully valid.

We want to compensate for the different skin colors before learning the average RTFs. From equation (2.30) we deduce that a scale in the RTF goes along with a scale in R, the radiance leaving the face, which we extract from the pixel intensities in the images. Let F be the set of the different faces in the dataset with $f \in F$ specifying a particular face. Before determining the average RTF for multiple people, we thus first *normalize* the pixel intensities of all training images of a face f by dividing by the albedo a_f of a respective face. We determine the albedo by the median over the intensities of all sample points in the frontal lit image of the particular face. For all these calculations we consider a gamma encoding of $\gamma = 1/2.2$.

After compensating for the per person albedo factor a_f, we assume that for a particular position x_j in the human face a single RTF T_j can be used to approximate the RTFs for all different persons.

The Extended Yale Face Database B [Geor 01, Lee 05] that we use for learning the RTFs only contains grayscale images. For estimating colored light, we will later assume in the online light estimation (section 2.5.5) that we can also reuse the RTF, that we learned from grayscale images, for different wavelengths of light by simply scaling the RTF by an albedo factor specific to the particular wavelength.

2.5.4.3 Setting up the System of Equations

As already indicated above, we examine – in the offline learning stage – each sample position x_j and its corresponding RTF T_j in separation of the other positions and RTFs.

Let p specify a particular picture from the dataset. The captured face in picture p then shall be denoted as f_p, as well as the distant illumination corresponding to picture p as E_p.

Let $I_{j,p}$ furthermore be the intensity of the pixel corresponding to sample position x_j in image p. The pixel intensity is related to the intensity of the reflected light $R_{j,p}$ (at surface point x_j into direction $\vec{\omega}_j$ by face f_p under illumination E_p). According to section 2.5.4.2 we account for the albedo a_{f_p} of a person f_p when deducing R from pixel intensities I in image p:

$$R_{j,p} = {I_{j,p}}^{(\gamma)} \cdot \frac{1}{a_{f_p}} \tag{2.33}$$

We rearrange equation (2.30) and for each image p form an equation $\hat{E}^{\top} \cdot \hat{T}_j = R_j$ between the *known* illumination E_p, the *unknown* RTF T_j at position x_j in the face , and the corresponding measured *known* reflected light intensity $R_{j,p}$.

We thus can use the set of $|K| \cdot |F|$ images to build a system of equations (2.34) for a particular sample position x_j where each image contributes one row.

$$\begin{pmatrix} \hat{E}^{\top}_{p=1} \\ \hat{E}^{\top}_{p=2} \\ \vdots \\ \hat{E}^{\top}_{p=|K|\cdot|F|} \end{pmatrix} \cdot \hat{T}_j = \begin{pmatrix} R_{j,(p=1)} \\ R_{j,(p=2)} \\ \vdots \\ R_{j,(p=|K|\cdot|F|)} \end{pmatrix} \tag{2.34}$$

Given on the left in this system of equations is the matrix of dimension $|K| \cdot |F| \times 9$ where each row contains the 9 coefficients specifying an illumination. Given on the right in this system of equations is the $|K| \cdot |F|$-dimensional column vector where each row contains the intensity of the reflected light (compensated by albedo).

Only the 9-dimensional vector $\hat{T}_j$ on the left side is unknown. Containing nearly[2] $|K| \cdot |F| = 4864$ equations, this system of equations is clearly overdetermined. The coefficients $\hat{T}_j$ thus can be calculated as the least squares solution of equation (2.34), which gives us our estimation for the average RTF for the particular location x_j.

We repeat this learning procedure for each x_j from the set of selected sample positions. Each time we build up this system of equations and estimate $\hat{T}_j$, so that in the end of the offline stage we have learned an average RTF for each sample position.

Figure 2.15 illustrates the learned average RTFs for six different sample positions on the face. Like before, the green intensity in the Lat-Long images corresponds to a positive value of the function for that direction. As you can see, sample position 2 on the left cheek responds to a greater extent to light coming from the left direction than to light coming from right, while for sample position 5 on the right cheek it is the other way round. Sample position 1 in figure 2.15 lies quite close to the nose, so that especially light coming from the right is occluded. The fact that the sample position is barely affected by light from the right is reflected by low intensities of the corresponding RTF for those directions.

2.5.4.4 Limitations

The assumptions and restrictions that we chose until here will enable a well performing light estimation that allows for coherent illumination of virtual objects in Augmented Reality. Still they will however also introduce some kind of imprecision, which we think is important to mention.

[2]Some images of the dataset are defective and have been removed.

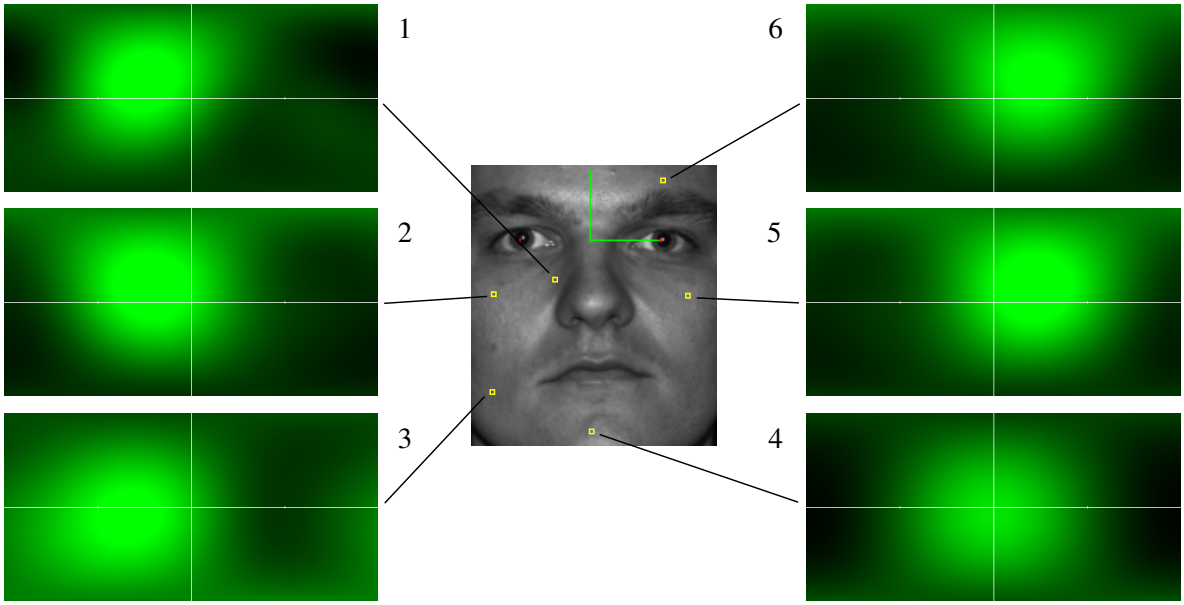


Figure 2.15: Six sample positions distributed over the area of the face with their corresponding RTFs that have been recovered in the offline learning procedure.

We already saw above, that the low degree we chose for the Spherical Harmonics approximation, introduced some degree of unsharpness for the ground truth illumination. The same unsharpness also becomes apparent in the recovered RTFs. These RTFs are also approximated by Spherical Harmonics of maximum degree $L = 2$, so that they hardly can model sharp edges from occlusions or specular reflections. While e.g. sample position 1 of figure 2.15 already reproduces the occlusion effect of light by the nose to some extent, it only does so in a very smoothed out version, which does not fully cope with the harsh occlusion and shadow edge.

Also adapting to different skin colors by simply scaling intensity values and RTFs respectively by a single albedo value is a coarse approximation that is not completely valid for objects which are not both fully convex and fully diffuse. Human skin exhibits a significant amount of glossy reflection and subsurface scattering as well as the geometry of a human face contains concavities, too. We may want to improve this approximation in future work.

Another potential shortcoming at that point are the neglected differences in head shape between different people. The way we currently define the coordinate system for the sample positions is simple. In a future implementation, a more elaborate way to specify the coordinate system could improve the accuracy in registration over multiple persons, and by that would make averaging over multiple persons more accurate. Options range from simple improvements like including the position of the nose to define the scale in y-direction, to more sophisticated approaches where sample positions could be defined in an elastic coordinate system that adapts to the head shape of a particular person e.g. by exploiting the facial fiducials all over the face instead of only the position of the eyes and nose.

2.5.5 Online Illumination Estimation

This section describes the online light estimation stage, where we want to estimate in real time the unknown directional distribution of incident light $E(\vec{\omega}_i)$ for a particular single image showing the face of the user. For this purpose we will apply the set of Radiance Transfer Functions (RTFs) T_j, that we learned beforehand in the offline learning stage (see section 2.5.4).

2.5.5.1 Face Tracking

The input for the online light estimation stage is an image of the user's face from a live video stream captured by a user-facing camera. We align the sample positions from the offline process to the image of the face by means of face tracking.

We use an image-based face tracking prototype as a black box. For an input image of a human face we obtain a pose with six degrees of freedom (6DoF pose) comprising 3-dimensional translation and 3-dimensional rotation. Note that in our prototype implementation we beforehand projected the 2-dimensional sample positions from the offline learning stage onto a 3-dimensional face model for the sake of simplicity, which results in 3-dimensional sample positions registered to the human face. We now use the 6DoF pose from the face tracker to project the sample positions x_j defined on the 3-dimensional face model back onto pixel positions of the captured camera image.

Figure 2.16 shows a set of projected sample positions during live tracking. For now, our light estimation algorithm assumes that the head pose is close to frontal in order to work properly.

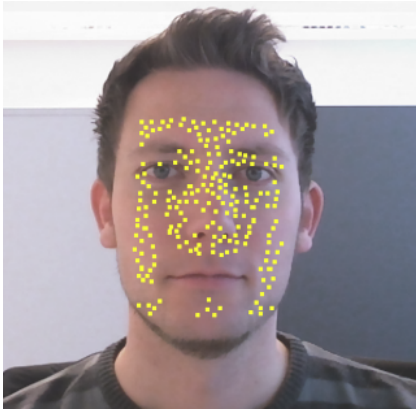


Figure 2.16: Visualized sample positions on the face during live tracking.

2.5.5.2 Setting up the System of Equations

Now that we know the pixel positions of the sample positions in the current image of the face, we can extract the intensity values of the pixels. To roughly linearize the mapping between pixel intensities

and radiance, we again assume a gamma encoding by the camera of $\gamma = 1/2.2$, for which we compensate by decoding with $\gamma = 2.2$. Like in the offline learning stage this is only a very coarse approximation of the real camera response curve.

This time we work on a single image. Let I_j now be the intensity of the pixel corresponding to sample position x_j in the current image.

Once again we rearrange equation (2.30) and form an equation for every sample position x_j in the current image:

$$\hat{T}_j^\top \cdot \hat{E} = R_j \tag{2.35}$$

$$= I_j^{(\gamma)}/a \tag{2.36}$$

This time, the RTFs $\hat{T}_j$ for the different sample positions are known from the offline estimation step, but $\hat{E}$ is unknown. R_j, the intensity of reflected light (at surface point x_j into direction $\vec{\omega}_j$ towards the camera), is provided by the intensity I_j of the pixel corresponding to sample position x_j. The albedo of the specific user is taken into account by factor a. This factor a at the moment is manually defined. In contrast to the offline learning stage, where we use the median intensity from a frontal lit image of the person, we at that point do not have a frontal lit image for the specific user available. In the offline stage we also assumed a normalized intensity of the illumination, this time we want to be able to also distinguish between dimmer and brighter illuminations.

Using the set J of sample positions we can again set up a system of equations similar to the previous one (system of equations (2.34)). Note that while in the offline stage we learned the RTF for a particular sample position independent of the other sample positions using a whole set of images, this time we use only one image but all sample positions at once. Each of the sample positions thereby contributes one row to the system of equations.

$$\begin{pmatrix} \hat{T}_{j=1}^\top \\ \hat{T}_{j=2}^\top \\ \vdots \\ \hat{T}_{j=|J|}^\top \end{pmatrix} \cdot \hat{E} = \begin{pmatrix} R_{j=1} \\ R_{j=2} \\ \vdots \\ R_{j=|J|} \end{pmatrix} \tag{2.37}$$

This gives us a $|J|$-dimensional vector on the right side for the reflected light intensities, with $|J|$ being the number of employed sample positions. On the very left side of the system of equations, this time the different RTFs of the sample positions build the rows of a matrix of dimension $|J| \times 9$. Unknown are the 9 SH coefficients of $\hat{E}$ specifying the directional distribution of incident light E.

By again solving this overdetermined system of equations by least squares we obtain our estimation for the illumination. We present visual results of this simple unconstrained least-squares solution in

section 2.6.1 as well as quantitative evaluations against ground truth in section 2.6.2. We there also examine the performance of our method for different numbers of sample positions.

In the next section 2.5.6 we will identify and address weaknesses of our so far presented method.

Estimating Colored Illumination Our training set from *The Extended Yale Face Database B* [Geor 01, Lee 05] only contains grayscale images. When we are only interested in a grayscale illumination, we simply convert the pixel intensities of the input image in the online estimation stage from color to grayscale.

For estimating colored illumination we work on each color channel of the input image individually. We make the assumption that the RTF for a particular light frequency can be approximated by just scaling the grayscale RTF from the offline stage by an albedo factor specific to the particular user and frequency, i.e. color channel. The three albedo factors $a_{\{r,g,b\}}$ for the red, green and blue channel thereby correspond to the components of the albedo color of the person.

In our prototype implementation we at the moment manually specify the albedo factors $a_{\{r,g,b\}}$ of the user. We support this by allowing to pick a color from the current video frame under the assumption that the face at this moment is lit under a white illumination of unit intensity. Future approaches could e.g. work with active lighting using flash to once automatically determine the user's albedo factors or employ a method that estimates the illuminant color based on highlights [Klin 88, Stor 00].

We make three separate light estimations, one on each color channel of the input image. We thus receive three separate SH coefficient vectors $\hat{E}_{\{r,g,b\}}$ of the directional distribution of incident light. According to equation (2.37) it is equivalent whether we thereby scale all the $|J|$ RTFs $\hat{T}_j$ by the albedo factor for the respective color channel or simply inversely scale the corresponding coefficients of the estimated colored illumination $\hat{E}_{\{r,g,b\}}$.

Visual results of our estimation of colored illumination from the face of the user are depicted in figure 2.27 in the evaluation part (section 2.6.1).

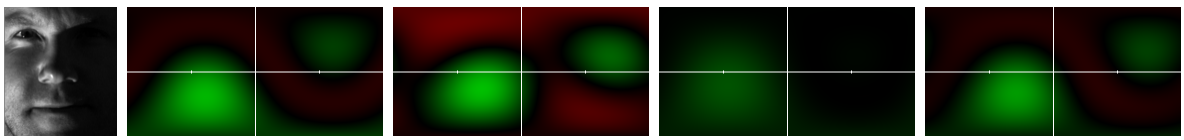
2.5.6 Improving the Online Light Estimation

The online light estimation part as presented in section 2.5.5 corresponds mainly to the original approach that we introduced in [Knor 14]. This approach already provides plausible visual results when the estimated illumination is applied to virtual content for coherent rendering – see section 2.6.1. However some limitations have been noted. In the following we will first address the problem of the algorithm tending to also estimate negative light intensities (section 2.5.6.1). We then will explain how measurements from multiple frames can be combined for a more reliable estimation (section 2.5.6.2). Parts of the averaged RTFs learned from the dataset may misfit for a particular person, e.g. because of a moustache or a macula, which falsifies the results of the light estimation. In section 2.5.6.3 we approach this problem proposing an outlier detection and removal procedure.

2.5.6.1 Constraining the Solution

The system of equations (2.37), as presented above and originally proposed in [Knor 14], is solved via unconstrained least-squares minimization. As there are no constraints, the solution space also contains SH coefficient vectors $\hat{E}$, which correspond to directional distributions of incident light that contain negative light intensities for some parts of the directions. Whereas a certain amount of negative intensities may arise from the low dimensional approximation of light sources by SHs, the least-squares solver in here finds a solution that uses negative lighting in combination with over-estimated positive lighting to reproduce harsh variations in intensities. The issue is supported by under-determined knowledge about light intensities coming from behind the user, which leaves some freedom for the light estimation solver.

Figure 2.17 illustrates this problem. On the left of figure 2.17 we have the input image (a) with its corresponding SH approximation of the ground truth illumination (b). This illumination contains mainly positive intensities. The estimated illumination in terms of the unconstrained least-squares solution (c) based on the system of equations (2.37) resembles the ground truth illumination in terms of the primary light direction, it however clearly contains over-estimated negative as well as over-estimated positive intensities.



(a) (b) (c) (d) (e)

Figure 2.17: Side-by-side comparison of the illumination estimated by the unconstrained solver (c) as well as the new constrained solver with $\varepsilon = 0$ (d) and $\varepsilon = -0.14$ (d) for the image of a face (a) with ground truth illumination (b) projected to SHs.

Negative light intensities especially cause problems when we e.g. afterwards apply the illumination to surface regions of virtual objects that predominantly receive light with negative intensities, as well as when we want to compute shadow cast from a virtual object onto the face. The overestimated light intensities cause trouble when we extract conventional lights like the primary light directions. The errors in the estimation also become problematic when we want to rotate and combine multiple measurements, as well as when we estimate colored illumination on separate channels.

In the real world, there are no negative light intensities, so a natural approach is to try to restrict the space of solutions to physically plausible ones, i.e. only positive light intensities. Ideally we thus would like to only allow solutions for the system of equations (2.37), where $E(\vec{\omega}_i) >= 0 \quad \forall \quad \vec{\omega}_i \in S^2$.

The incident light E in our approach is modeled as a SH approximation described by the SH coefficient vector $\hat{E} \in \mathbb{R}^9$. To restrict the illumination E to positive intensities, we cannot directly apply bound (box) constraints on the components of $\hat{E}$ individually. The intensity of light that is incident from a particular direction $\vec{\omega}_m$ is evaluated in our model by a linear combination of the SH basis functions Y_0 to Y_8 each evaluated for that direction $\vec{\omega}_m$ and weighted by the coefficients of $\hat{E}$:

$$E(\vec{\omega}_m) \approx \sum_{n=0}^{8} \hat{E}_n Y_n(\vec{\omega}_m) \tag{2.38}$$

Constraining the solution space $\hat{E} \in \mathbb{R}^9$ in terms of a constraint on $E(\vec{\omega}_m)$ thus is possible by linear constraints modeled by SH basis functions Y_0 to Y_8 evaluated for $\vec{\omega}_m$.

We reformulate the objective function from the least-squares minimization of equation (2.37) as a quadratic programming problem. Let $\boldsymbol{T} \in \mathbb{R}^{|J|\times 9}$ be the matrix on the left of equation (2.37) containing all the stacked RTF coefficients and $\boldsymbol{i} \in \mathbb{R}^{|J|}$ be the vector on the right of equation (2.37) containing all the stacked intensities. When can rewrite the objective function as:

$$\begin{aligned} \frac{1}{2}\left\|\boldsymbol{T}\hat{E} - \boldsymbol{i}\right\|^2 &= \frac{1}{2}(\boldsymbol{T}\hat{E} - \boldsymbol{i})^\top(\boldsymbol{T}\hat{E} - \boldsymbol{i}) \\ &= \frac{1}{2}(\hat{E}^\top \boldsymbol{T}^\top \boldsymbol{T}\hat{E} - 2\hat{E}^\top \boldsymbol{T}\boldsymbol{i} + \boldsymbol{i}^\top \boldsymbol{i}) \end{aligned} \tag{2.39}$$

We now can estimate the illumination as a quadratic programming problem with linear constraints modeled by a matrix $\boldsymbol{A}$:

$$\begin{aligned} \underset{\hat{E}\in\mathbb{R}^9}{\text{minimize}} \quad & \frac{1}{2}\hat{E}^\top \boldsymbol{Q}\hat{E} - c^\top \hat{E} \\ \text{subject to} \quad & \boldsymbol{A}\hat{E} \geq \boldsymbol{\varepsilon} \end{aligned} \tag{2.40}$$

where $\boldsymbol{Q} = \boldsymbol{T}^\top \boldsymbol{T}$, $c^\top = \boldsymbol{i}\boldsymbol{T}$, and $\boldsymbol{A} \in \mathbb{R}^{M\times 9}$.

Equation (2.40) allows us to define a lower bound[3] of ε for intensities of the estimated illumination $\hat{E}$ for a finite number of directions $\vec{\omega}_m$ over S^2. This constraint is modeled by setting $\boldsymbol{A}_{m,n} = Y_n(\vec{\omega}_m)$. Each row m of $\boldsymbol{A}$ then contributes a constraint for one direction $\vec{\omega}_m$:

$$E(\vec{\omega}_m) \approx \sum_{n=0}^{8} \hat{E}_n Y_n(\vec{\omega}_m) \geq \varepsilon \tag{2.41}$$

In our current implementation we use $M = 100$ uniformly distributed directions. We solve for the illumination $\hat{E}$ employing the Goldfarb-Idnani active-set dual method [Gold 83].

An intuitive approach would be setting $\varepsilon = 0$ to only allow physically plausible positive light intensities. Estimating the real-world illumination restricted by $\varepsilon = 0$ works well in an environment where the light distribution is quite smooth and light is coming from everywhere. For illuminations with a very dominant light direction like in the dataset which is also used for the ground truth evaluation, the method however fails to well represent the dynamics in the illumination like depicted in figure 2.17 (d). Restricting solutions to only positive intensities in this case results in a flattened solution. This shows, that due to the low dimensional approximation some amount of negativity is needed for SHs to efficiently model the illumination.

We thus modify the constraints in (2.40) to $\varepsilon = -0.14$. This number was chosen based on the maximum negative intensity arising by projecting a directional light source of *unit* intensity into SHs. In future work, we want to investigate determining this constraint value on-the-fly based on the estimated solution. The effectiveness of the constrained solver with the new lower bound is illustrated in figure 2.17 (e), showing that the estimated illumination now features a high similarity to the ground truth illumination. Figure 2.18 additionally provides a visual comparison between the unconstrained and constrained estimation in a live sequence and demonstrates that negative intensities are clearly mitigated.

In section 2.6 we give quantitative results for the light estimation, compare the unconstrained solution to the constrained solutions with $\varepsilon = 0$ as well as $\varepsilon = -0.14$ and show that the latter one strongly outperforms the two others.

2.5.6.2 Combining Multiple Measurements

Our light estimation method only requires a *single* image to estimate the incident light which allows the estimation to always be up-to-date for a sequence of frames even during rapid changes in illumination. Sometimes it is however beneficial to combine the measurements from multiple images.

[3] Precisely $\varepsilon \in \mathbb{R}^M$, but we at the moment use the same lower bound value for all directions.

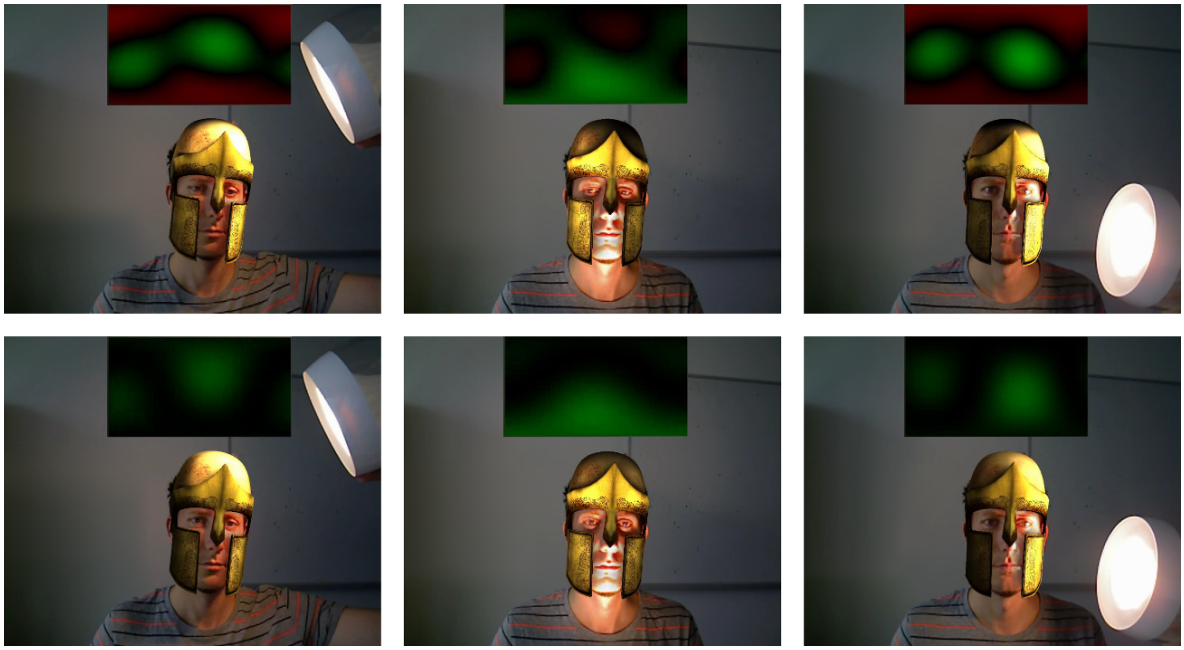


Figure 2.18: Comparison between the unconstrained solution (top row) and the constrained solution with $\varepsilon = -0.14$ (bottom row).

Temporal smoothing Albeit in many cases the estimation from a single image already provides stable results when running on a sequence of images, in some scenarios the estimated illumination changes a little bit from frame to frame which is noticeable in a flickering illumination of the augmentations. A simple temporal smoothing over the estimations from consecutive frames eliminates this problem without introducing any noticeable delay, e.g. by simply taking the average vector over the estimated SH coefficient vectors $\hat{E}$ of the last 4 frames.

Combining different orientations We deduce information about the illumination from the image of a face. A single image however contains only roughly one half of all possible surface orientations – those that are facing the camera. Ramamoorthi [Rama 02] has analyzed this fact and demonstrated that the variation within a single image of a convex diffuse object under arbitrary illumination can be even modeled by only 5 basis functions. He showed that orthogonality of the SH basis functions is no longer given for the restricted domain of visible surface orientations in one image. We investigate how far cast shadow from concavities and non-diffuse reflectance in faces as well as multiple images with different orientation in the world reduce this phenomenon.

For that purpose we analyze the correlation between the coefficients of the different SH basis functions over the set of learned RTFs. We use a particular set of 758 sample positions which we refer to as *N758*. Details on this set of sample positions can be found in section 2.6.2.2. Figure 2.19 illustrates the learned SH coefficients of the RTFs at the different sample positions of *N758*. Each image shows the coefficients of one particular SH basis function.

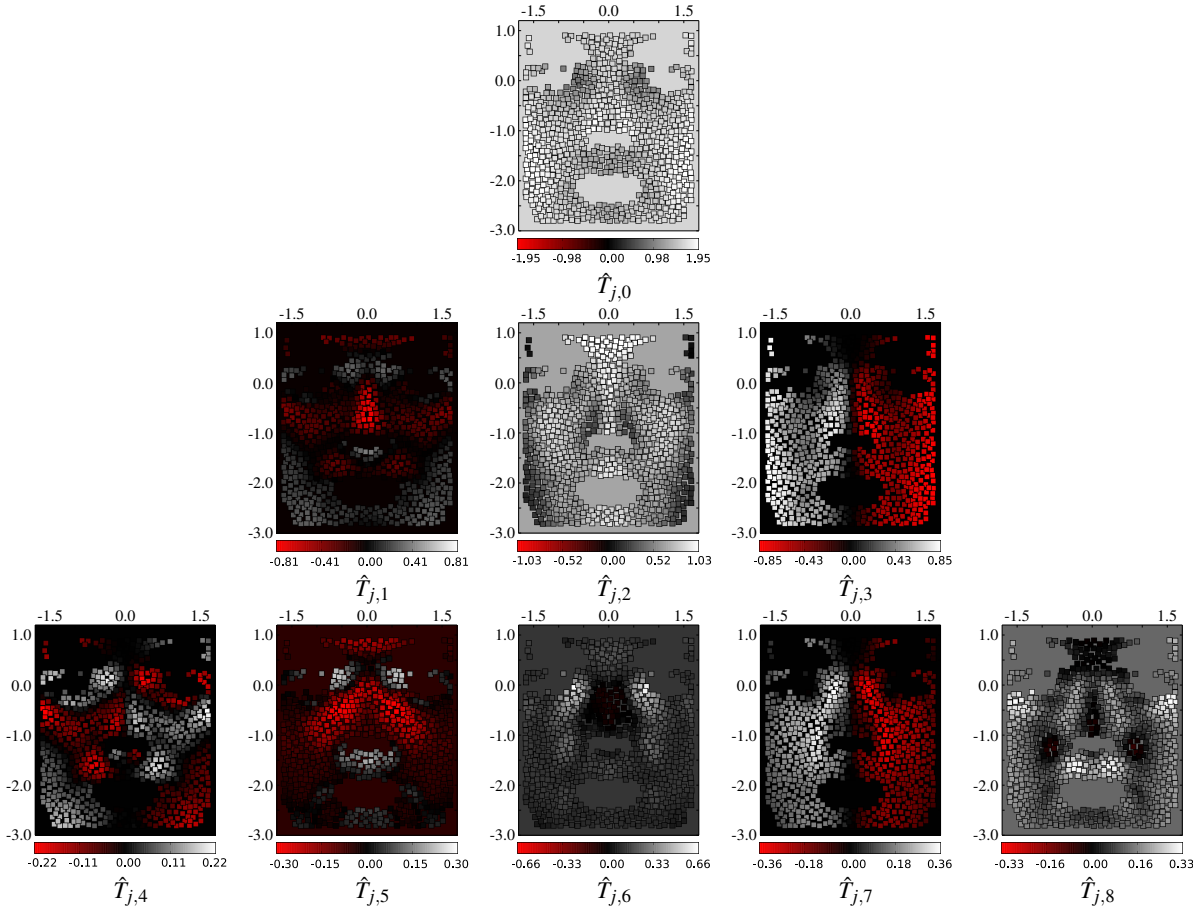


Figure 2.19: For each sample position x_j we learn a SH coefficient vector $\hat{T}_j = (\hat{T}_{j,0}, \hat{T}_{j,1}, \cdots, \hat{T}_{j,8})^\top \in \mathbb{R}^9$ representing the RTF at that position. The resulting coefficients of the particular SH basis functions are illustrated in pyramidal SH structure (in accordance with table 2.1).

From the set of learned RTFs we build the correlation matrix, i.e. matrix of Pearson product-moment correlation coefficients [Pear 95]. Each coefficient measures the linear dependence between two variables, in our case between two SH basis functions. A value of 0 corresponds to no linear dependence, while a value of 1 corresponds to a total positive linear correlation. A high correlation between two functions indicates in our case, that the two SH basis functions that model the illumination have a similar impact onto the appearance of the face under illumination, which in reverse makes it hard to attribute the appearance of the face to exactly one of the two functions.

The resulting correlation matrix for the set of learned RTFs is displayed in figure 2.20 (a). We can see a non negligible amount of ambiguity especially between Y_3 and Y_7, with a coefficient of 0.88. This similarity is also directly visible from the visualization of the RTFs in figure 2.19. From a *single* image it is thus hard to distinguish how the energy in illumination is distributed between Y_3 and Y_7. Y_3 corresponds to a simple linear variation in light intensities from left to right, while Y_7 contains an additional variation from front to back – see the SH basis functions in figure 2.5.

If the user however is turning around in the world while keeping the camera in front of their face, the multiple *frontal* face images at different orientations in the real world contribute different parts of the information about the illumination. Hence combining these measurements allows creating a larger set of observations and may tackle those ambiguities.

In order to combine the observations at different orientations into one system of equations, we need to rotate the once learned RTFs of the sample positions according to the current orientation of the face in the real world. The rotation of the RTFs can be performed by rotating the SH coefficient vectors $\hat{T}_j$ as presented in [Gree 03]. To estimate the illumination based on multiple orientation, we then stack the new RTFs as well as the extracted light intensities from multiple images row-wise to augment matrix T and vector i for equation (2.40).

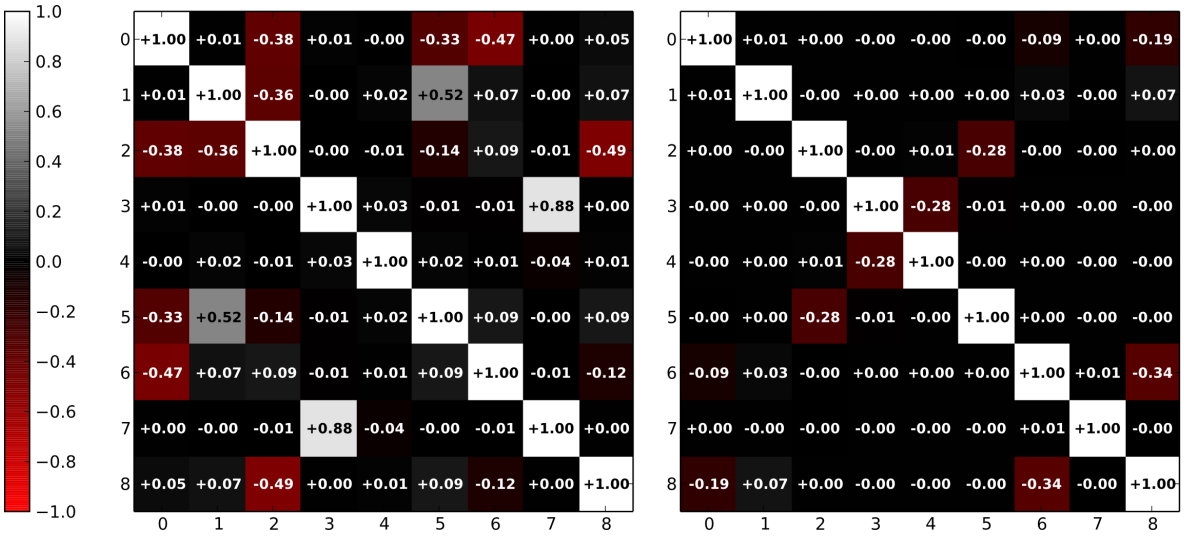


Figure 2.20: Some of the SH basis functions of the RTFs are correlated when using only a single image of a face (a). Combining multiple (4) images of a full 360° rotation around the yaw (b) axis dissolves this correlation.

In order to investigate how far combining information from multiple orientations mitigates the correlation between the RTF coefficients, we recalculate the correlation matrix for the RTFs in such an augmented matrix T.

We focus in here on the most natural case, a turn of the user around the yaw axis. We combine the RTFs from three rotations of the user, at 90°, 180°, and 270°, with the original ones and recalculate the correlation matrix, which is depicted in figure 2.20 (b). The resulting matrix points out that the correlation between the basis functions can be effectively overcome by including multiple images taken at different rotations around the yaw axis.

2.5.6.3 Deviations in the RTFs from the Learned Model

Our method for estimating the illumination relies on a limited range in variations between different human faces and the assumption that the learned RTF for a particular position is valid for all humans. In reality of course all faces are different, but small unbiased deviations in the RTFs are averaged out over the number of sample positions.

If for some person however a whole region of the face is very different from the learning dataset e.g. because of a macula, tattoo, beard, or bangs covering that region, the RTFs of sample positions in this region may heavily deviate from the learned ones. The resulting image intensities that are strongly inconsistent with the learned model thus may falsify the results of the estimated illumination.

In figure 2.21 (a) we for example see a black stick hold in front of the face covering part of the left cheek. The face exhibits light mainly incident from the upper left. Without the stick, our light estimation would create a coherent illumination of the virtual mask as demonstrated in figure 2.21 (b). The black stick however influences the estimation, the dark intensity values in this region vote against light incident from the left and thus lead to an inadequate result depicted in figure 2.21 (c).

(a) (b) (c) (d)

Figure 2.21: Parts in the image of the face, that do not comply with the learned model – simulated by a black stick in image (a) – falsify the estimation of incident light (c) compared to the estimation (b) without the stick. Detecting compromised sample positions as outliers (marked red in (a) and (d)) and removing them from the estimation counteracts this corruption in the estimation and recovers a solution (d) close to the true illumination.

To make the algorithm more robust for cases where the majority of the sample positions of the face is in accordance with the learned model but certain positions strongly deviate, we implemented an outlier detection step and exclude inconsistent sample positions from the light estimation.

To detect outliers, we multiply the estimated illumination onto each RTF and determine the residual in intensity between the predicted intensity and the actual intensity at the image position. Sample positions with residuals higher than the third quartile plus three times the interquartile range are labeled inconsistent and outliers respectively.

The estimation then can either be redone on the same frame this time without the outliers, or the outliers can be excluded for the next frame. Excluding the sample positions from the light estimation is equivalent to removing the corresponding rows from the system of equations 2.37 or respectively in the constrained case from matrix $\boldsymbol{T}$ and $\boldsymbol{i}$ in equation 2.40.

Sample positions labeled inconsistent by our experimental outlier detection are marked red in figure 2.21 (a), the resulting estimation after removing the respective outliers from the system of equations is illustrated in figure 2.21 (d) and indicates the effectiveness of the approach. Our experimental outlier detection works best for a small number of well-defined outliers but fails for too many outliers. An initial random sample consensus (RANSAC) [Fisc 81] that estimates the illumination on different smaller subsets of sample positions could perform better in finding the biggest consensus set of sample positions for a particular user. This iterative method however would also be a much more expensive process.

Even with the outlier detection, our approach still requires that the overall appearance of the particular face complies with the learned model RTFs. The outlier detection will fail in its current implementation, when the face shape is too different. Here in the future facial fiducial tracking could be employed, so that the sample positions better match the particular face shape. Our method will also deliver bad results when a person is for example wearing a cap which casts a shadow on the face, as this would violate the distant scene assumption. Multiple models for different faces and conditions could be learned instead of averaging over the whole training set and the most appropriate model for a user could be selected.

Note that for the following evaluations we did not use the outlier removal step.

2.5.7 Rendering of Virtual Objects

The goal why we estimate the real-world illumination is to match the lighting of virtual objects in Augmented Reality to the appearance of the real world. Now that we have estimated the present illumination and also have the pose of the camera relative to the face determined by face tracking, we are able to render the augmented image.

Not only the estimation of the incident light but also the rendering of the augmented scene (real plus virtual content) using the estimated illumination must run in real time. For that we pre-compute the radiance transfer of incident light for the virtual objects, as described in section 2.5.7.1. During real-time rendering (section 2.5.7.2) this pre-computed data then is combined with the live estimated directional distribution of incident light, which allows shading the virtual objects coherently with the appearance of the real world.

2.5.7.1 Offline Pre-Computation for Rendering

For the shading of our virtual objects, we at the moment only support direct lighting without inter-reflections as well as only diffuse materials. The incident light for the rendering stage is specified as before in form of the 2-dimensional function $E(\vec{\omega}_i)$ modeled as a Spherical Harmonics (SH) approximation with maximum degree $L = 2$. We pre-compute the influence of the incident light from the distant environment on the virtual geometry as described in [Sloa 08, Gree 03]. For every vertex x of the triangle mesh of a virtual 3-dimensional model we compute 9 coefficients c_n which describe the influence of SH basis function Y_n of the incident light on the intensity of the vertex:

$$c_n = \int_{\Omega(x)} V(x, \vec{\omega}_i) \cdot Y_n(\vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n}(x)) \, \mathrm{d}\vec{\omega}_i \tag{2.42}$$

This influence c_n depends on the surface orientation $\vec{n}(x)$ at the vertex as well as on the occlusion of the distant environment by the local scene itself. The occlusion by the local scene is represented by the visibility function $V(x, \vec{\omega}_i)$, which evaluates to 1 in case that the distant environment is visible into direction $\vec{\omega}_i$, and to 0 otherwise. Besides the virtual object, the local scene in these computations also contains a proxy geometry for the human head which may also occlude parts of the environment, depicted in gray in figure 2.22.

The calculation of the integral is done using Monte-Carlo integration [Hamm 64] by casting rays starting from the vertex position x on the mesh randomly into all directions. For ray casting we make use of the ray-tracing based rendering system pbrt [Phar 10]. When a direction is unoccluded, which means that the distant environment is visible in that direction, the value of the SH basis function is evaluated and compensated by the cosine of the angle between the sample direction and the surface orientation. All values are summed up and multiplied by 4π divided by the number of cast rays.

For each vertex we thus obtain a SH coefficient vector $\hat{C} \in \mathbb{R}^9$, which is supplied as a per vertex attribute in the rendering stage. Figure 2.22 illustrates the obtained coefficients of the SH basis functions over the surface of the virtual geometry model of a helmet. Each image corresponds to the influence of one SH basis function.

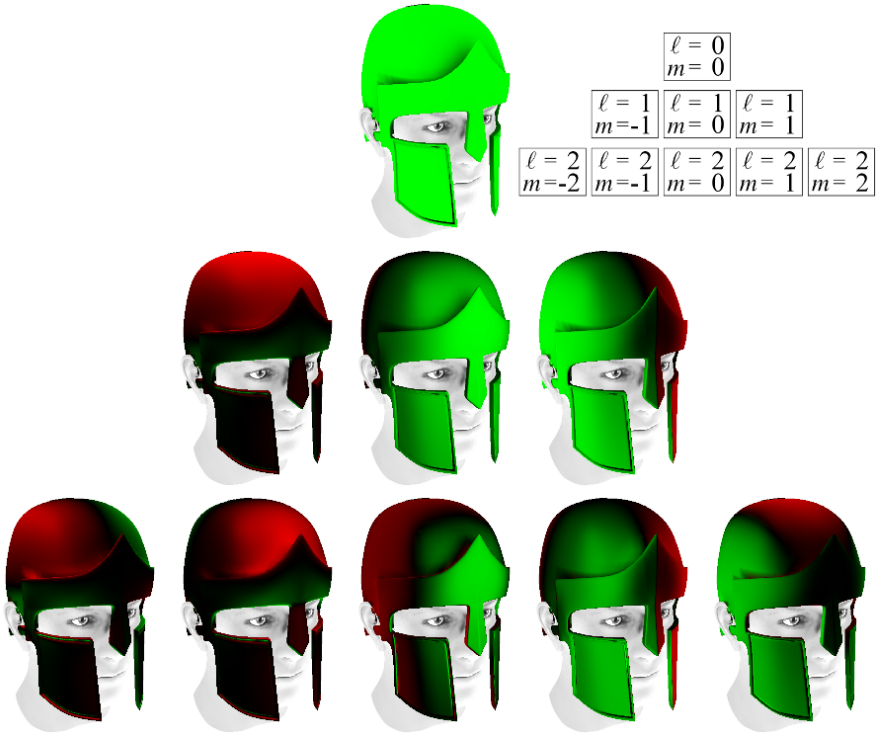


Figure 2.22: The pre-computed radiance transfer (shadowed, no interreflections) for the real-time rendering stage describes the influence (green symbolizes positive, red negative influence; the brighter the greater the influence) of each SH basis function (plotted in pyramidal SH structure in accordance with table 2.1) on the particular surface point.

In order to also simulate shadow cast by virtual objects onto the real face (see figure 2.24), we additionally pre-compute the differential change in the radiance transfer $\hat{C}$ with and without the virtual content for vertices of the proxy head model. As we at the moment use a generic proxy head model, the shadow contours from time to time are not perfectly aligned to the real face. We plan to address this problem by better adjusting the generic model to the particular user's face by fitting facial fiducials using a deformable face alignment framework like the one presented by Asthana *et al.* [Asth 14].

2.5.7.2 Real-Time Rendering using Pre-Computed Radiance Transfer

Our implementation for the real-time rendering part is based on the Metaio SDK [Meta 15] using OpenGL and GLSL. Thanks to the image-based face tracking, virtual geometry can be rendered in a fixed spatial relationship to the face.

The pre-computed SH coefficient vectors $\hat{C}$ from section 2.5.7.1 are supplied as per vertex attributes to the rendering stage. The estimated SH coefficients $\hat{E}$ of the directional distribution of incident light from section 2.5.5 are supplied in form of uniform arrays, with 9 coefficients each for red, green and blue light. The final irradiance for a vertex is determined by the dot product of $\hat{C}$ and $\hat{E}$.

Note that SH coefficients pre-computed for the geometry and SH coefficients estimated for the lighting are already in the same coordinate system as long as the virtual geometry is fixed with regard to the face. In order to support rotations of virtual objects, we can simply inversely rotate the illumination instead of rotating all the different SH coefficient vectors $\hat{C}$ of the geometry. The pre-computed shadow cast by the virtual objects onto the face however is invalidated when the objects are moved or rotated.

(a) (b) (c)

Figure 2.23: Coherent rendering with estimated light in different environments, where light sources are lamps on a ceiling (a), the sun and sky in an outdoor scene (b) as well as a combination of sky and lamps (c).

2.6 Results and Evaluations

In this section we will evaluate our light estimation method. Firstly we present visual results in section 2.6.1 in form of screenshots from an Augmented Reality application, which estimates the real-world lighting conditions using our method and then shades the virtual objects coherently. Secondly we compare our estimated illumination against the ground truth in section 2.6.2. Therefore we again employ images from *The Extended Yale Face Database B* [Geor 01, Lee 05] that we already know from the offline learning stage. Besides visually comparing estimated illuminations against ground truth, we also quantitatively evaluate our method in that section by measuring the difference between the estimated incident light and the ground truth. Here we also examine how the quality of the estimations correlates with the number of employed sample positions.

2.6.1 Qualitative Results on Webcam Sequences

For producing the visual results presented in here, we ran our light estimation method on live video sequences that either are captured by a webcam connected to a PC or by the user-facing camera of a tablet computer. The illuminations are estimated from the faces of two different users that both

have not been part of the training dataset. In most of the sequences the simple unconstrained solver variant with 294 sample positions has been employed. As described in the online estimation part (section 2.5.6), the pose of the camera relative to the face is determined by image-based face tracking. This pose is used for projecting the sample positions onto the camera image as well as for rendering virtual geometry in a fixed spatial relationship to the face ontop of this camera image. The geometry has been beforehand pre-processed (see section 2.5.7.1), so that once the illumination is estimated, it can be applied to the virtual content in real time.

Figure 2.23 depicts our method running under a variety of scenarios, indoor as well as outdoor. In figure 2.23 (a) the user is walking down a hallway. The light is coming from lamps at the ceiling. When the user is passing through below a lamp, the virtual helmet accordingly features a moving highlight on the top. In figure 2.23 (b) the user stands in a patio. On the image at the top the user is facing a wall in a corner of the patio and thus nearly no light is reaching the face and accordingly the virtual mask. On the image at the bottom the user has turned around. Now sky light is reaching the face, primarily from the right side, which adequately is reproduced on the mask. Figure 2.23 (c) shows a mixture of outdoor and indoor illumination. While the user itself is located outside and some amount of sky light falls onto his face, mainly artificial light from indoor lits the local scene through a window front at the right side. The image at the top shows the resulting illumination of the face, while the image at the bottom shows the final augmented image, where also the virtual helmet receives primarily light from the right, and casts itself a shadow onto the face.

The synthetic shadow cast from virtual objects onto the face, which is calculated using a proxy geometry for the face, is also spotlighted in figure 2.24.

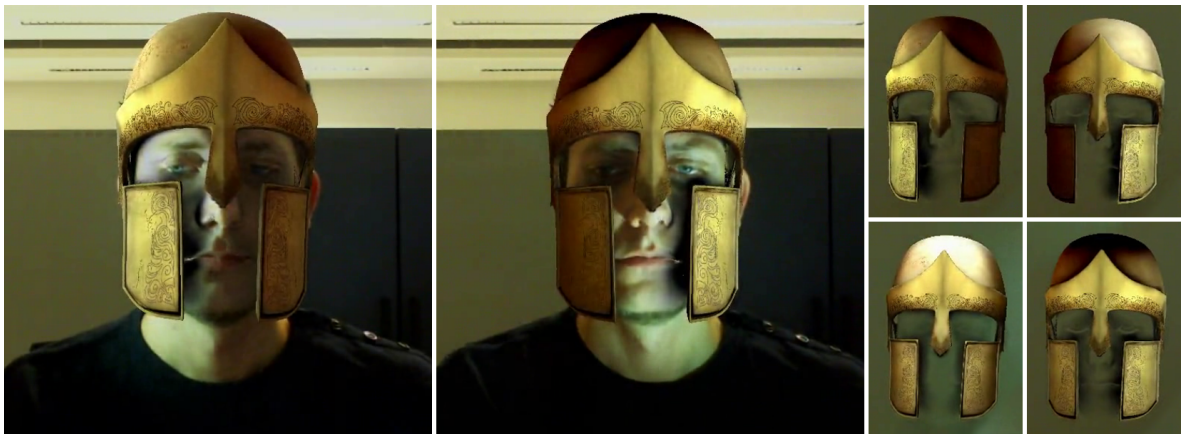


Figure 2.24: The left two images show simulated shadow cast by a virtual helmet on the user's face based on pre-computed differential radiance transfer using a generic 3-dimensional head model. The right subfigures show the differential shadow in front of a uniform background for better visualization of the change in cast shadow according to the estimated real-world illumination.

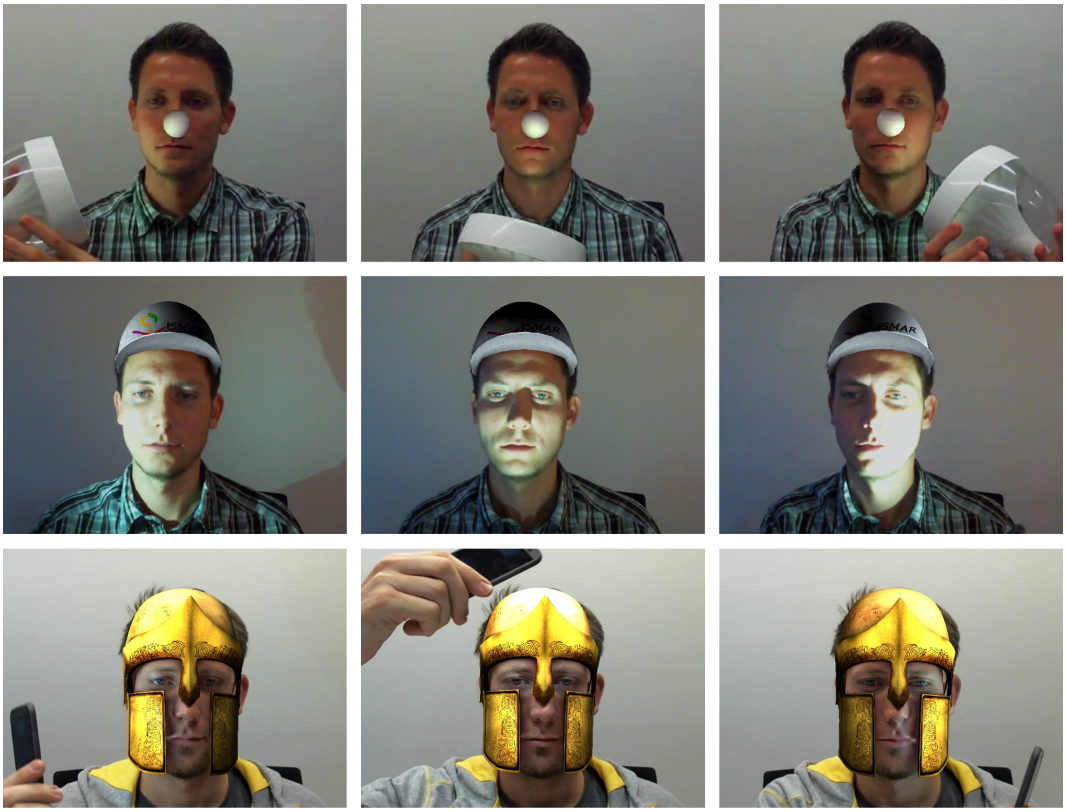


Figure 2.25: To illustrate the adapting illumination of the virtual objects, the user illuminates his face from different directions (from left, from above/below, and from the right) using a synthetic light source.

Figure 2.26: Examples showing a face under four different light directions and the coherent shading of a virtual cap. Note that the shading assumes that the virtual cap is positioned on the user's head – it has been moved and rotated for better visibility of the illuminated face.

In order to exaggerate how the estimated illumination adapts to changes, the person in figure 2.25 and figure 2.26 uses an artificial light source in order to illuminate his own face from different directions. Note, that this use case strictly speaking does not fully comply with the distant scene assumption. Still it is clearly visible, that the virtual helmet is illuminated consistently with the position of the light source and therefore with the illumination apparent in the face.

(a) (b) (c)

Figure 2.27: Based on separate estimations on the red, green, and blue image channels, we are able to reconstruct the color of the illumination, demonstrated in column (c), where the color of the virtual illumination of the white mask matches the appearance of an illuminated white paper sheet that is hold into the camera for ground truth comparison.

Our approach is also capable of estimating the color of the incident light by performing three separate estimations, one on each color channel (i.e. red, green, and blue). This is demonstrated in figure 2.27 where we use a light source with controllable color to illuminate the face of the user. Visible particularly well in the insets showing the virtual white clown's nose (i.e. sphere) attached to the user's nose in figure 2.27 (a, b), the estimation of color and direction succeeds and provides plausible illumination of the virtual contents. In the three images of figure 2.27 (c) another user is illuminated by the light source with controllable color. The user additionally holds a white sheet of paper, which is also illuminated by the colored light, into the view of the camera. The images demonstrate the ability of our approach to reproduce the colors of the illumination.

Further examples and visual results including real-time performance on image sequences can be found in the supplemental materials of [Knor 14].

All these examples show for a variety of lighting conditions that by applying the estimated illumination to the virtual content, the augmented scene consisting of real and virtual parts is shaded coherently. It demonstrates that our approach provides plausible results considerably enhancing the visual realism in Augmented Reality applications.

Timing Our algorithm thereby is able to run in real time. The simple unconstrained per frame illumination estimation for example takes less than 1 ms for grayscale and less than 2 ms for RGB estimation on a Lenovo ThinkPad Helix i7-3667U (Windows 8.1 Pro) using a set of 294 sample posi-

tions. The constrained quadratic solver with M = 100 constraints and 758 sample positions performs similarly fast with a runtime of less than 2 ms for both grayscale and RGB estimation.

Our full pipeline including camera tracking, light estimation, and rendering achieves real-time framerates. Rendering of the helmet (60k vertices) including cast shadows is for example performed at a framerate throttled at 30fps on a Lenovo ThinkPad Helix i7-3667U (Windows 8.1 Pro).

2.6.2 Comparison against the Ground Truth Illumination

Besides inspecting the resulting *augmented* images from live video sequences in order to visually examine the output of our algorithm in terms of coherent illumination, we additionally evaluate the performance of our light estimation method on the images from *The Extended Yale Face Database B* [Geor 01, Lee 05], the dataset we also use for training the RTFs. Because the directional illumination for these images is given, we can directly compare the estimated illuminations against ground truth. For all the evaluations on the images from the database, we beforehand divide the set of images into one part for training and a separate part for the evaluation.

In the following, we first *visually* compare the estimated incident light against the ground truth illumination in section 2.6.2.1. We then in section 2.6.2.2 also *quantitatively* evaluate our method by measuring the difference between the estimated incident light and the ground truth. We additionally investigate how the quality of the estimations correlates with the number of employed sample positions and compare the results from the unconstrained and the constrained solver.

2.6.2.1 Visual Comparison of the Estimated Illumination against Ground Truth

Our estimation of the illumination employs sparse sample positions that are distributed over the area of the face. Our methods thereby works for different numbers of sample positions. The first set of sample positions that we will use in our following evaluation is a set of 294 sample positions. This set is illustrated in figure 2.28.

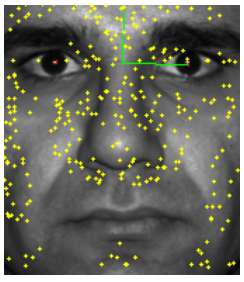


Figure 2.28: Set *N294* of sample positions.

For creating this set we started by distributing 512 sample positions by hand uniformly as possible over the image area of a frontal face. We then learned for all those sample positions their RTFs

according to section 2.5.4. Afterwards we selected a subset of the initial 512 sample positions, namely those ones with an influence, i.e. absolute coefficient, above the 90-th percentile for at least one SH basis function of the learned RTFs. By that we tried to guarantee, that every basis function is still represented after reducing the number of samples.

We refer to the resulting 294 sample positions in the following as *N294*. Unless stated otherwise, the subsequent visual comparisons of the estimated illumination against ground truth in this section are produced using this set of sample positions *N294* in combination with the unconstrained solver.

Figure 2.29 illustrates different parts of the original 512 sample positions that have an influence above a certain threshold. In this illustration the parts are chosen based on the 75-th percentile for each particular SH basis function while the final set *N294* was chosen according to the 90-th percentile in order to further reduce the number of sample positions.

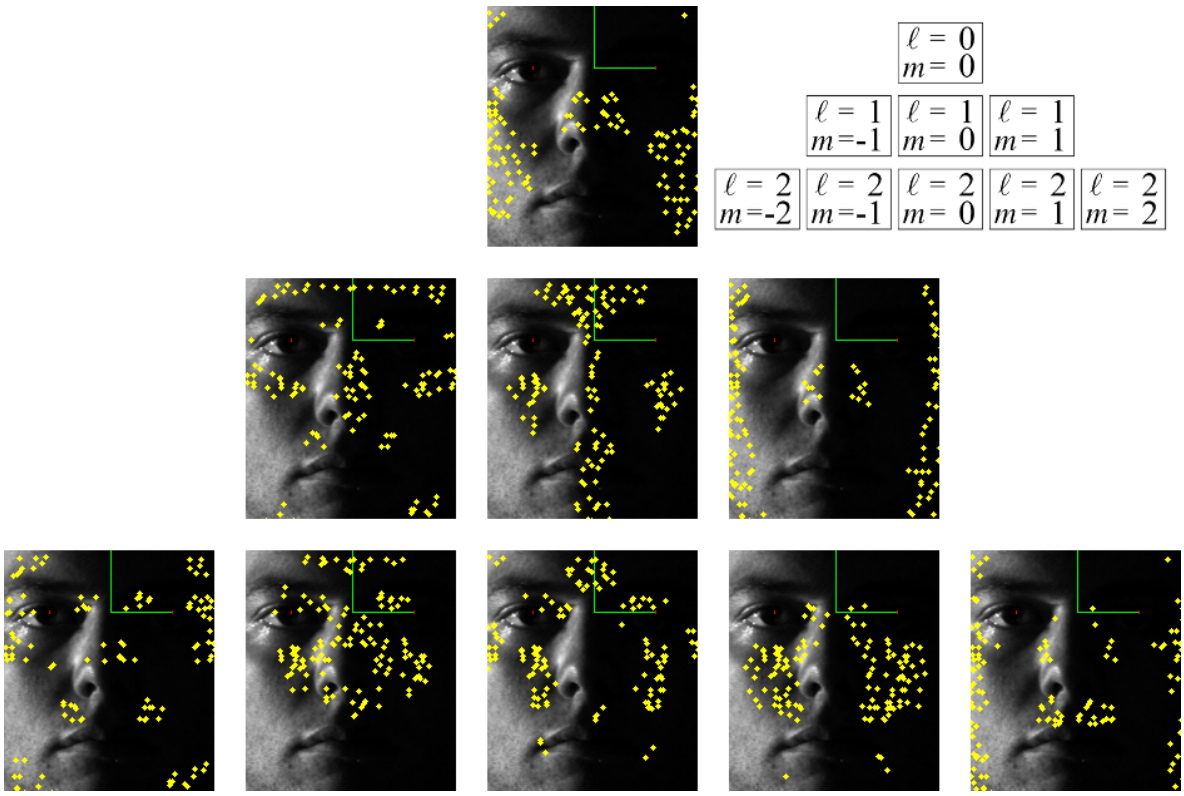


Figure 2.29: Parts of a set of original 512 sample positions, where sample positions in each part have an influence above the 75-th percentile in the particular SH basis function – illustrated in pyramidal SH structure (in accordance with table 2.1).

Figure 2.30 illustrates the results in estimated illumination by the unconstrained solver with 294 sample positions for a number of images from the database, comprising different faces under different directional illuminations.

The figure is structured as a grid, where each estimation is illustrated by 6 parts. Part (a) displays the

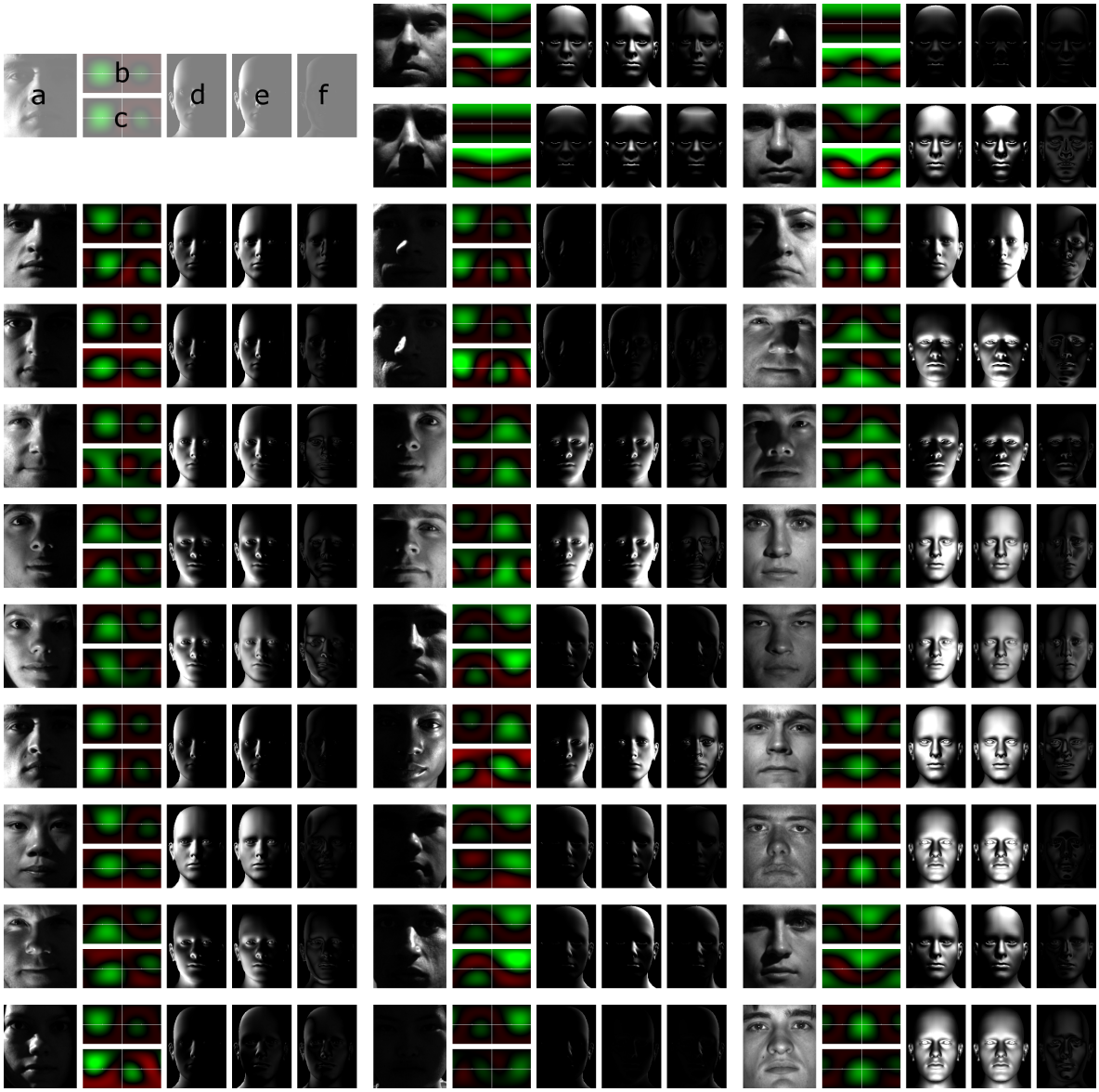


Figure 2.30: Comparisons between ground truth SH lighting and estimated SH lighting; Part (a) displays the input image of a face used for estimating the incident illumination. Part (b) illustrates the ground truth illumination. Part (c) illustrates the estimated illumination based on the image shown in part (a). Parts (d) and (e) show renderings of a virtual face geometry using the illumination from part (b) and part (c) respectively. Part (f) shows a difference image between parts (d) and (e).

input image of a face used for estimating the incident illumination. Part (b) shows a Lat-Long image depicting the *ground truth* illumination resulting from projecting the known directional illumination into SHs. The brightness of a pixel in this image represents the light intensity out of the direction corresponding to the pixel. Green values symbolize positive light intensities, while red values represent negative light intensities. These physically non reasonable negative values for particular directions arise from the approximation of the directional light source by projecting it into the low dimensional space of SHs and cutting off higher frequencies of the SH expansion. Part (c) shows the same kind of Lat-Long image, however depicting the *estimated* illumination based on the image shown in part (a). Part (d) and (e) show renderings of a virtual face geometry using the illumination from part (b) and (c) respectively for a better visual comparison of the effect of the illumination. The renderings do not consider occlusions (accounting for the surface orientation only) and use fully diffuse reflectance. Part (f) finally shows the difference image between the images from part (d) and (e).

The results demonstrate that the estimated illumination from the unconstrained solver with the set of sample positions *N294* in general is already comparable to the ground truth illumination also under harsh illumination from the side. The unconstrained solver however tends to overestimate intensities and compensates in return using also higher negative intensities like also illustrated in figure 2.17 (c). The constrained solver with $\varepsilon = -0.14$ introduced in section 2.5.6 overcomes this problem as shown in figure 2.17 (e). We suspect, that using negative intensities allows the unconstrained estimator to reproduce higher frequency effects visible in the input image like cast shadow and specularities, which could not be modeled by the low frequency RTFs.

The estimated albedo for different faces of the database seems to work reasonable well, visible in the similar scale of illumination of parts (d) and (e) in figure 2.30.

All the images from the dataset however only contain a single directional light source. For real-world applications, light is coming from all directions. To still have a valid ground truth illumination but not only a single directional light source, we create new images by blending two images of the same person under different illuminations similar to Zhang and Samaras [Zhan 03]. For that, we create a new image as linear combination of the two images with factors of 0.5 each considering gamma correction. Let $\hat{E}_1$ and $\hat{E}_2$ be the SH coefficient vectors representing the ground truth illumination of the first and the second image. Following the linearity of light, we calculate the ground truth illumination for the new blended image: $\hat{E}_B = 0.5 \cdot (\hat{E}_1 + \hat{E}_2)$.

Figure 2.31 illustrates some of these blended images. Column (a) contains a first image of a face, column (c) a second image of the same face with the corresponding ground truth illuminations $\hat{E}_1$ and $\hat{E}_2$ for the two images depicted in column (b) and (d). Column (e) contains the blended image resulting from the two original images with its corresponding combined ground truth illumination $\hat{E}_B$ in column (f). The last column (g) shows the estimated illumination, this time using the constrained solver with $\varepsilon = -0.14$.

For different people and illuminations the estimated solution from the blended images in most cases

(a) (b) (c) (d) (e) (f) (g)

Figure 2.31: We blend together images under different illumination to overcome the limitation of only a single directional light source in the ground truth dataset.

nicely reproduces the ground truth illumination and recovers the two main directions of incident light as far as they are representable by the low frequency SH approximation. Yet for example in row 4 of figure 2.31, the estimated illumination joined the two separate clusters from the ground truth illumination into one.

2.6.2.2 Quantitative Evaluation of the Estimated Illumination against Ground Truth

In addition to the previous analyses, which have been visual inspections of the estimated illuminations, we also quantitatively evaluate the results of our light estimation method. For that we again employ the images from *The Extended Yale Face Database B* [Geor 01, Lee 05], so that we have ground truth illumination available. Like before we divide the set of images from the used database into one part for training and a separate part for the evaluation.

Selection and Reduction of Sample Positions In the following quantitative evaluations we also want to analyze the influence of the number of sample positions on the quality of the light estimation in more detail. We therefore this time start with an even higher number of sample positions than for set *N294*, namely 1000. For an easier initialization as well as a more uniform distribution compared to the manual picking procedure applied for set *N294*, we this time employ Poisson disk sampling [Cook 86]. We will refer to the resulting set of 1000 sample positions as *N1000*. In a first step we remove, similar like for set *N294*, those sample positions from the initial set that are not

well-suited for estimating light. We this time do not rate sample positions based on the learned coefficients of the RTFs, but instead directly look at the variance in image intensity at the particular sample positions over the different images as we think this is more demonstrative while results are comparable.

Firstly, an *informative* sample must change its appearance when the illumination changes. Per sample location, we calculate for each person the variance in (albedo-corrected) image intensity over the different incident light directions from the dataset and then take the median of the variances from the different persons. The resulting values are plotted in figure 2.32 (a). Sample positions with good values are framed in green, bad ones in red. We consider values as bad, when below a certain threshold, which we set to half the maximum. As we can see, the mouth, eye and eyebrow regions are rated bad because their appearance does not change a lot depending on illumination.

Secondly, a *reliable* sample should behave consistently over the plurality of people. Per sample location, we calculate for each light direction the variance in (albedo-corrected) image intensity over different persons from the dataset. We then take the median of the variances from the different light directions. The resulting value per sample position is plotted in figure 2.32 (b). This time values are considered as bad if above half the maximum value, which identifies regions at the bottom part of the nose as well as above the eyebrows, most probably because of glossiness as well as differences in the face shape in these regions. Samples on the most lower left and right side are also labeled bad because of differences in the face shapes.

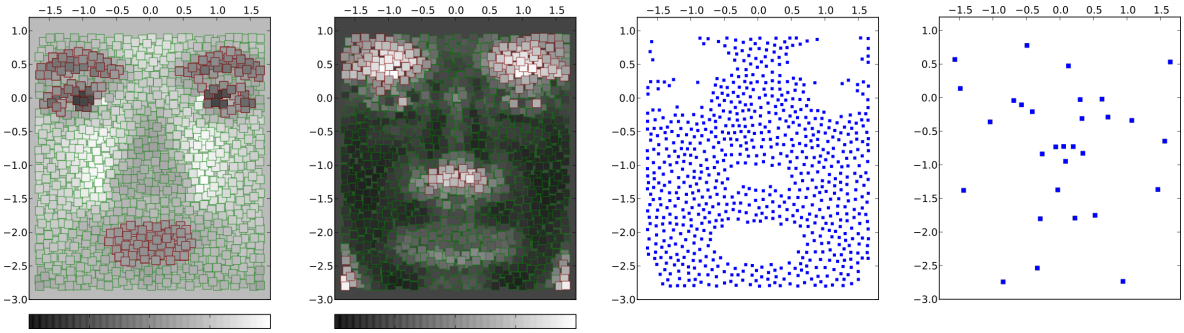


Figure 2.32: 1000 sample positions (set *N1000*) are distributed over the face region by Poisson disk sampling. The grey value of a sample corresponds to the variance in intensity over different light directions (a) or rather over different persons (b). Based on (a) and (b) a subset (c) of 758 sample positions (set *N758*) is selected which then is successively reduced (d).

Sample positions rated informative and reliable build the final set of 758 selected sample positions, shown in Figure 2.32 (c), which we refer to as *N758*. In figure 2.19 we already illustrated the corresponding learned RTFs for the set *N758*. Each image displays the coefficients of one particular SH basis function – negative coefficients are plotted in red, positive ones in white, the background is set to the median value of the coefficients. $\hat{T}_{j,0}$ for example depicts the response in intensity to ambient

illumination, $\hat{T}_{j,1}$ to $\hat{T}_{j,3}$ illustrate the response to linear parts of the SH illumination, $\hat{T}_{j,4}$ to $\hat{T}_{j,8}$ the response to the quadratic terms.

Starting from *N758* we now successively further reduce the number of sample positions by removing in each step the sample position which we consider most redundant. We therefore find two sample positions that are spatially close to each other and have a similar RTF. We then remove that one of the two which we consider less reliable based on the offline RTF estimation step.

By repeating this procedure we incrementally reduce the number of sample positions. Figure 2.32 (d) for example depicts a set of 30 remaining sample positions. Within the following quantitative evaluation of our light estimation method we also provide an analysis regarding different numbers of employed sample positions.

Angular Error in the Estimated Primary Light Direction We consult two quantities for our quantitative evaluation. The first one is the angular error δ between the ground truth light direction specified by the dataset and the primary light direction of our estimation. We determine the estimated primary light direction as the *optimal linear direction* [Sloa 08] extracted from the estimated SH illumination vector $\hat{E}$. Note that the extraction of the *optimal linear direction* and consequently the comparison thereof only utilizes the *linear* coefficients of the estimated lighting environment.

We evaluate the angular error δ for the different solvers described in section 2.5.6.1 as well as for the different sets of sample positions. The results thereof are depicted in figure 2.33 (a,b). Albeit there is some imprecision in the estimated primary direction, the estimations show a high degree of reliability. Note that the set of images that we used for the evaluation also contains images with lighting under extreme angles.

The candle stick diagram in figure 2.33 (a) compares the angular error δ for the three different solvers and the sets of sample positions *N294*, *N758*, and *N1000*. The diagram reveals that the different sets of sample positions perform comparably well, and that the number of sample positions in these higher dimensions has no major influence on the quality of the estimation.

The different solvers however make a difference. The unconstrained solver delivers a median error of $\approx 9^\circ$ with an upper quartile of around $\approx 14^\circ$. The constrained solver with $\varepsilon = 0$ performs notably worse with a median error of $\approx 14^\circ$ and an upper quartile of $\approx 21^\circ$. The constrained solver with $\varepsilon = -0.14$ finally performs best with a median error of $\approx 8^\circ$, an upper quartile of $\approx 12^\circ$ and a 95th percentile of $\approx 20^\circ$.

Figure 2.33 (b) shows the angular error δ plotted against the number of employed sample positions (starting from *N758* and successively removing a sample position) for the unconstrained solver (dotted lines) and the constrained solver with $\varepsilon = -0.14$ (solid lines). We in this figure omitted the underperforming constrained solver with $\varepsilon = 0$. The results indicate that our light estimation method still performs well for very sparse sampling, especially with the constrained solver, but also reveals a steady loss of accuracy with decreasing number of sample positions.

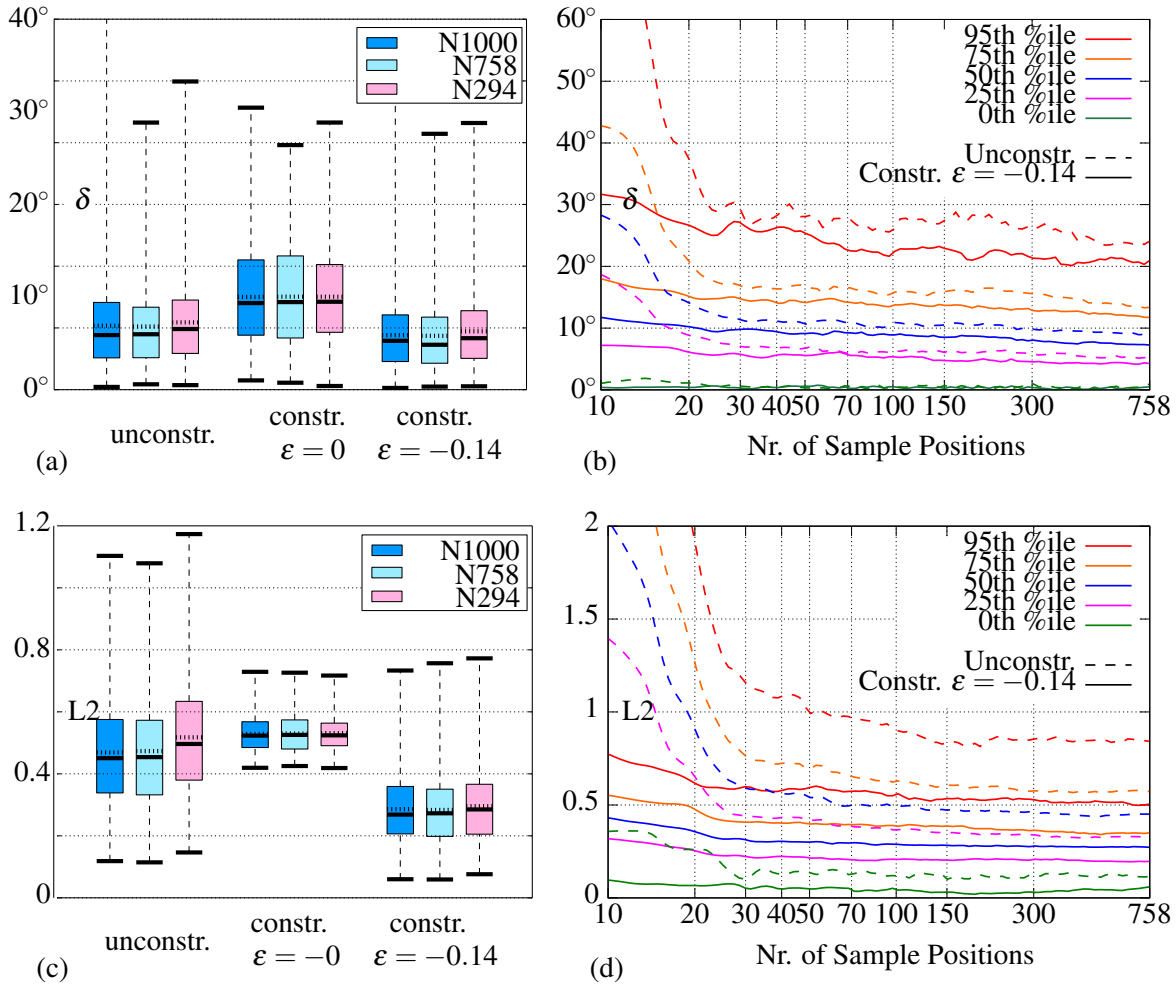


Figure 2.33: We evaluate the angular error δ in the estimated primary light direction (a,b) as well as the L2 Norm between the ground truth illumination and the estimated illumination both in SHs (c,d) for different solvers and numbers of sample positions.

L2 Norm between Estimation and Ground Truth Illumination To not only evaluate the primary light direction based on the *linear* coefficients but the whole directional distribution of incident light, we employ a metric induced by the L2 norm over the domain of directions as a second quantity for our evaluation, which measures the distance between the estimated illumination E_{Est} and the ground truth illumination projected to SH E_{GT}:

$$D(E_{Est}, E_{GT}) = \sqrt{\int_{S^2} (E_{GT}(\vec{\omega}) - E_{Est}(\vec{\omega}))^2 \, \mathrm{d}\vec{\omega}} \tag{2.43}$$

Note that while integrating the SH approximation of a directional illumination E_k of unit intensity over all directions results in a value of ~ 1, the L2 norm of that SH approximation equals ~ 0.85.

Figure 2.33 (c,d) shows the distribution of distance $D(E_{Est}, E_{GT})$ for light estimations using different solvers and sets of sample positions. The constrained solver with $\varepsilon = -0.14$ clearly outperforms the two other solver variants. The overestimated intensities as well as the negative intensities in the unconstrained solutions (see e.g. figure 2.17 (c)) produce a high discrepancy between the ground truth directional distribution of incident light and the estimation. In the case of the constrained solver with $\varepsilon = 0$ the flatted solutions (see figure 2.17 (d)) lead to a high median error of 0.52 with a narrow interquartile range.

Figure 2.33 (d) plots the performance in terms of distance $D(E_{Est}, E_{GT})$ against the number of employed sample positions. Similar to before the results demonstrate that especially the constrained solver still achieves good performance even for a much lower number of sample positions.

Summary This evaluation indicates, that already a smaller number of sample positions suffices in many use cases to estimate the illumination from the image of a face. A higher number however still performs adequately fast and delivers a more accurate, precise, and robust estimation.

Additionally the results clearly point out the superiority of the constrained solution over the unconstrained one. The constrained solver with $\varepsilon = -0.14$ does not only deliver the smallest distance $D(E_{Est}, E_{GT})$ but also achieves the highest accuracy with regard to the estimated primary light direction. The much improved quality resulting from the constrained solver with $\varepsilon = -0.14$ compared to the constrained solver with $\varepsilon = 0$ is a remarkable finding.

A quantitative evaluation of our method on images with environment light incident from all directions instead of only one directional light source is part of our future work.

2.7 Discussion

In the following we give a short summary and discussion about identified shortcomings of our current implementation. We also give pointers to open potential improvements that we see.

The results that we presented in section 2.6 demonstrate that our method lives up to the expectations we initially had. It accomplishes a coherent illumination of virtual objects in real time and considerably contributes to a plausible augmented view. Still for the future we see plenty of strategies for further improving the results.

First of all, the implementation of our method at the moment is restricted to a frontal view of the human face. This limitation could be loosened up and the method could be extended to other views. For different head poses different sets of RTFs could be learned based on images with known illumination showing faces under the relevant pose. For the online estimation the learned sets of RTFs that match best to the current head pose of the user could be applied for estimating the illumination.

A restriction that we chose deliberately is the approximation of the RTFs and the incident light by only nine SH basis functions. This low dimensional representation would be sufficient for lighting of convex diffuse geometries [Rama 01, Rama 06, Basr 03]. Like Ramamoorthi [Rama 02] analyzed, the variation within a single image of a convex diffuse object under arbitrary illumination could be even modeled by only five basis functions – as only roughly one half of all possible surface orientations is visible. These studies however deliberately disregard cast shadow. A human face is not fully convex, especially the region around the nose exhibits concavities. Skin also does not feature fully diffuse reflection. According to Epstein *et al.* [Epst 95] modeling variations by illumination with only a certain number of eigenvectors results in residuals, which indicates that the image contains more information about the illumination. Albeit our RTFs at the moment are recovered from real images and thereby *try* to capture occlusion effects within concave regions as well as skin specularities, the low dimensional approximation does not well model those higher frequent features. To better capture and evaluate effects like cast shadows and glossy reflections, we plan to investigate higher degrees of SHs as well as different function bases like (haar) wavelets [Okab 04] and sparse representations [Mei 11]. A reflectance model tailored to skin like the one presented by Weyrich *et al.* [Weyr 06] could be consulted.

Another limitation of the current approach is the way we align the sample positions to the faces. In the current implementation we align them using a coordinate system that is based on the positions of the eyes only. This approach does not well describe differences in facial proportions between different humans. Instead defining the positions on the face using additional facial fiducials like e.g. the positions of mouth and nose [Sim 01, Asth 14] or employing a full Morphable Model [Fuch 05], could result in a more accurate alignment of the sample positions and their RTFs.

A related limitation lies in the fact, that our goal was to find a *single* compact model for the RTFs that fits on different humans. We thus calculated the average RTFs over *all* persons from the train-

ing dataset. The intention to equally fit for all people however introduces imprecision. For higher precision, alternatively separate sets of RTFs could be learned for different persons from the training dataset and during the real-time light estimation the best fitting set of RTFs could be selected for the user e.g. based on similarities regarding the facial fiducial characteristics. A similar approach could also be used to learn different sets of RTFs for different conditions like when a person is wearing a mustache, hat, or a cap.

Another deficit of the current implementation is our simplified model of the camera response curve. We took this decision to easily support all different kinds of cameras without requiring the user to calibrate a particular camera. For physically meaningful estimations of the incident light a radiometric calibration of the camera however would be crucial. At the moment we only apply a standard gamma correction and beside that neglect non-linearity of the camera response function as well as parameters such as exposure, contrast or color saturation settings. By that our approach however mimics some of the camera effects because the effects are directly included into the light estimation. An underexposure of the imaged face for example leads to an estimation of very low light intensity and to coherent *underexposure* of the virtual objects.

Another related remaining challenge is an (online) albedo estimation for the user's face which becomes especially important for estimating colored (RGB) illumination and physically meaningful values. Here e.g. approaches based on active lighting using the camera flashlight or approaches that estimate the illuminant color based on highlights [Klin 88] – in our case highlights on the skin [Stor 00] – could be investigated.

Finally our work until now mainly focuses on coherent illumination for augmentations on the image of the user-facing camera. Here the virtual objects are close to the face which acts as light probe. Exploiting the knowledge about the illumination gained from the image of the user-facing camera also for augmentations on the world-facing camera image would make the method considerably more versatile in use.

Here however the limited knowledge about light coming from behind the user would need to be considered. It is actually exactly this light that is coming from behind the user which has a big impact on the objects visible in front of the user, while it is hard to estimate this relevant information about the illumination from the image of the user's face due to the visible surface orientations of the face. Potentially making use of the image intensities in the background region around the user visible in the image of the user-facing camera could help defining constraints onto the light estimation for light intensities incident from behind similar to the constraints used to enforce non-negativity in the constrained solver variant.

For an onstage demonstration of our light estimation approach with a dual camera set-up of user-facing camera and word-facing camera at the Augmented Reality conference InsideAR 2014 (see figure 2.34), we addressed this problem differently by restricting the possible real-world illuminations to a very small predefined discrete set. Spotlights above the stage could either be switched on or off.

We then employed our light estimation method to first estimate the present illumination in Spherical Harmonics. From this representation we then extracted the main directional light directions. Based on these directions we selected the best matching illumination for the virtual objects from a set of pre-defined configurations. For shading the virtual objects we picked a corresponding pre-baked texture including a shadow plane.

Figure 2.34: InsideAR 2014: The illumination present on stage is estimated from the image of a face captured by the user-facing camera and the augmentation – a virtual armchair – on the image of the world-facing camera is shaded accordingly.

Figure 2.34 shows two photographs of the onstage presentation. The person in the back is holding a tablet PC. This tablet PC features two cameras, a user-facing camera and a world-facing camera. The two video streams captured by the both cameras are shown on the screen in the background of the persons. The image of the user-facing camera – shown as smaller image on the right of the screen – which captures the face of the user is used to estimate the illumination. The world-facing camera captures the scene in front of the user – shown as larger image on the left of the screen. Here the estimated illumination is used to coherently shade an augmented virtual armchair. Between the two images in figure 2.34 a different set of spotlights is switched on, so that the illumination changes. The shadow direction and illumination of the virtual armchair augmented on the world-facing camera image adapts to the illumination visible in the real world, visible at the coherent direction of the shadow cast by the other person.

In future work we plan to further investigate the suitability and expandability of our light estimation approach for augmentations on the image of the world-facing camera.

2.8 Conclusions

In this chapter, we presented an approach for coherent illumination of virtual objects in Augmented Reality applications. We proposed to leverage the user's face as a light probe for estimating the lighting present in the real world. We will in this section conclude the light estimation part of this thesis with a short wrap-up of the main findings and contributions of our approach.

We demonstrated a real-time method for estimating the illumination within a scene that is particularly suitable when augmentations are rendered directly into the image of the *user-facing* camera, like e.g. in virtual try-on applications or augmented video conferences. For these kind of AR applications our method enables a coherent illumination for the virtual content.

By discovering and exploiting the fact that the face of the user is always within the scene and can be conveniently captured in many cases by a user-facing camera, we eliminated the need for a separate illumination estimation step present in many state-of-the-art approaches without us demanding any special hardware. We also overcame the requirement of additional known objects or markers. Hidden from the user, the estimation can be performed without the user even taking notice.

The algorithm thereby runs in real time even on mobile devices. The limited range in variations between different human faces makes it possible to create a two-step algorithm.

In a first step we learn the correlation between the light intensity incident on the face and the light intensity leaving the face for a set of sample positions on the human face. The learning is based on a set of images showing frontal views of different human faces under different known illuminations.

The gained knowledge subsequently is used in a second step for estimating in real time the incident light from a single image of a potentially unknown face. The more expensive learning process thereby has been extracted into the offline part, allowing a fast estimation at run time with low impact on power consumption.

We intentionally started by designing our light estimation approach as simple as possible employing an unconstrained least squares solution [Knor 14]. We showed that this simple approach already achieves pleasing results in a variety of cases.

Afterwards we identified weaknesses of the existing approach and extended our original method by a constrained solver in order to overcome negative light intensities in the solutions. We demonstrated effectiveness of the improved method in a quantitative evaluation against ground truth and pointed out that it is important to allow for some amount of negative light intensities.

Additionally we analyzed how our method performs for a varying number of sample positions. We could show that a small number of sample positions suffices, which underlines the suitability of the *sparsity* of our approach.

We also touched upon inherent limitations of our chosen approach and gave pointers to solutions e.g. by using multiple images or an outlier removal procedure.

The effectiveness of our method has been demonstrated in ground truth comparisons as well as under a variety of scenarios presented in image and video footage. Realistically showcasing products to the user will be a major requirement for successful AR kiosks and web- or mobile-based shopping applications. Estimating the present illumination is an important step for coherent rendering and is achieved by the method presented in here without posing any additional challenge to the user.

We thus have presented a method, that can already be employed as is, in order to enhance the realism of the augmented view in Augmented Reality applications by providing a coherent illumination of the virtual objects. We however hope that our findings are just the starting point for future research based on the idea to employ the user's face as a known object to gain knowledge about the real-world surroundings.

3 Absolute Scale - The User's Face as an Object of Known Size

In this chapter we will employ the *user's face as an object of known size* to resolve the ambiguity in scale of the reconstruction of an unknown environment by images of a monocular camera. We target the use case, where the scene is reconstructed with a handheld device which features two cameras: a first camera that is facing the world and that is used to reconstruct the unknown environment using monocular SLAM, and a second camera that is facing the user. When we know the distance between two points on the user's face in absolute units like meters – e.g. the distance between the user's eyes – we can also determine the motion of the user-facing camera with respect to the face in absolute units using face tracking. As long as the face remains stationary in the real world, the motion of the user-facing camera with respect to the face can be transferred to the motion of the rigidly connected world-facing camera with respect to the world. Knowing the latter motion at absolute scale allows us to bring the reconstruction of the unknown environment that is purely based on images of the world-facing camera from arbitrary scale to absolute scale. Tracking the world at absolute scale enables us to superimpose virtual objects in an Augmented Reality view at true size. The reconstruction at absolute scale also allows us to directly perform distance measurements in the reconstruction.

We present an approach to estimate absolute scale in handheld monocular SLAM by simultaneously tracking the user's face with a user-facing camera while a world-facing camera captures the scene for localization and mapping (figure 3.1). This chapter is structured as follows.

We start by giving an introduction to the topic of absolute scale in the context of Augmented Reality (AR) in section 3.1, where we first introduce monocular SLAM, a popular technique broadly used

Figure 3.1: In monocular visual SLAM the structure of a scene as well as the motion of the (world-facing) camera are only estimated up-to-scale but can be brought to absolute scale by simultaneously capturing and tracking the user's face in the user-facing camera (left). This enables superimposing virtual objects, e.g. the green wire frame model of a parcel, at absolute scale (right).

for camera localization for handheld AR in unknown environments. After pointing out the inherent ambiguity in scale in monocular SLAM we explain why this ambiguity presents a problem for AR. We then give a brief summary on how we will overcome this problem by estimating the scale using an additional user-facing camera.

After this short wrap-up of our own method, we subsequently compare our method in section 3.2 to existing state-of-the-art approaches for obtaining absolute scale in monocular SLAM.

We further elaborate the idea behind our particular method in detail in section 3.3, before we then provide details on the actual implementation in section 3.4. Quantitative evaluations of the accuracy and precision of our method are presented in section 3.5, where we examine the performance of the absolute scale estimation both for an idealized case using marker tracking (section 3.5.1) to explore the future potential, as well as for the real-world scenario using face tracking (section 3.5.2).

On grounds of the evaluated performance, section 3.6 illustrates multiple applications that are enabled by our approach.

Finally we conclude this chapter, which is targeted on absolute scale estimation, with a discussion and conclusions about our presented approach in section 3.7 .

Note on Publication All major contributions of this part of the thesis have already been published by Knorr and Kurz [Knor 16] in the proceedings of the IEEE *International Symposium on Mixed and Augmented Reality (ISMAR)* 2016. The fundamental research was conducted by the first author, Sebastian Knorr, under the technical and project-administration guidance of the second author, Daniel Kurz. In particular, the theory of the approach as well as the implementation was developed by the first author.

3.1 Introduction to Absolute Scale in Augmented Reality

In this section we will introduce the concept of absolute scale in the context of scene reconstruction and camera localization for AR. We first will give a brief summary about basic principles of camera localization for AR in unknown environments in section 3.1.1, with an emphasis on the commonly used technique called monocular SLAM in section 3.1.2. From that we will point out in section 3.1.3, that there is an inherent ambiguity in scale when we reconstruct an unknown scene with a monocular intensity camera without having additional knowledge about either the dimensions of the scene or the dimensions of the camera motion. We also will point out why this ambiguity in scale presents a problem for AR, where we want to use the scene reconstruction and camera localization to embed virtual objects into the real world. In section 3.1.4 we then will give a short introduction to how our particular approach overcomes the problem of ambiguity by estimating the scale of the environment using an additional user-facing camera that captures the user's face.

3.1.1 Camera Localization for Augmented Reality

Augmented Reality (AR) enriches our perception of the environment by additional digital content that is spatially registered to the real world. Technologically AR often is implemented as video see-through AR. Here the visual view of the real-world environment is provided in form of a live-video stream which is captured by a camera and then is presented to the user on a display. The digital content thereby is embedded into the view of the real world by overlaying computer-generated renderings of the digital content on top of the video stream. Digital content may comprise virtual 3-dimensional objects which act as surrogates for real objects. AR thus can for instance be used for product previews allowing the user to see how some furniture would look like in the own living room.

To generate the illusion for the user that the virtual objects are actually placed within the real world, the virtual objects must be rendered into the augmented video in a way that they are spatially registered with the real world. That means they must stay at the same position with respect to the real world. When the real video camera is moved or rotated, the perspective of the rendering of the virtual objects must adapt accordingly to the changed perspective of the real world visible in the video stream.

The pose of the camera with respect to the real world thus must be determined, in order to use the same pose for the virtual camera during rendering the virtual objects. The terminology *pose* used inhere refers to both *the position and the orientation of the camera*. In 3-dimensional space, this pose comprises six degrees of freedom (6DoF): three modeling the position, three modeling the orientation in space. These parameters are also referred to as the extrinsic parameters of the camera because they are independent – i.e. no intrinsic properties – of the camera itself.

Beside the extrinsic parameters of the camera, additional parameters influence how the 3-dimensional world is projected onto a 2-dimensional representation during image formation. A commonly used model to describe the projection by a camera in a simplified way is called *central projec-*

tion (see e.g. [Hart 03]) or pinhole camera model. In this model, all the rays of light that are captured by the camera pass through a common point – the pinhole, which is the center of projection of the camera. The direction of each ray determines the position where the ray of light hits the image sensor. In the simplest case, a point in the 3-dimensional world is imaged at the position where the ray from that 3-dimensional point through the center of projection of the camera intersects with the image plane. The light intensity along a ray incident on the sensor creates the intensity of the image pixel located at that position. Each location of the 2-dimensional image that is created by the camera thus corresponds to a ray through the center of projection of the camera into some direction in the 3-dimensional world. How directions are mapped to image locations in particular depends on intrinsic parameters of the camera like focal length, sensor format, and principal point and may additionally be influenced by lens distortion which alters the optical path of light through the lens system. The intrinsic parameters of a camera can for example be calibrated using images of a checkerboard by the method of Zhang [Zhan 00]. Commonly in camera localization the intrinsic parameters of the camera are considered to be fixed and calibrated in advance. In the following we also will assume that the intrinsic parameters of the employed cameras have already been calibrated, and we will focus on the estimation of the 6DoF pose of the camera within 3-dimensional space.

As mentioned earlier, the current camera pose with respect to the real world is needed to superimpose virtual objects under the adequate perspective that matches the perspective of the real world in the camera image. Determining the camera pose with respect to the real world is often also referred to as *camera tracking*. Tracking the pose of a moving object like the camera can be performed in various ways, e.g. using GPS [Fein 97], magnetic sensors [Livi 97], Wi-Fi-based signals [Ferr 07, Liu 07], inertial measurements [Harl 13], or some combination thereof [Miro 13]. These approaches however lack accuracy. In order to track the pose of an object with a very high accuracy and precision often OptiTrack[1], an optical outside-in tracking method, is used. For this technology, small reflective marker balls are attached to the object to be tracked. These reflective balls are captured and tracked by a calibrated array of cameras which emit invisible infrared light. While this method delivers very high accuracy, it involves expensive hardware and set-up steps and thus is not reasonable for the mass market. For handheld video see-through AR instead inside-out camera tracking, i.e. vision-based localization of the handheld device using the camera stream that is captured anyway for the augmented view, is an effective and commonly used way. It allows for determining the current camera pose at an accuracy that is adequate for AR without posing additional requirements. Inside-out tracking thereby determines the camera pose with respect to the captured environment.

If we already have a 3-dimensional model of the real world and are able to identify parts from this model in our images, the 6DoF pose of a (calibrated) camera can be estimated relative to the real world based on a set of correspondences between 3-dimensional points of the model and their corresponding projected 2-dimensional image locations in the captured image. Beside establishing the correspondences, estimating the camera pose involves solving a system of equations that is constructed

[1] `http://optitrack.com/`

based on the observed correspondences – a problem known as Perspective-n-point (PnP). Methods solving the PnP-problem come in various flavors, from methods that work on the minimal set of three correspondences (P3P) like the ones presented by Gao *et al.* [Gao 03] or Kneip *et al.* [Knei 11], to approaches that work with a higher number of point correspondences like e.g. EPnP from Lepetit *et al.* [Lepe 09], which is applicable for any number of correspondences greater than three. This principle of estimating the camera pose from correspondences between 2-dimensional image locations and 3-dimensional points of the model of the scene is also used for planar object tracking, where the 3-dimensional world is restricted to a planar known object like an image or specialized marker.

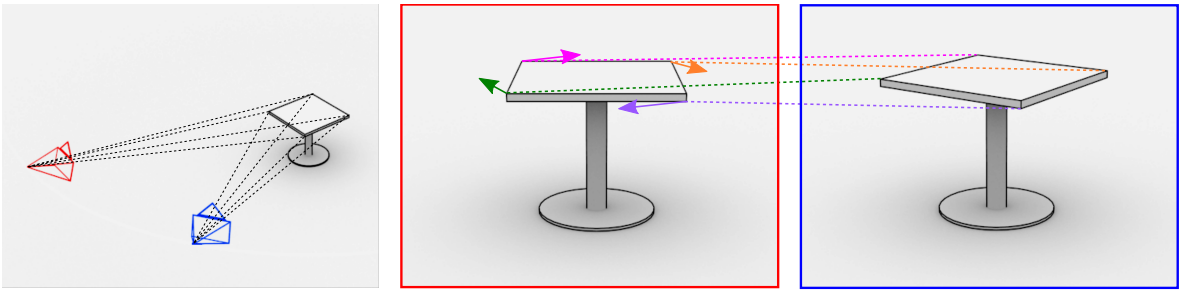
In many scenarios of handheld AR, the environment around the user however is unknown, e.g. when the user is executing the AR application in their own living room. Image-based camera tracking thus also has to work in generic and unknown environments where no *initial* predefined model of the real world and its 3-dimensional structure is available. In this case also the model of the world must be estimated from the 2-dimensional images of the environment which are captured by the camera from different views, while the user is moving through the environment. This technique, to infer structure from sequences of projections, is referred to as structure from motion (SfM). Generally speaking, SfM approaches find a model of the world that best explains the observations in the multiple images. The environment thereby often is also referred to as the *scene*, and the model of the scene is called map. Mapping accordingly refers to the process of creating the map of the environment.

3.1.2 Monocular SLAM for Camera Localization in Unknown Environments

Visual *Simultaneous Localization and Mapping* (SLAM) is a real-time variant of SfM. It describes the process of observing a scene with at least one camera from different viewpoints, and over time building a consistent 3-dimensional model of the scene, as well as *simultaneously* estimating the poses of the camera at the observations. The incremental creation of the map during runtime in combination with the real-time estimation of the current camera pose make this process an ideal candidate for camera tracking for AR, where the current camera pose is needed instantaneously for rendering the virtual objects on top of the current camera image.

Various methods of visual SLAM exist for different types and setups of cameras. In the simplest case only a single (intensity) camera is used to observe and reconstruct the scene. This is referred to as *monocular* SLAM, see e.g. Davison *et al.* [Davi 07]. The benefit of monocular SLAM for handheld AR lies in its low hardware requirements. Nowadays the required kind of camera is already an integral part of nearly all available handheld devices.

The consecutive images from the video stream that is captured by the camera are also referred to as frames. When the moving camera captures images of the scene from different viewpoints, and thereby observes the same scene location multiple times (see figure 3.2), the 3-dimensional position of that location can be recovered by triangulation using the parallax, i.e. the angle between the ray directions



(a) (b) (c)

Figure 3.2: Two images (b, c) of a table are captured from two different camera poses (a). Four correspondences for corners of the table are depicted with stitched lines. The displacements of the correspondences between the first (b) and the second image (c) are visualized as arrows (b).

associated with the image locations of the observations, to estimate the depth for the scene location. This however requires knowledge of the camera poses.

Over the past years, various implementations of monocular SLAM have been presented. Our method, which estimates the scale of the reconstruction that is created by monocular SLAM, treats the particular implementation as a black box. Our method directly works on the estimated camera poses. By that it is agnostic of the particular algorithm that estimates the poses and thus would even work in combination with methods like visual odometry (VO), see e.g. Nistér *et al.* [Nist 04] and more recently Engel *et al.* [Enge 13, Enge 16]. These methods do not necessarily create a global map, but primarily sequentially estimate the motion of the camera over time. For VO, our method could determine the scale in which the trajectory of the camera is estimated.

An implementation of a visual SLAM system consists of several building blocks, and every building block can be designed in various ways. While an exhaustive overview is beyond the scope of this thesis, we will in the following shortly review some of the main differences between popular approaches of monocular SLAM.

Filter-Based versus Keyframe-Based Visual SLAM One main characteristic of a visual SLAM implementation is the way how the history of observations as well as the current state of the process is kept and updated over time.

Earlier methods of visual SLAM like *MonoSLAM* by Davison *et al.* [Davi 07] often employed a filter-based approach. Here the information gained from the images over time is sequentially *fused into probability distributions* for the observed features and the camera pose.

More recently, visual SLAM methods switched to a different approach, where for a subset of the captured images, the camera poses as well as the observations from the images are stored as so called *keyframes*. While camera tracking still can be performed frame by frame, the stored keyframes allow for a global optimization of the reconstructed map that is decoupled from the tracking thread. This

idea was introduced by Klein and Murray [Klei 07] in their *Parallel Tracking and Mapping* (PTAM) framework. The global optimization in this case is done using *Bundle Adjustment* [Trig 99], a technique where the 3-dimensional features in the map are refined simultaneously with the estimated camera poses at the keyframes. The refinement is realized by minimizing a cost function e.g. the reprojection error of the 3-dimensional map features onto the 2-dimensional keyframe observations.

Based on simulating a number of experiments with camera motion in local scenes, Strasdat *et al.* [Stra 12] concluded that typically Bundle Adjustment is more efficient (in terms of accuracy per computational cost) for visual SLAM than filtering. Bundle Adjustment over 6DoF camera poses however is not the only solution for a global map refinement. In *Large-scale direct monocular SLAM* (LSD-SLAM), Engel *et al.* [Enge 14] for example optimize over a pose graph of keyframes which additionally considers scale between different poses by using similarity transformations instead of rigid transformations.

The global map of keyframes and the reconstructed 3-dimensional map features that are generated over time can also be used to recover the camera pose with respect to the world when tracking has been lost. The camera pose then can be reinitialized from the map. Even when tracking is not lost, the global map can be used to detect areas that have already been visited. The connection between the current camera pose from sequential tracking and the estimated camera pose based on another keyframe stored in the past can be used to compensate drift in the camera pose estimation that was accumulated over time and to again refine the global map. This correction to the map and to the camera poses is known as *loop closure*. The detection of already visited areas can for example be performed using the extracted 3-dimensional features of the map [Clem 07] or by searching for stored keyframe images that are similar to the current frame [Ange 08].

Direct versus Indirect, Dense versus Sparse Monocular visual SLAM recovers the 3-dimensional structure of the scene based on multiple observations of the scene from different viewpoints. It thereby relies on identified correspondences between multiple 2-dimensional images. These correspondences are pairs of 2-dimensional locations within different images which correspond to the same part of the scene. Where the part of the scene is projected to in each image thereby depends on the particular camera pose with respect to the scene part. The observations thus build up constraints on both the unknown structure of the scene and the unknown camera poses.

To identify correspondences between images, some methods of visual SLAM directly work on image intensities and minimize the photometric error. These methods are referred to as *direct* methods. Often direct methods thereby recover the relative pose between images by aligning the whole images through minimizing the photometric error between the images over all the pixels. Methods that take the full images into consideration, and by doing that exploit all information present in the image, are referred to as *dense* methods. *Dense Tracking and Mapping* (DTAM) by Newcombe *et al.* [Newc 11] is an example for a direct and dense method. Also LSD-SLAM by Engel *et al.* [Enge 14] aligns

two images by directly minimizing the photometric error between the two images. The small per-pixel disparities between consecutive frames are used to successively build and filter a (inverse) depth map along with corresponding variances. Each keyframe that is created over time stores the captured camera image as well as the corresponding (inverse) depth map including variances. The depth map however only is defined for those image regions close to large intensity gradients, i.e. regions where the per-pixel disparities are more reliable. Due to this selective non-dense information, LSD-SLAM can be described as direct and *semi-dense*.

The alternative to dense methods are so called *sparse* methods, that only establish correspondences for a sparse set of image locations. In a first processing step, small image regions are identified in the image that are well-suited for finding the corresponding image region in other images. Regions with high texture obviously are easier to match than uniform areas. The identification of well-suited image regions is called *keypoint or (salient) feature detection*. Famous examples of keypoint detection methods comprise the *Harris corner and edge detector* by Harris and Stephens [Harr 88], *difference-of-Gaussian* (DOG) by Lowe [Lowe 99], as well as the *Features from Accelerated Segment Test* (FAST) method by Rosten and Drummond [Rost 06], which is a computationally efficient corner detector. Good features are regions in the image that look similar in different images while each feature looks distinct. A good feature detector (e.g. FAST) is repeatable and invariant. It detects the same features under different illumination, viewpoint, blur, etc.

Sparse methods often do not directly minimize the photometric error to find the perfect match for an image region in the other image, like direct methods do. Instead, before matching, the features (image regions) identified as well-suited are first further processed into a different representation. In this case the method is referred to as *indirect*. After processing, an image region is described by a so-called feature descriptor, e.g. a 128-dimensional vector. The algorithm that transforms the image region into its new representation in a particular way is called feature descriptor method. Two widely used feature descriptor methods are the *Scale Invariant Feature Transform* (SIFT) by Lowe [Lowe 04] as well as the *Binary Robust Independent Elementary Features* (BRIEF) from Calonder *et al.* [Calo 10]. A feature descriptor method does not only define the processing of image regions into feature descriptors but also provides a similarity measure on the feature descriptor space for matching. While SIFT for example considers the Euclidean distance between two feature descriptors, BRIEF uses the Hamming distance [Hamm 50]. A good feature descriptor method is invariant and distinctive. It describes one and the same feature under different illumination, viewpoint, blur, etc. with a similar feature descriptor but describes two different features with preferably distant descriptors.

While sparse methods are only able to reconstruct a sparse 3-dimensional point cloud of the scene, dense methods can recover a much denser 3-dimensional model.

Motion Model Visual SLAM tracks the camera motion over consecutive images. The small motion assumption implies that the change in camera pose between the last frame and the current frame is relatively small and as a consequence also the images are quite similar. The camera pose for the

current frame needs to be estimated by either aligning the current image to the model or by tracking sparse features. For both of these matching procedures the camera pose of the last frame can be used as an initialization: either for the image alignment or for the feature locations in the current frame.

Using directly the camera pose of the last frame as initial guess for the current pose is referred to as constant position model. Starting from this initialization, the alignment then can be successively optimized.

Some methods incorporate a more elaborate motion model than simply starting from the previous frame: PTAM [Klei 07] for example uses a constant velocity model, which includes the velocity of the camera into their prediction for the current camera pose that then is used as initialization for the optimization. This allows for a better initialization and thus smaller search area for the feature locations in the images even for larger camera motion between two consecutive frames, but imposes the assumption of a smooth camera motion.

Initialization The map in SLAM is extended incrementally. Camera poses are estimated with respect to the map and new map features are integrated based on estimated camera poses. At the very beginning however both the scene as well as the camera pose with respect to the scene are unknown. An important aspect for an implementation of monocular SLAM thus is the initialization phase.

MonoSLAM [Davi 07] for example is restricted to only initialize with the camera facing a planar known initialization target, e.g. a black rectangle. The features of the target, i.e. the corners of the rectangle, need to be already predefined in the map, and the camera has to be held in a certain known location relative to the target. MonoSLAM uses this restricted set-up to directly estimate a camera pose for the first frame.

In order to not rely on a specific predefined target, more recent methods of monocular SLAM try to initialize by exploiting information from multiple frames captured by the moving camera to establish the first initial map.

PTAM [Klei 07] for example adds an initialization phase, where the user has to capture the scene while performing a smooth translation with the camera. The user explicitely marks the start and end frames of the captured initialization sequence. These frames are considered to be the first two initial keyframes. Features detected in the images are tracked frame to frame. By that, image correspondences between the first and last frame are established. These 2D-2D correspondences then are used to calculate the relative camera rotation and translation between the first and the last keyframe and to determine the 3-dimensional coordinates of the tracked features by triangulation. This triangulation of features only works well when the baseline, i.e. the translational offset, between the camera positions of the two initial keyframes is sufficiently large, depending on the distance of the observed features from the camera.

In the original version of PTAM [Klei 07], the matched 2D-2D correspondences are fed into a five-point-algorithm [Stew 06] that estimates the essential matrix, which is a matrix that describes the

relation between image coordinates of a 3-dimensional point captured in two different images by a calibrated camera. From this essential matrix the relative camera pose between the two keyframes can be extracted and the 3-dimensional coordinates of the features can be determined.

In a later version, the initialization for PTAM is limited to planar scenes, and instead of the more general essential matrix, the homography matrix is estimated, which is a transformation matrix that relates two projective mappings of points on a plane in 3-dimensional space. Again the relative camera pose between the two keyframes can be extracted from this homography, and the 3-dimensional coordinates of the points on the plane relative to the camera poses can be determined.

In PTAM, the first frame and the last frame of the initialization sequence have to be manually specified by the user. Some implementations of monocular SLAM try to replace this manual specification with some heuristic that automatically determines when the baseline between the first camera position and the current one is sufficient. To be more robust against mostly rotational motions during initalization as well as during map extension, Gauglitz *et al.* [Gaug 12] combine visual SLAM tracking, which requires parallax-inducing camera motion for feature triangulation, with panoramic mapping and tracking. Their method is able to switch between the two modes depending on the detected camera motion and thus can take advantage of both the approaches.

The initialization of LSD-SLAM by Engel *et al.* [Enge 14] also requires a translational camera movement for the first seconds. The system is initialized by filling the depth map of the first keyframe with random values and large variance. Updating the depth map of the keyframe successively based on the estimated per-pixel disparities resulting from the translational camera movement lets the depth map converge to a correct depth configuration.

3.1.3 The Problem of Ambiguity in Scale for Augmented Reality

Having in mind the challenge to initialize monocular SLAM and the preference for a solution that works in an arbitrary environment without the need for a predefined model of the scene, this leads us to a well-known shortcoming of monocular visual SLAM. The reconstruction of the scene as well as the estimation of the camera trajectory purely from images is ambiguous in its scale.

This means that it is unknown what *absolute* distance (e.g. in meters) corresponds to a unit of the coordinate system in which the reconstructed scene model and the estimated camera poses are defined in. This scale ambiguity results from the projective nature of image capturing with a monocular camera, which means that images captured by the camera only measure a 2-dimensional projection of the 3-dimensional scene.

Figure 3.3 demonstrates this ambiguity in scale for a single image. When we capture an image of a table, a pixel captures the intensity of light incident through the aperture of the camera along a certain ray direction. No information however is captured about the distance to the pictured surface. The same image could for example also have been generated by smaller versions of the table that are

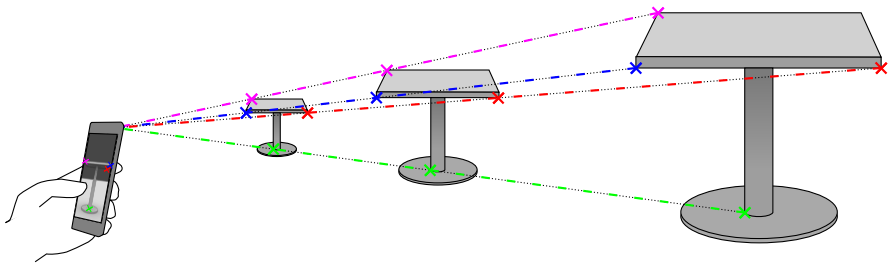


Figure 3.3: An image taken with a monocular camera captures light intensities incident through the aperture of the camera as pixel intensities. Thereby only a 2-dimensional projection of the scene is captured, that contains no information about the real dimensions. Tables of different sizes may for example lead to identical images.

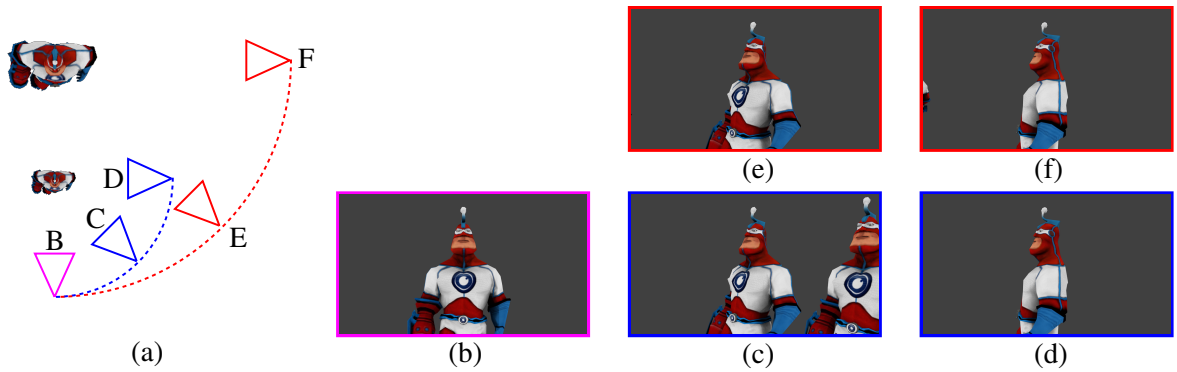


Figure 3.4: Images (b-f) captured with a monocular intensity camera (a) do not contain information about the absolute dimensions of the pictured scene. A scaled version of the scene with an equally scaled camera motion would result in identical images, as illustrated by the identical appearance of the two virtual characters of different size in images (c) and (e) as well as (d) and (f).

closer to the camera or larger versions further away. As we have no information about the distance to the surface, a single image does not even tell us if we really captured a 3-dimensional object or if we for example are just viewing a 2-dimensional image showing the table.

Besides scale, the 3-dimensional structure of the scene can however be recovered from multiple images showing the scene from different view points as we do in monocular SLAM, by exploiting correspondences between the images as shown in figure 3.2.

The remaining ambiguity in scale is demonstrated in figure 3.4. The example scene shown in figure 3.4 (a) contains two virtual characters, that are identical except for their size. One character is twice as large as the other one.

We observe the scene through a camera from multiple point of views B-F. First the camera is moved on a small circle around the smaller character covering camera positions B, C, and D while pointing at the smaller character. In a second run the camera again starts at position B, but then is moved on a circle twice as large around the larger character. While pointing at the larger character it thereby

covers camera positions B, E, and F. The larger camera motion path together with the larger character thus is just a scaled, twice as large, copy of the smaller camera motion path together with the smaller character.

The camera captures three images at the positions B, C, and D. The images are shown in figure 3.4 (b, c, d). The camera also captures three images in the second run at the camera positions B, E, and F. These images are shown in figure 3.4 (b, e, f). When we compare the corresponding images, i.e. image (c) to image (e) as well as image (d) to image (f), we see that the appearance of the smaller character in the images (c) and (d) is identical to the appearance of the larger character in the images (e) and (f). Without knowledge about the dimensions of a character or the camera motion, the absolute scale of scene and camera motion thus cannot be recovered.

Methods that simultaneously reconstruct a scene and estimate the camera poses therefore usually assign an arbitrary scale to the 3-dimensional reconstruction and camera poses.

Many applications, however, require a scene reconstruction or camera poses at absolute scale, e.g. vision-based navigation or Augmented Reality (AR) applications that superimpose virtual objects (e.g furniture) at absolute scale in a previously unknown real environment.

Our method will specifically target the use case of handheld AR. Imagine that we create an application that lets you preview how an armchair from a furniture catalogue would look like in your own living room. The dimensions of the armchair are specified by the furniture catalogue in meters (see figure 3.5 (a)) and obviously the goal is to preview the armchair in its real dimensions, as shown in figure 3.5 (b). As the living room of the user is unknown to the application, we will use monocular SLAM for camera tracking. The arbitrary scale of the reconstruction now leads to the problem that we do not know at what size we must render the virtual object that it is consistent with the real world. The arbitrary scale of the reconstruction thus results in an arbitrarily scaled superimposed virtual object as shown in figure 3.5 (c). This makes the augmented image inappropriate as a preview for how a particular object, in this case the armchair, would fit into the surroundings. We thus need to know the absolute scale of the reconstruction, so that we can render virtual objects at correct size.

Sometimes it is not necessary to recover the *absolute* scale, i.e. the relation to real-world distances, but still it is beneficial to perform SLAM at *repeatable* scale, e.g. to overcome scale drift or to obtain a consistent scale of separately mapped parts of a scene. Many of the existing monocular SLAM solutions however assign a new arbitrary scale factor with each initialization.

3.1.4 Preview on Our Approach

The method we propose to address the problem of scale ambiguity in monocular SLAM leverages the user's face as an object of known size.

We specifically target handheld Augmented Reality applications. Many handheld devices feature a camera setup which consists of a world-facing camera and a user-facing camera.

(a) (b) (c)

Figure 3.5: The dimensions of an armchair are specified in absolute units like meters (a) and the goal is to superimpose virtual objects onto the camera image at correct size (b) to provide a realistic preview. The unknown arbitrary scale of the real-world reconstruction however leads to superimposed virtual objects that are evenly arbitrarily scaled (c).

This hardware setup firstly allows us to reconstruct and track the unknown environment using images of the world-facing camera. The estimated camera motion then is used to augment the video stream of this camera with virtual objects and the augmented stream is displayed to the user.

Secondly the hardware setup allows us to simultaneously capture the user's face with the user-facing camera. Given face tracking at absolute scale, two images of a face taken from two different viewpoints enable estimating the translational distance between the two viewpoints in absolute units, such as millimeters.

Under the assumption that the face itself stayed stationary in the scene while taking the two images, the motion of the user-facing camera relative to the face can be transferred to the motion of the rigidly connected world-facing camera relative to the scene. This allows determining also the latter motion in absolute units and enables reconstructing and tracking the scene at absolute scale.

3.2 State of the Art and Related Work

Methods of monocular SLAM are widely used for 3-dimensional scene reconstruction and camera tracking in unknown environments based on a sequence of images captured by a single moving camera. The inherent ambiguity in scale as described in section 3.1.3 thereby is a well-known shortcoming. Additional information must be provided to overcome the fact, that reconstructing an unknown environment purely on images of a monocular camera is under-determined. Different ways to achieve absolute scale in monocular SLAM have been proposed in the past.

Davison *et al.* [Davi 07] propose to add stationary calibration objects of known size into the scene. From these calibration objects they determine the absolute scale of the camera motion and scene structure. As the scene and the calibration objects are both captured by the same camera, the added objects however change the appearance of the scene to be reconstructed. Additionally the user needs to have the specific calibration objects at hand, needs to *actively* position them in the scene, and needs to capture them with the camera.

Alternatively scale information can also be provided by defining the baseline between two camera poses in absolute units. Klein and Murray [Klei 07] for example manually provide the absolute distance between the two camera positions where the two images for the initial 3-dimensional triangulation are captured. The user can for example be instructed to capture a second image of the scene at a certain distance like half a meter apart from the camera location of the first image. Obviously this task is cumbersome for the user and may lead to errors.

Apart from monocular SLAM, the same idea of a known baseline however is used for rigidly connected stereo camera systems (i.e. two cameras with overlapping frusta). Here the baseline between the two cameras is fixed and can be calibrated offline as proposed by Lemaire *et al.* [Lema 07]. The two cameras of handheld devices, i.e. the world-facing camera and the user-facing camera, which we will employ for our method however have no overlapping frusta, they rather point in opposite direction.

Clipp *et al.* [Clip 08] describe how to estimate the absolute scale using a multi-camera setup with *non-overlapping* camera frusta. They also leverage the known fixed baseline between the cameras together with an additional single point correspondence within the images of the second camera. To estimate the scale they exploit the fact that rotations will induce differences in translation between the motions of the two cameras. Clipp *et al.* [Clip 08] however use their method for camera set-ups where the cameras are placed on each side of a vehicle, so that they have a large inter-camera distance of 1.9 meters. As the inter-camera distance correlates with the difference in translation induced by rotations, this approach most probably is unsuitable for handheld devices, e.g. mobile phones, where the displacement between the cameras obviously is much smaller, so that noise in the motion estimation likely will corrupt the scale estimation.

Information about the absolute scale can also be provided by cameras that are equipped with depth

sensors – sensors that measure the distance of an imaged surface from the camera in physical units, such as meters. These cameras are also referred to as RGB-D cameras, as they deliver an additional depth (D) channel. Depth information from an RGB-D camera is for example integrated into vision-based SLAM by Lieberknecht *et al.* [Lieb 11] as well as Kerl *et al.* [Kerl 13]. Such kind of sensors, however, are currently not commonly available in handheld devices. Additionally active depth cameras projecting and measuring infrared light do not work reliably outdoors during daylight.

Sensor fusion of vision with an Inertial Measurement Unit (IMU) is used by Nützi *et al.* [Nutz 11] to estimate absolute scale in monocular SLAM for moving vehicles by double integrating acceleration measurements over time yielding a position in meters.

Also Weiss and Siegwart [Weis 11] employ an IMU to recover the metric scale for camera pose estimation with a monocular camera. They mount a camera on an IMU and perform handheld movements. Similar to our approach they treat the particular visual pose estimation method as a black box, which makes their approach more versatile for different tracking approaches. The decoupling of pose estimation and scale estimation also leads to a constant computational complexity for their Extended Kalman Filter (EKF), which estimates the scale over time solely based on camera poses and corresponding uncertainties from the pose estimation method as well as acceleration and rotational velocity from the IMU. While Weiss and Siegwart [Weis 11] show good results for the estimated scale after filtering over a time of 80 seconds, such a long initialization time with continuous motion is unsuitable for our use cases, as a user wants to have the overlay of virtual objects at correct size nearly instantaneous. Additionally Weiss and Siegwart note that a good initial guess of the scale is important for their EKF approach.

Tanskanen *et al.* [Tans 13] employ inertial sensors in off-the-shelf handheld devices for estimating metric scale. Those IMUs tend to be somewhat inaccurate resulting in an error of about 10-15% in scale estimates.

For our use case in handheld AR, the main drawback of all these IMU-based approaches however is the non-negligible amount of motion required over a longer period of time (e.g. 15-30 seconds [Nutz 11, Tans 13]) to estimate the scale.

A problem that is related with the ambiguity in scale is the drift in scale over time during monocular SLAM, which leads to inconsistently scaled parts in larger reconstructions. Engel *et al.* [Enge 14] present an approach, that still tracks at an arbitrary scale but explicitly takes drift in scale into consideration in its pose graph, a global map of keyframes, which is built along with tracking the motion of the camera. In this global map, keyframes are connected by similarity transformations instead of rigid transformations. This new formulation takes scale between keyframes into account and thereby significantly enhances the performance of monocular SLAM for large scenes and scenes with large variations in scale.

Our proposed method lies in between methods relying on objects of known size, a determined baseline between two camera poses, and sensor fusion. We employ the user-facing camera of a handheld

device as an additional sensor. Even though we are using two rigidly connected cameras with non-overlapping camera frusta, we do not directly utilize baseline information between the two cameras like [Clip 08] to estimate absolute scale, as this baseline in most handheld devices is negligibly small. Instead, we capture images of the user's face and use the face as kind of a known object providing us with camera poses relative to the face at absolute scale.

By that we substitute the extra object of known size by a body part of the user, comparable to Lee and Höllerer [Lee 09] who derive scale information for their markerless tracking approach by an initial camera pose estimation from the user's outstretched hand captured by the world-facing camera. While their approach uses a single camera and requires the user to reach out, such that their hand becomes visible in the image of the camera, our approach in contrast relies on the user-facing camera in which the user's face is automatically present most of the time.

Our approach uses the camera pose relative to the user's face to induce information of the absolute scale. Face tracking algorithms work universally over almost all humans because the appearance of facial fiducials can be well approximated by a limited range of variation. Many methods for detecting facial fiducials [Cao 14] and for determining the pose of a face [Murp 09] exist. Often these methods however deliver a pose at arbitrary scale. For unknown subjects, Flores *et al.* [Flor 13] estimate the absolute distance of a face from the camera using the perspective distortion visible in the 2-dimensional images in combination with knowledge about how facial fiducials are distributed across people, learned from a small training set of exemplary 3-dimensional models of human faces. Similarly, Burgos-Artizzu *et al.* [Burg 14] estimate the distance of the camera from an unknown person, based on training in image space on a dataset of frontal portraits of 53 individuals each captured from seven distances. They also investigate which facial landmarks are suitable for the estimation.

We establish face tracking at absolute scale by combining a standard 6DoF face tracking method with knowledge about the absolute dimensions of some part of the particular face. For this purpose we employ, in our current implementation, the human interpupillary distance (IPD). The IPD of a particular user can be configured in two ways. Firstly, we can rely on statistics and simply take the mean IPD of an adult person. Dogson [Dodg 04] analyzed multiple studies regarding the IPD with the key results, that the mean adult IPD is around 63 mm, with a standard deviation around 3.8 mm, which is about 6%. The vast majority of adults has an IPD within the range of 50 mm to 75 mm. A generic face model with mean IPD thus can be used for the user introducing some degree of uncertainty. Alternatively, for improved accuracy, the dimensions of the face of a particular user, in this case the IPD, can also be calibrated.

The method we propose in this paper takes advantage of the ability to estimate camera poses relative to human faces at absolute scale. It further makes use of commonly available handheld devices comprising a world-facing camera and additionally a user-facing camera, which captures the user's face. Our method works non-intrusively, not affecting the appearance of the scene to be reconstructed, and works well even in outdoor scenarios during daylight. It neither requires a separate calibration

object, such as a marker, to be available and added to the scene, nor does it rely on dedicated sensing hardware, such as depth sensors, stereo cameras with overlapping frusta, or IMUs.

A similar handheld set-up like ours, where the user-facing camera is tracking the head of the user while the world-facing camera captures the scene, is also used in the context of user perspective AR, for example by Hill *et al.* [Hill 11] and by Grubert *et al.* [Grub 14]. Grubert *et al.* also uses the video stream from the world-facing camera to estimate the camera pose with respect to the world.

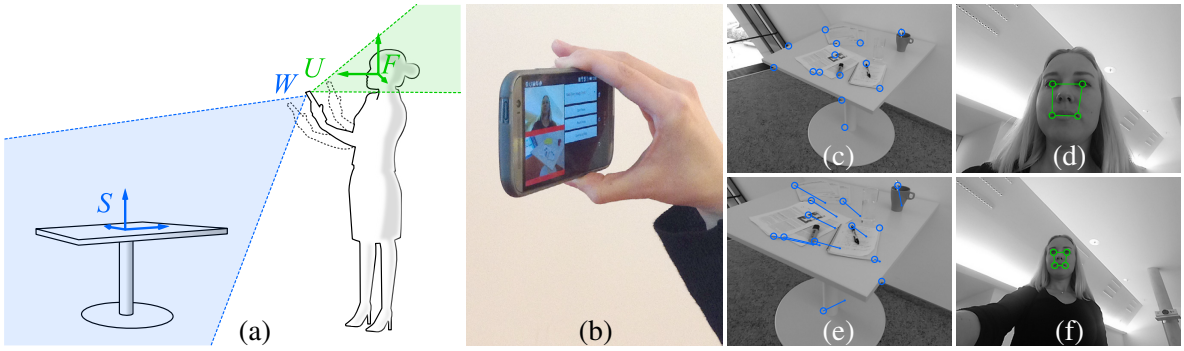


Figure 3.6: Simultaneous capturing with the world-facing camera W and the user-facing camera U (a) of a mobile phone (b) delivers a sequence of image pairs of the scene S (c,e) and the face F (d,f).

3.3 Approach

In order to enable monocular SLAM at absolute scale, our proposed method requires a handheld device comprising a world-facing camera and a user-facing camera as shown in figure 3.6 (a).

Instead of adding a marker of known size to the scene and capturing both – scene and marker – with the world-facing camera, we propose to use the user's face as scale reference, which is usually visible in the user-facing camera. Taking advantage thereof renders any instrumentation of the scene unnecessary.

The absolute dimensions of the user's face can be calibrated once as described in section 3.5.2.3 and can then be re-used subsequently. If no calibration data is available, a generic average face model can be selected instead as a fallback since facial dimensions, such as the IPD, vary only moderately among different adult humans [Dodg 04].

With the face model defined at absolute scale, the pose of the user-facing camera relative to the face can be determined at absolute scale in real time based on the image of the user-facing camera by means of a 6DoF face tracking method [Murp 09].

Since the world-facing camera and the user-facing camera are rigidly connected to each other and the 6DoF transformation in between them is at least approximately known, also the pose of the world-facing camera can be determined at absolute scale relative to the face, as derived in section 3.3.1. Under the assumption that the face has not moved relative to the scene over a period of time, the meanwhile determined absolute poses relative to the face are also valid relative to the scene, which enables reconstructing the scene at absolute scale. Analogously it enables transforming an existing reconstruction of the scene from arbitrary scale to absolute scale.

We introduce the following notation. At a time t the user-facing camera captures the image $U(t)$ and the world-facing camera captures the image $W(t)$. The pose of the user-facing camera relative to the coordinate system of the user's face is referred to as $\mathbf{U}_F(t)$, see figure 3.7 (a). The pose of

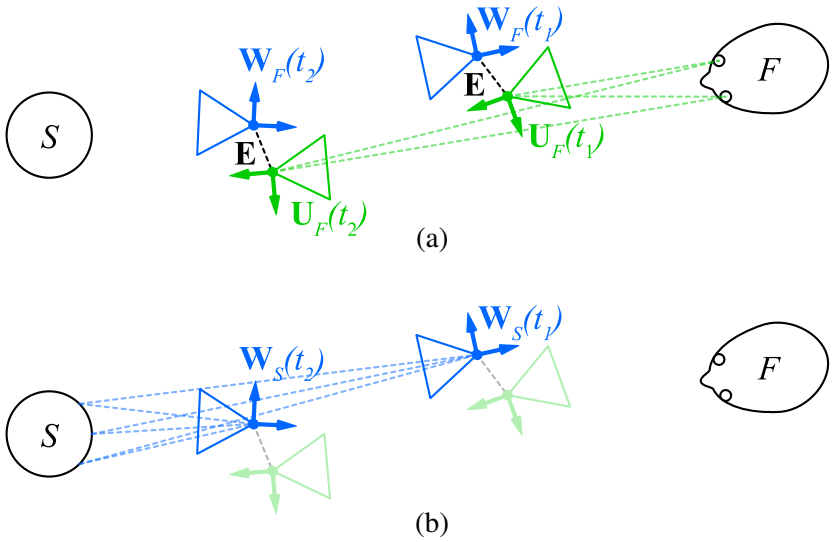


Figure 3.7: Face tracking (a) allows to determine poses relative to the face at absolute scale for the user-facing and the world-facing camera, which then (b) can be used to transform poses from monocular SLAM relative to the scene from arbitrary to absolute scale.

the world-facing camera in the coordinate system of a reconstruction of the scene at arbitrary scale is called $\mathbf{W}_S(t)$ (figure 3.7 (b)) while the pose of this camera in the coordinate system of the user's face is referred to as $\mathbf{W}_F(t)$ (figure 3.7 (a)).

3.3.1 Absolute Scale from two Keyframes

At a first keyframe t_1, we store the image $W(t_1)$ of the world-facing camera (figure 3.6 (c)), and the corresponding image $U(t_1)$ of the user-facing camera (figure 3.6 (d)). After moving the camera to a different viewpoint, a second keyframe t_2 with images $W(t_2)$ (figure 3.6 (e)) and $U(t_2)$ (figure 3.6 (f)) is stored. We then use image $U(t_1)$ to determine pose $\mathbf{U}_F(t_1)$ and image $U(t_2)$ to determine pose $\mathbf{U}_F(t_2)$ using a face tracking method at absolute scale (figure 3.7 (a)). Given the extrinsic rigid body transformation $\mathbf{E}$ (see section 3.4.1) between the user-facing and the world-facing camera we can determine $\mathbf{W}_F(t_1)$ as $\mathbf{E}\mathbf{U}_F(t_1)$ and $\mathbf{W}_F(t_2)$ as $\mathbf{E}\mathbf{U}_F(t_2)$.

Under the assumption that the face stayed stationary in the scene, the poses $\mathbf{W}_F(t_1)$ and $\mathbf{W}_F(t_2)$ are valid relative to the scene, which enables reconstructing the scene at absolute scale using triangulation. It further allows to compute the scale factor a from the arbitrary units of the coordinate system of an up-to-scale model of the scene S to the absolute units of the coordinate system of the user's face F as

$$a = \frac{\|\tau(\mathbf{W}_F(t_1)) - \tau(\mathbf{W}_F(t_2))\|}{\|\tau(\mathbf{W}_S(t_1)) - \tau(\mathbf{W}_S(t_2))\|} \tag{3.1}$$

where the operator τ extracts the translation vector of a pose and $\mathbf{W}_S(t_1)$ and $\mathbf{W}_S(t_2)$ are determined using visual camera localization relative to the model of the scene at arbitrary scale (figure 3.7 (b)).

3.3.2 Absolute Scale from Multiple Keyframes

While in theory two keyframes t_1 and t_2 suffice, a longer sequence of keyframes of user-facing and world-facing camera images lets us compute one scale factor a_i for each pair of keyframes (t_i, t_j), which gives us a set of scale factors. Some of them are more accurate than others. It is important to keep in mind that the calculation of the factor only works reliably, if the keyframes t_i and t_j differ sufficiently in terms of translation of the cameras. To determine a more reliable and robust overall scale factor based on more than two keyframes we randomly select from the sequence a set of N pairs of keyframes (each keyframe comprising of an image of the user's face and an image of the scene) such that the user-facing camera moved at least a distance of d_{min} between the two keyframes of each pair and compute a scale factor a_i per pair of keyframes using equation (3.1). Finally we compute a factor $\widetilde{a} = \text{Median}(A)$ of the set of all scale factors $A = \{a_1, a_2, ..., a_N\}$ and use $\widetilde{a}$ to scale the reconstruction.

Figure 3.8 plots the distribution of scale factor estimates a_i and the median $\widetilde{a}$ for an example sequence. In section 3.5 we quantitatively evaluate how accurate our proposed method estimates the absolute scale of a scene based on the median scale estimate over a sequence.

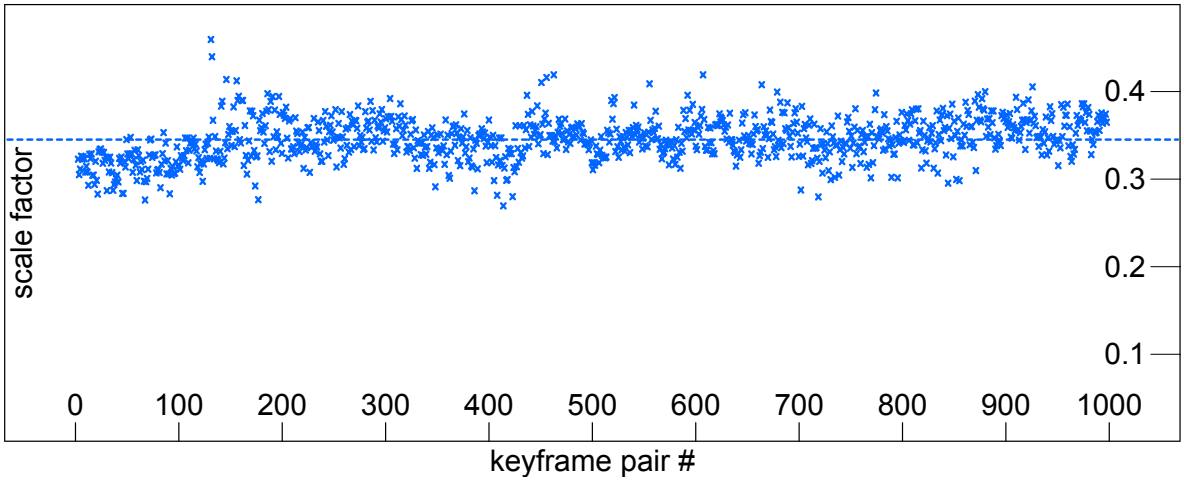


Figure 3.8: Distribution and median of estimated scale factors for a set of 1000 pairs of keyframes capturing a scene and a face.

3.4 Implementation

To proof our proposed method, we implemented it based on an HTC One M8 mobile phone (figure 3.1), which allows simultaneous capture from the world-facing camera and the user-facing camera. We use a resolution of (640×480) pixels for both cameras and determine intrinsic parameters of each camera using images of a checkerboard [Zhan 00].

3.4.1 Extrinsic Inter-Camera Calibration

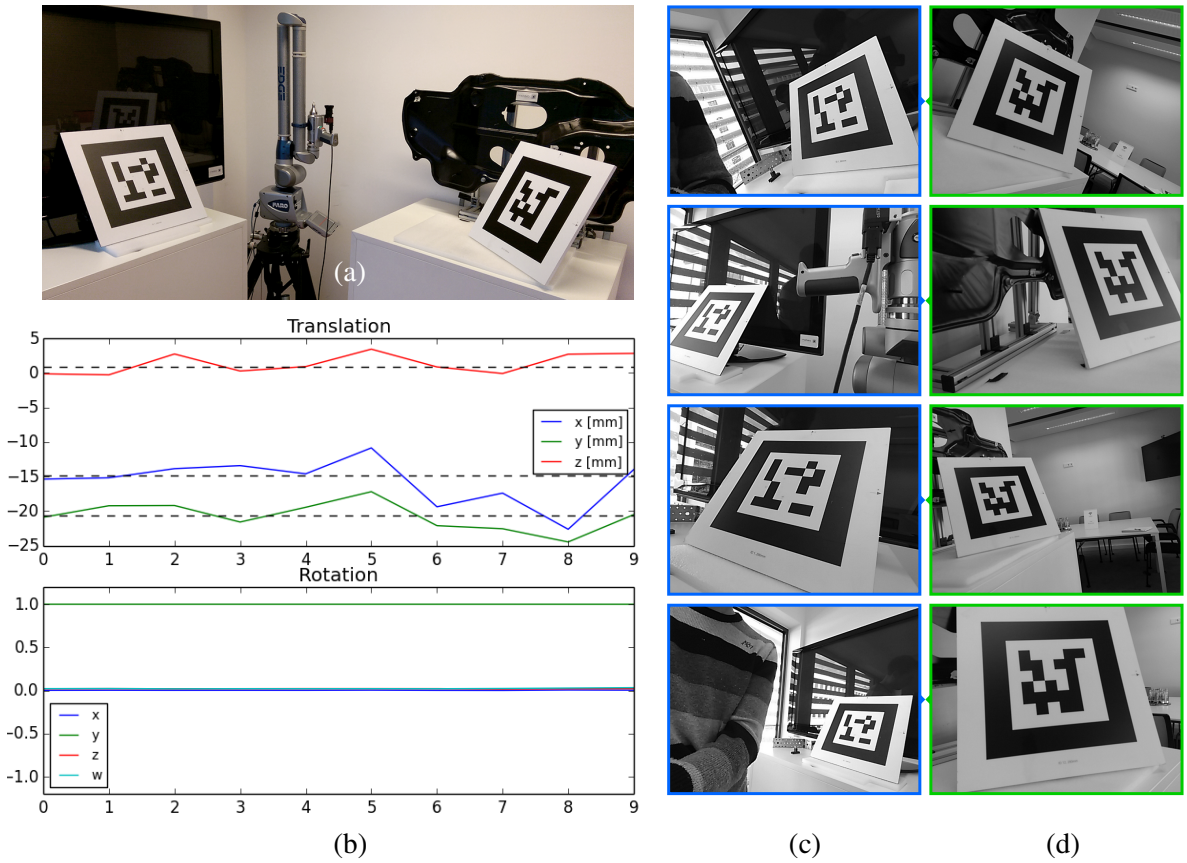


Figure 3.9: The marker setup (a) was calibrated with the help of a mechanical measurement arm. We then determined the extrinsic parameters (b) between the two cameras using a set of image pairs (c,d).

The user-facing camera on the front and the world-facing camera on the back of a handheld device are not located at the exact same spot. For evaluation purposes we calibrated the extrinsic parameters $\mathbf{E}$, i.e. translation and rotation, between the two cameras of the employed phone.

For that we first accurately determined the positions and rotations of two markers (figure 3.9 (a)) by touching their respective corners with the tip of a mechanical measurement arm. We then moved

the mobile phone between the two markers such that the user-facing camera captures the first marker (figure 3.9 (c)) while the world-facing camera sees the second marker (figure 3.9 (d)). For each image pair, the camera poses were determined in a common coordinate system based on marker tracking, and the resulting 6DoF transformation between the two cameras was computed (figure 3.9 (b)). Finally we computed the median of the coordinates of the translation vector and the rotation expressed as quaternion to determine the extrinsic parameters $\mathbf{E}$ transforming from the coordinate system of the user-facing camera to world-facing camera coordinates.

The results of the extrinsic calibration show that the two cameras are facing exactly in opposite direction. The translational offset between the cameras is nearly orthogonal to their optical axes, with a length of 26 mm.

3.4.2 Offline Evaluation

Our experiments – both for evaluation and real-time applications – are based on a proprietary monocular SLAM system from the Metaio SDK [Meta 15]. The SLAM system is capable of running in real time on the mobile phone mapping a real scene and tracking the pose $\mathbf{W}_S(t)$ of the world-facing camera relative to it at arbitrary scale. Besides poses, the SLAM system provides the 3-dimensional coordinates of the reconstructed points.

We use two approaches in our evaluations for determining the pose $\mathbf{U}_F(t)$ of the user-facing camera relative to the user's face. To simulate perfect 6DoF face tracking at absolute scale we place, in section 3.5.1, a square marker where the user's face would usually be and track it using the marker tracking framework of the Metaio SDK. In section 3.5.2 we further use a proprietary face tracking method which provides the 6DoF pose of a camera relative to a face given an image of it. This method is based on a generic face model which can be adjusted by one parameter, the IPD, to account for the faces of different users.

For the quantitative evaluations in section 3.5 we merely use the phone as a capturing device. A custom app allows to store synchronized video sequences of the world-facing camera and the user-facing camera to files at a framerate of $\sim$30 Hz. All the further processing then is performed offline on a PC. This enables repeatable evaluations and systematically testing the impact of different parameters on the estimated absolute scale.

3.4.3 Real-Time Applications

In addition to the quantitative offline evaluations we present in section 3.6 different applications that are enabled by our proposed method. These applications run in real time on the mobile phone without any additional PC. The deployed SLAM system is the same as in the offline evaluation, while we use a mobile-specific proprietary method for the face tracking. In the real-time application the scale is

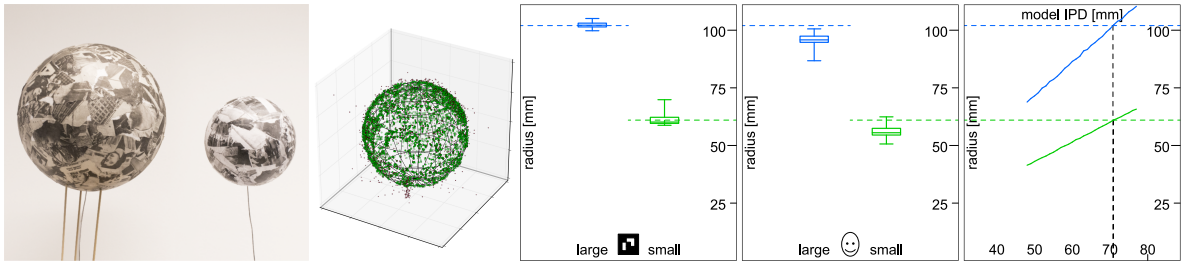
estimated using equation (3.1) for the first and last keyframe of the sequence instead of the median over the whole sequence.

3.4.4 Stationary Face Assumption

Our current implementation assumes that the user's face remains static with respect to the environment during the scale estimation.

In the real-time application, the user at the moment manually triggers the scale estimation procedure by holding down a button at the lower left of the user interface, see figure 3.12 (a). The first keyframe for the scale estimation is taken when the user presses the button down, the second keyframe is taken as soon as the user releases the button. By performing a motion with the smartphone towards or away from the face, the face stays intuitively stationary. Before and after the scale estimation procedure, which roughly takes a second, the user can again freely move around.

If the user's face does not remain stationary in the scene during the procedure, the estimate will be inaccurate. The relative error in measuring the distance traveled by the user-facing camera relative to the world falsifies the estimated scale factor a proportionally.



(a) (b) (c) (d) (e)

Figure 3.10: We create multiple reconstructions of two spherical scenes (a) with different known radiuses. To each reconstruction at arbitray scale we fit a virtual sphere and determine its radius (b). When then apply our proposed scale estimation method to each reconstruction and by that bring each radius to absolute scale. The distribution of estimated radiuses at absolute scale (mm) is shown for an idealized case using marker tracking (c) and for the real-world case using face tracking (d) in comparison with ground truth. The influence of inaccurate interpupillary distance calibration is plotted in subfigure (e) for both scenes.

3.5 Evaluation

We quantitatively evaluate the accuracy and precision in estimated scale achieved by our method in order to assess which use cases it enables. We compare the dimensions of the reconstructed maps of a scene, which we brought to absolute scale using our method, against ground truth information about the dimensions of the scene. The scenes we use in our evaluation are spherical because this allows for an easy and reliable evaluation against ground truth. Our proposed method itself supports scenes of any shape.

We use two styrofoam spheres (figure 3.10 (a)) at two different sizes pasted up with newspaper. The large sphere has a radius of 102 mm, while the small sphere has a radius of 61 mm. By moving the world-facing camera around the respective sphere, we obtain a 3-dimensional reconstruction using monocular SLAM at arbitrary scale. We then track the sphere based on this reconstruction and in parallel track the user's face in the user-facing camera. This enables us to determine the scale factor $\widetilde{a}$ (see section 3.3.2 with $N = 1000$ and $d_{min} = 120$ mm) between the arbitrary scale of the reconstruction and the real dimensions of the scene in millimeters.

The device in these sequences is moved mainly along the optical axes of the cameras as illustrated in figure 3.7. This movement makes it more convenient for the user to not move the head relative to the scene as opposed to sideways motions. The predominant translational motion also reduces the influence of the transformation $\mathbf{E}$ between user-facing and world-facing camera on the covered distances of the respective cameras.

To measure the accuracy of the scale estimation, we fit a virtual sphere to each set of reconstructed 3-dimensional points (figure 3.10 (b)), finding the best sphere using random sample consensus (RANSAC) [Fisc 81]. We scale the fitted radius by the estimated scale factor and compare it

against the ground truth radius. To evaluate on as much data as possible, we created for each of the two real spheres six reconstructions using SLAM. The radiuses of these arbitrarily scaled reconstructions vary between 95.9 and 718.2 for the large styro sphere and between 68.6 and 454.5 for the small styro sphere. Additionally we captured ten sequences of a few seconds each with the respective sphere being tracked with the world-facing camera while the user-facing camera captures a face or marker. All combinations of reconstructions and sequences result in 60 radius estimates per sphere.

3.5.1 Under Perfect Conditions – Marker Tracking

To get an idea of the accuracy and precision achievable under perfect conditions, i.e. without any motion between face and scene and with very accurate 6DoF face tracking, we first replace the user's face with a square marker on a tripod at the position where the user's face would usually be and use 6DoF marker tracking instead of face tracking.

3.5.1.1 Results

The resulting estimated radiuses for the two spherical scenes based on all combinations of six reconstructed maps and ten camera sequences are plotted in figure 3.10 (c). For each scene, a candlestick chart shows minimum, first quartile, median, third quartile, and maximum value of the estimated radiuses. A dashed horizontal line shows the ground truth radius for the reader's reference. We see that in this configuration our method achieves to estimate the radiuses of the two spheres with both high accuracy and precision. The median of 102.17 mm over all estimates for the large styro sphere corresponds to a relative error of 0.16 % (equivalent to 0.17 mm) with a standard deviation of 1.20 mm over all estimates. For the small sphere the median of 60.17 mm corresponds to a relative error of 1.36 %, (equivalent to 0.83 mm) with a standard deviation of 2.59 mm over all estimates.

3.5.1.2 Influence of the Extrinsic Inter-Camera Calibration

For all the estimations evaluated above as well as plotted in figure 3.10 (c) we considered the extrinsic rigid body transformation $\mathbf{E}$ between the user-facing and the world-facing camera, determined in section 3.4.1. $\mathbf{E}$ can be assumed to vary between different handheld devices, and potentially there is not always a calibration available. We therefore separately run the estimations on the same sequences *ignoring* the extrinsic calibration, i.e. using a generic extrinsic calibration $\mathbf{E}$ assuming that the two cameras are located exactly at the same position. Note that the rotation between the two cameras is irrelevant for the distances used in the scale calculation.

The simplification of ignoring the extrinsic rigid body transformation $\mathbf{E}$ only slightly affects the results with a median of 102.87 mm for the large and 60.49 mm for the small styro sphere with comparable standard deviations. This negligible influence of $\mathbf{E}$ has multiple reasons.

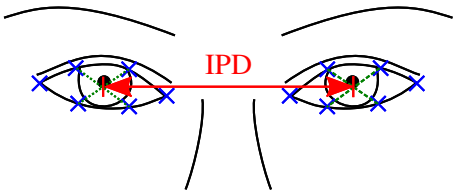


Figure 3.11: We determine the IPD (depicted in red) in a captured image, by using the average position (green) of four detected fiducials around the eye contour (depicted in blue) instead of the pupils in order to be invariant against eye convergence.

First of all, the scale estimation using equation (3.1) only considers the covered distance between two poses. This distance is not affected by the rotational part of **E**, which expresses the difference in viewing directions between the two cameras.

The translational part of **E**, which is the baseline between the two cameras, however has an impact on the distances covered by the cameras when the smartphone is moved. As long as the smartphone is moved purely translational, both cameras cover the same distance. A rotation of the smartphone around some axis however induces a translation for all the points of the smartphone, that do not lie on this axis of rotation, e.g. the positions of the cameras. The induced translation thereby varies for different points. The length is directly proportional to the perpendicular distance of a point from the rotation axis. The additional translation falsifies the scale estimation, when we do not compensate for the extrinsic rigid body transformation **E**. As in our sequences the smartphone however is moved mainly translational, and the baseline between the cameras is small with less than 3 cm compared to the covered distance between time t_1 and t_2, the impact of the extrinsic rigid body transformation **E** becomes negligible.

The results demonstrate, that our method works well even without an extrinsic calibration for a particular device.

3.5.2 Under Realistic Conditions – Face Tracking

We then evaluate our method using face tracking instead of marker tracking. To enable face tracking at absolute scale we provide the IPD of the particular person to the face tracking method. Note, that in our implementation we always refer to the distant IPD, which is the IPD when the person is focusing at infinity. As eyes turn inward (converge) to focus on closer objects, the IPD changes. We thus do not track the pupils directly but rely on fiducials on the eye contour, as depicted in figure 3.11.

3.5.2.1 With Calibrated Interpupillary Distance

For this part of the evaluation the IPD of the user has been calibrated manually using a ruler and a mirror. During capturing the sequences used for the scale estimation the user tried to avoid moving their head but we can assume that small motions occurred.

The distribution of radiuses of reconstructed spheres in 60 runs per scene are plotted in figure 3.10 (d). We observe that in this case estimations are less accurate and the radiuses, and hence the scale of the scene, are mostly underestimated.

The median of 95.81 mm over all estimates for the large styro sphere corresponds to a relative error of 6.07 % (equivalent to 6.19 mm), the median of 55.65 mm over all estimates for the small styro sphere corresponds to a relative error of 8.78 % (equivalent to 5.35 mm). With a standard deviation of 2.35 mm for the large styro sphere and 2.60 mm for the small one, the estimates however are only slightly less precise than those obtained with marker tracking.

3.5.2.2 Influence of the Interpupillary Distance

We use the IPD to enable face tracking at absolute scale. If for a user the exact IPD is not available the mean IPD of an adult person, which is about 63 mm [Dodg 04], could be assumed. Hence we evaluate the impact of an inaccurately calibrated IPD on the absolute scale estimate. Therefore we estimate the radiuses of the two spheres based on sequences of a user with an IPD of 68 mm while configuring the face tracking method to use an IPD between 48 mm and 77 mm in steps of 1 mm which covers the vast majority of adults [Dodg 04].

Figure 3.10 (e) plots the radius of the reconstruction at absolute scale as a function of the assumed IPD. We observe a linear dependency between the two parameters. The introduced percental error in scale estimation is linearly coupled to the error between real and assumed IPD. Statistically the potential lack of accuracy from relying on statistics for this distance instead of calibrating it for a user follows the same distribution as the IPD. According to the Ansur database mentioned in [5], the IPD (age 17 to 51) follows a normal distribution with mean 63.4 mm and standard deviation of 3.8 mm, corresponding to $< 6\,\%$.

Interestingly the most accurate reconstructions were achieved with an IPD 71 mm for both the large and the small styro sphere, while the manually measured IPD for the subject is 68 mm. This suggests some bias in our applied face tracking method.

3.5.2.3 Per User Calibration of the Interpupillary Distance

The IPD may be calibrated manually using e.g. a ruler. People with glasses may also already have their IPD measured by an optician.

Beside that, a semiautomatic calibration of the IPD or other facial features can be performed using the dual camera setup presented in here by inverting the scale transfer. Camera motion can be estimated at absolute scale by means of e.g. marker tracking using the world-facing camera. By simultaneously tracking the facial features to be calibrated on the user-facing camera, the absolute scale can be transferred to the facial features as long as the user's face stays static with respect to the marker.

We implemented a prototype of this semi automatic approach and performed 8 calibration runs for a person with an IPD of 68 mm. The resulting estimates for the IPD had a mean value of 68.9 mm with a standard deviation of 2.3 % (corresponding to 1.6 mm). The deviation towards an overestimated IPD is to a certain extent consistent with our observation of the underestimated sizes for the small and large styro spheres based on face tracking in section 3.5.2.1. Enhancements with regard to the face tracking method could in future eliminate this bias and additionally lower the standard deviation for improved accuracy and precision in IPD calibration as well as scale estimation.

The calibration procedure for a particular user only needs to be performed once. If multiple users share a device, visual face recognition could be employed to select the stored calibration corresponding to the particular user from the set of available calibrations.

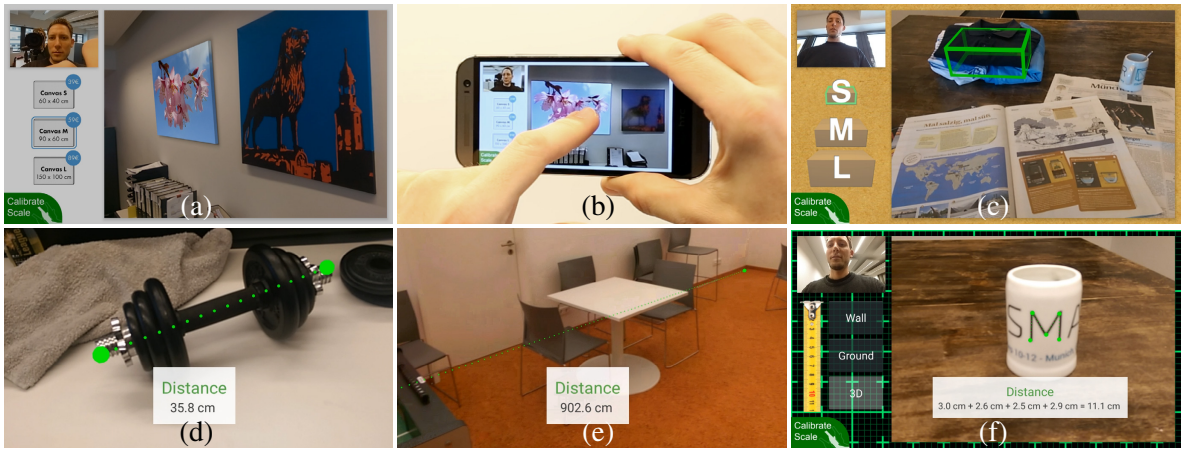


Figure 3.12: Examples of the various applications which our proposed method enables. Performing SLAM at absolute scale enables superimposing virtual objects at absolute scale (a-c) as well as measuring distances at absolute scale, e.g. in centimeters (d-f).

3.6 Applications

Our proposed method enables a variety of handheld AR applications, which require camera pose estimation or mapping of a real environment at absolute scale, e.g. superimposing virtual objects at absolute scale, or interactive distance measurements in the scene.

3.6.1 Superimposition at Absolute Scale

When virtual objects in AR act as a substitute of a real object, it is beneficial to superimpose them at absolute scale. Common examples include virtually placing a piece of furniture, e.g. an armchair (figure 3.5), in the living room to test if it would fit in the room and how it matches with the remaining (real) furniture. Here it is crucial that the armchair is superimposed at correct size (b), compared to a superimposition at arbitrary scale (a), which provides a wrong visual feedback to the user about the potential real appearance.

Virtually placing canvas prints on the wall to pick the right size out of the available selection (figure 3.12 (a,b)) is another example when superimposition of virtual 3-dimensional objects requires absolute scale.

Absolute scale is also needed when not visual appearance within the surroundings but the physical dimensions themselves matter, see e.g. figure 3.12 (c) where the size of a parcel is visualized so that the user can visually decide which parcel size is needed for a particular shipment.

3.6.2 Measurements at Absolute Scale

Additionally to augmentations of virtual objects at correct size, the reconstruction of a scene at absolute scale enables measuring distances within the scene.

Conventionally, different tools are needed for measuring depending on the use case and scale, from rulers to tape measures. It is even harder to measure when either the direct connection between measurement points is not possible e.g. the length of a wall occupied in between by furniture or when measurement points are visible but out of reach.

Our proposed method enables using the same measurement tool – the smartphone – to perform all these measurements – from a few millimeters to many meters – in a convenient non-contact manner. By simply clicking on the touch screen which shows the scene captured by the world-facing camera, the corresponding 3-dimensional locations in the scene are selected and the distance between successively selected scene locations is provided to the user. This allows a convenient way to perform measurements for a large variety of settings and objects. It for example makes it possible to measure a longer path (figure 3.12 (e)) through a building telling you the needed length for a cable, as well as measuring the total length over multiple path segments (figure 3.12 (f)). It further allows to measure the linear distance between 3-dimensional points (figure 3.12 (d)) that cannot be directly measured with a ruler or a measuring tape.

Depending on the particular use case, scene locations can be selected on a plane aligned to either the ground or a wall or on an arbitrary object in 3-dimensional space.

For selecting locations on a plane we project a touch position in the camera image onto the respective plane and thus obtain the 3-dimensional coordinates of the corresponding scene point. For general 3-dimensional scenes, we project all 3-dimensional features of the reconstructed SLAM map into the camera coordinate system and select the feature which projects closest to the touch position.

We evaluate the accuracy of distance measurements performed with the tool explained above. Note that this evaluation includes the accuracy of the SLAM system and the user's ability to select points on the screen. Results are listed in table 3.1. Except for the outcome for the envelope, the achieved measurements have a relative error below 7 % over the whole range of small scale measurements of a stamp up to large scale measurements of a whole room. While the achieved accuracy is sufficient for many use cases we plan to further improve it in the future.

Table 3.1: Measurement results in comparison with ground truth distances.

	Distances (cm)		
Object	GT	Measurement	Relative Error (%)
Stamp (diag.)	3.6	3.8	5.6
Envelope (diag.)	24.4	27.3	11.9
Book (diag.)	30.0	31.7	5.7
Barbell	34.7	35.8	3.2
Newspaper (diag.)	69.5	72.3	4.0
Table	122.6	131.0	6.8
Room	898.0	902.6	0.5

3.7 Conclusions and Future Work

In this part of the thesis, we presented the first approach to take advantage of the world-facing and user-facing camera in current handheld devices to estimate absolute scale in handheld monocular SLAM. In combination with leveraging the face of the user as trackable object of known size this brings multiple benefits over common approaches. It supersedes the need to place an additional marker or object of known size into the scene and is non-intrusive to the scene to be reconstructed. Our method enables a variety of AR applications from displaying virtual objects superimposed onto a scene at the correct size (figure 3.12 (a-c)) to distance measurements (figure 3.12 (d-f)).

Our experiments showed for different scenes, that scale could be estimated with a relative median error of less than 9 % which outperforms the IMU based approach by Tanskanen *et al.* [Tans 13] who report an error of 10-15 %. However direct comparison to alternative approaches is hard to achieve. In order to estimate scale, the IMU based approach requires stronger movements of the camera over a longer period of time of about 30 seconds, while the scale estimation in our implementation can be performed within a second using only a simple translational motion. Our implementation on the other hand would fail at the moment if the head is not kept static and delivers a higher error when face dimensions are not calibrated.

Our method is largely independent of the particular employed systems for monocular SLAM and face tracking as it uses both as black boxes that provide poses. It however depends on the quality of the poses and hence will immediately benefit from any improvements in both the SLAM system or the face tracker in terms of precision, accuracy and robustness. Potential for high accuracy has been demonstrated in section 3.5.1 where we substituted face tracking with marker tracking and achieved a median relative error <1.4 %.

We showed that using a generic IPD still results in reasonable estimates which are slightly inaccurate but still precise, i.e. repeatable. This allows to map parts of a larger scene separately at a consistent scale. For several applications, e.g. playing (augmented) games, approximate information on the absolute scale of a scene may suffice.

Our method requires the user's face to be stationary in the scene during the scale estimation, which in practice takes about a second. In our evaluations, the user was instructed to not move their face. In future work, we will look into methods to automatically determine when the face did not move relative to the scene for a set of keyframes and then automatically perform (re-)estimation of the absolute scale as a background process of a SLAM system. Continuous scale estimates can not only be combined into a more robust scale factor but also prevent scale drift – an important problem to address in monocular SLAM.

Evaluating if the face remained stationary could be done based on a similarity transformation [Umey 91] between the two camera trajectories from SLAM and face tracking (considering the extrinsic parameters **E**). This would deliver the wanted scale factor, and remaining inconsistencies would indicate a motion of the face. Motions of the face could also be identified by transforming epipolar constraints from one camera to the other and evaluating if the constraints hold for the moving features of the respective tracked target. For both these approaches however there remain certain motions of the face that cannot be identified, which we plan to address in future work.

4 Conclusion

This thesis proposed to employ the image of the user's face captured by a user-facing camera in order to deduce information about the real world – information that is needed for plausible augmentations consistent with the real world. By relying on the fact that faces from different people exhibit a lot of similarities in appearance, we leveraged the face as a known object. We showed in particular how this idea can be implemented to reconstruct the real world in terms of illumination and absolute scale, two topics that especially in the domain of Augmented Reality on handheld devices are not completely solved until now. We believe that our idea to focus on the image of the user's face will also inspire future research in this area and thereby will further advance the realism in consumer Augmented Reality applications.

Seamlessly integrating virtual content into the view of the real world requires information about the real-world environment. If this information is not yet given, the environment must be reconstructed on-the-fly. Throughout this thesis we addressed two different challenges existing in the context of scene reconstruction for Augmented Reality (AR) applications, namely a coherent illumination of virtual objects that matches the illumination present in the real world as well as the reconstruction of an unknown environment at absolute scale for augmenting virtual objects at correct size with regard to the real world.

We already presented specific conclusions and future work for both areas of research at the end of their respective chapters, namely in section 2.8 (Illumination Estimation) and section 3.7 (Scale Estimation). In this last part, we place emphasis on what both proposed approaches have in common and by that highlight the broader idea and potential we see in employing the user-facing camera and the image of the user's face for deducing information about the real world.

In order to be successful, AR applications targeting the mass market need to consider which hardware components and processing power is available to the consumers. As smart phones became

companions of our daily life these devices represent the ideal tool for enabling ubiquitous AR applications for everybody. Hardware requirements for handheld AR applications thus should be tailored to the existing devices. Besides low hardware requirements, the used methods need to be *simple and fast* to perform, and should not depend on additional tools.

In both our approaches we proposed to employ the face of the user as a known object. The face of the user which is already part of the scene can conveniently be captured by the user-facing camera of current mobile devices. We employed the face of the user as a known object, firstly by leveraging known reflectance properties to estimate the incident illumination, and secondly by leveraging the known size of the user's face to estimate the dimensions of the environment. Our presented solutions hereby were able to overcome limitations of state-of-the-art methods, which either require special hardware like depth cameras or special known objects like e.g. markers or mirror spheres.

For both our presented approaches we demonstrated that the limited variations over different humans make it possible to rely on *pre-learned* knowledge, either in terms of Radiance Transfer Functions or in terms of spatial dimensions. By taking advantage of this additional knowledge we were able to eliminate the need for acquiring geometry or absolute depth during run-time using special depth sensors. Instead our approaches run on simple monocular intensity cameras, so that they are *ready for use* on most of the commonly available mobile devices.

Utilizing pre-learned knowledge thereby does not only reduce hardware requirements with respect to sensors, but it also allows to shift expensive calculations, e.g. the computation of Radiance Transfer Functions, from run-time into an offline process. This allows us to improve the *processing time* as well as *power consumption*, which is especially crucial for *mobile* AR.

Our approaches also meet the second key requirement for methods targeting nonprofessional users of handheld AR: ease of use. The light estimation can be performed completely unnoticed to the user, the scale estimation procedure only requires a short and straight-forward user interaction.

For both the presented topics we first provided the context as well as the related work in this direction. We then derived our particular idea and presented a working implementation thereof. Our subsequent evaluations of our implementations have proven the effectiveness of our approaches.

Still both the presented methods must be seen as proof-of-concept implementations, where our focus was on demonstrating the idea and feasibility. Both implementations were realized in a quite straight-forward manner, and we already pointed out different directions for further improvements in the respective chapters. Throughout the derivations of the algorithms we put an emphasis on bringing up the various assumptions that we made, so that future work can specifically address resulting limitations.

We believe that our proposal to employ the user-facing camera and the image of the user's face for deducing information about the real world revealed its potential and will give rise for future work in this domain.

Abbreviations

6DoF	Six Degrees of Freedom
AR	Augmented Reality
ARVIDA	Angewandte Referenzarchitektur für virtuelle Dienste und Anwendungen
BMBF	Bundesministerium für Bildung und Forschung
BRDF	Bidirectional Reflectance Distribution Function
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
FOV	Field Of View
GT	Ground Truth
HDR	High Dynamic Range
IMU	Inertial Measurement Unit
IPD	Interpupillary Distance
ISMAR	International Symposium on Mixed and Augmented Reality
Lat-Long	Latitude-Longitude
RANSAC	Random Sample Consensus
RGB	Red, Green, and Blue; usually refers to color channels of an image
RGB-D	Red, Green, and Blue with additional Depth; usually refers to channels of an image
RTF	Radiance Transfer Function
SDK	Software Development Kit
SfM	Structure from Motion
SH	Spherical Harmonics
SLAM	Simultaneous Localization And Mapping
VO	Visual Odometry

Bibliography

[Aitt 10] M. Aittala. "Inverse lighting and photorealistic rendering for augmented reality". *The Visual Computer*, Vol. 26, No. 6-8, pp. 669–678, 2010.

[Ange 08] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. "Fast and incremental method for loop-closure detection using bags of visual words". *Trans. on Robotics (T-RO)*, Vol. 24, No. 5, pp. 1027–1037, 2008.

[Arie 12] I. Arief, S. McCallum, and J. Y. Hardeberg. "Realtime Estimation of Illumination Direction for Augmented Reality on Mobile Devices". In: *Proc. Color and Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications (CIC)*, 2012.

[Asth 14] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. "Incremental Face Alignment in the Wild". In: *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Atke 97] C. G. Atkeson, A. W. Moore, and S. Schaal. "Locally Weighted Learning". *Artificial Intelligence Review*, Vol. 11, No. 1, pp. 11–73, 1997.

[Barr 13] J. Barron and J. Malik. "Shape, Illumination, and Reflectance from Shading". Tech. Rep., Berkeley Tech Report, 2013.

[Barr 78] H. G. Barrow and J. M. Tenenbaum. "Recovering intrinsic scene characteristics from images". *Computer Vision Systems*, pp. 3–26, 1978.

[Basr 03] R. Basri and D. W. Jacobs. "Lambertian Reflectance and Linear Subspaces". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 25, No. 2, pp. 218–233, 2003.

[Blan 99] V. Blanz and T. Vetter. "A Morphable Model For The Synthesis Of 3D Faces". In: *Proc. SIGGRAPH*, 1999.

[Boom 13] B. Boom, S. Orts-Escolano, X. Ning, S. McDonagh, P. Sandilands, and R. Fisher. "Point Light Source Estimation based on Scenes Recorded by a RGB-D camera". In: *Proc. British Machine Vision Conference (BMVC)*, 2013.

[Brac 99] R. N. Bracewell. *The Fourier Transform and its Applications*. *McGraw-Hill Series in Electrical and Computer Engineering*, McGraw-Hill, 3 Ed., 1999.

[Burg 14] X. P. Burgos-Artizzu, M. R. Ronchi, and P. Perona. "Distance Estimation of an Unknown Person from a Portrait". In: *Proc. European Conference on Computer Vision (ECCV)*, 2014.

[Bute 15] P.-E. Buteau and H. Saito. "[POSTER] Retrieving Lights Positions Using Plane Segmentation with Diffuse Illumination Reinforced with Specular Component". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR) - Posters*, 2015.

[Cali 13] D. A. Calian, K. Mitchell, D. Nowrouzezahrai, and J. Kautz. "The Shading Probe: Fast Appearance Acquisition for Mobile AR". In: *Proc. SIGGRAPH Asia- Technical Briefs*, 2013.

[Calo 10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "Brief: Binary robust independent elementary features". In: *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[Cao 14] X. Cao, Y. Wei, F. Wen, and J. Sun. "Face alignment by explicit shape regression". *Int. Journal of Computer Vision (IJCV)*, Vol. 107, No. 2, pp. 177–190, 2014.

[Clem 07] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós. "Mapping Large Loops with a Single Hand-Held Camera". In: *Proc. Robotics: Science and Systems (RSS)*, 2007.

[Clip 08] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley. "Robust 6dof motion estimation for non-overlapping, multi-camera systems". In: *Proc. Winter Conference on Applications of Computer Vision (WACV)*, 2008.

[Cohe 93] M. F. Cohen, J. Wallace, and P. Hanrahan. *Radiosity and realistic image synthesis*. Academic Press Professional, Inc., 1993.

[Cook 86] R. L. Cook. "Stochastic sampling in computer graphics". *Trans. on Graphics (TOG)*, Vol. 5, No. 1, pp. 51–72, 1986.

[Davi 07] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. "MonoSLAM: Real-time single camera SLAM". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 29, No. 6, pp. 1052–1067, 2007.

[Debe 00] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. "Acquiring the Reflectance Field of a Human Face". In: *Proc. SIGGRAPH*, 2000.

[Debe 97] P. Debevec and J. Malik. "Recovering high dynamic range radiance maps from photographs". In: *Proc. SIGGRAPH*, 1997.

[Debe 98] P. Debevec. "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography". In: *Proc. SIGGRAPH*, 1998.

[dEon 07] E. d'Eon and D. Luebke. "Advanced techniques for realistic real-time skin rendering". In: H. Nguyen, Ed., *GPU Gems 3*, Chap. 14, pp. 293–347, Addison-Wesley, 2007.

[DiVe 08] S. DiVerdi, J. Wither, and T. Höllerer. "Envisor: Online environment map construction for mixed reality". In: *Proc. Int. Conference on Virtual Reality (VR)*, 2008.

[Dodg 04] N. A. Dodgson. "Variation and extrema of human interpupillary distance". In: *Proc. Int. Society for Optics and Photonics (SPIE)*, 2004.

[Enge 13] J. Engel, J. Sturm, and D. Cremers. "Semi-dense visual odometry for a monocular camera". In: *Proc. Int. Conference on Computer Vision (ICCV)*, 2013.

[Enge 14] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-scale direct monocular SLAM". In: *Proc. European Conference on Computer Vision (ECCV)*, 2014.

[Enge 16] J. Engel, V. Koltun, and D. Cremers. "Direct Sparse Odometry". In: *arXiv:1607.02565*, July 2016.

[Epst 95] R. Epstein, P. W. Hallinan, and A. L. Yuille. "5±2 Eigenimages Suffice: An Empirical Investigation of Low-Dimensional Lighting Models". In: *Proc. Workshop on Physics-Based Modeling in Computer Vision*, 1995.

[Fein 97] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment". *Personal Technologies*, Vol. 1, No. 4, pp. 208–217, 1997.

[Fenn 01] R. Fenn. *Geometry*. Springer London, 2001.

[Ferr 07] B. Ferris, D. Fox, and N. D. Lawrence. "WiFi-SLAM Using Gaussian Process Latent Variable Models". In: *Proc. Int. Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[Fisc 81] M. A. Fischler and R. C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395, 1981.

[Flor 13] A. Flores, E. Christiansen, D. Kriegman, and S. Belongie. "Camera Distance from Face Images". In: *Proc. Int. Symposium on Visual Computing (ISVC)*, 2013.

[Four 93] A. Fournier, A. S. Gunawan, and C. Romanzin. "Common Illumination between Real and Computer Generated Scenes". In: *Proc. Graphics Interface (GI)*, 1993.

[Frah 05] J.-M. Frahm, K. Koeser, D. Grest, and R. Koch. "Markerless augmented reality with light source estimation for direct illumination". In: *Proc. Conference on Visual Media Production (CVMP)*, 2005.

[Fran 13] T. A. Franke. "Delta light propagation volumes for mixed reality". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

[Fuch 05] M. Fuchs, V. Blanz, H. Lensch, and H.-P. Seidel. "Reflectance from Images: A Model-Based Approach for Human Faces". *Trans. Visualization and Computer Graphics (TVCG)*, Vol. 11, No. 3, pp. 296–305, 2005.

[Gao 03] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. "Complete solution classification for the perspective-three-point problem". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 25, No. 8, pp. 930–943, 2003.

[Gaug 12] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Höllerer. "Live tracking and mapping from both general and rotation-only camera motion". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2012.

[Geor 01] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 23, No. 6, pp. 643–660, 2001.

[Gibs 00] S. Gibson and A. Murta. "Interactive rendering with real-world illumination". *Rendering Techniques*, Vol. 11, pp. 365–376, 2000.

[Gibs 03] S. Gibson, J. Cook, T. Howard, and R. Hubbold. "Rapid shadow generation in real-world lighting environments". In: *Proc. Eurographics (EG) - Workshop on Rendering*, 2003.

[Gold 83] D. Goldfarb and A. Idnani. "A numerically stable dual method for solving strictly convex quadratic programs". *Mathematical programming*, Vol. 27, No. 1, pp. 1–33, 1983.

[Gree 03] R. Green. "Spherical Harmonic Lighting: The Gritty Details". In: *Archives of the Game Developers Conference*, 2003.

[Gros 07] T. Grosch, T. Eble, and S. Mueller. "Consistent interactive augmentation of live camera images with correct near-field illumination". In: *Proc. Symposium on Virtual Reality Software and Technology (VRST)*, 2007.

[Grub 12] L. Gruber, T. Richter-Trummer, and D. Schmalstieg. "Real-time photometric registration from arbitrary geometry". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2012.

[Grub 14] J. Grubert, H. Seichter, and D. Schmalstieg. "Towards user perspective augmented reality for public displays". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2014.

[Hamm 50] R. W. Hamming. "Error detecting and error correcting codes". *Bell System technical journal*, Vol. 29, No. 2, pp. 147–160, 1950.

[Hamm 64] J. M. Hammersley and H. D. C. *Monte Carlo Methods*. London: Methuen, 1964.

[Harl 13] R. Harle. "A survey of indoor inertial positioning systems for pedestrians". *Communications Surveys & Tutorials*, Vol. 15, No. 3, pp. 1281–1293, 2013.

[Harr 88] C. Harris and M. Stephens. "A combined corner and edge detector". In: *Proc. Alvey Vision Conference*, 1988.

[Hart 03] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[Hill 11] A. Hill, J. Schiefer, J. Wilson, B. Davidson, M. Gandy, and B. MacIntyre. "Virtual transparency: Introducing parallax view into video see-through AR". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[Hiro 96] G. Hirota, D. T. Chen, W. F. Garrett, M. A. Livingston, *et al.* "Superior augmented reality registration by integrating landmark tracking and magnetic tracking". In: *Proc. SIGGRAPH*, 1996.

[Izad 11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.* "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera". In: *Proc. Symposium on User Interface Software and Technology (UIST)*, 2011.

[Jach 12] J. Jachnik, R. A. Newcombe, and A. J. Davison. "Real-Time Surface Light-field Capture for Augmentation of Planar Specular Surfaces". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2012.

[Jaco 05] K. Jacobs, J.-D. Nahmias, C. Angus, A. Reche, C. Loscos, and A. Steed. "Automatic generation of consistent shadows for augmented reality". In: *Proc. Graphics Interface (GI)*, 2005.

[Jung 13] Y. Jung, T. Kim, J. Oh, and H. Hong. "Mobile AR Rendering Method Using Environmental Light Source Information". In: *Proc. Int. Conference on Information Science and Applications (ICISA)*, 2013.

[Kaji 86] J. T. Kajiya. "THE RENDERING EQUATION". In: *Computer Graphics*, 1986.

[Kan 12] P. Kán and H. Kaufmann. "High-quality reflections, refractions, and caustics in Augmented Reality and their contribution to visual coherence". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2012.

[Kan 13] P. Kán and H. Kaufmann. "Differential Irradiance Caching for Fast High-Quality Light Transport Between Virtual and Real Worlds". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

[Kanb 04] M. Kanbara and N. Yokoya. "Real-time Estimation of Light Source Environment for Photorealistic Augmented Reality". In: *Proc. Int. Conference on Pattern Recognition (ICPR)*, 2004.

[Kaut 00a] J. Kautz and M. D. McCool. "Approximation of glossy reflection with prefiltered environment maps". In: *Proc. Graphics Interface (GI)*, 2000.

[Kaut 00b] J. Kautz, P.-P. Vázquez, W. Heidrich, and H.-P. Seidel. "A unified approach to prefiltered environment maps". In: *Rendering Techniques*, pp. 185–196, Springer, 2000.

[Kell 97] A. Keller. "Instant Radiosity". In: *Proc. SIGGRAPH*, 1997.

[Kerl 13] C. Kerl, J. Sturm, and D. Cremers. "Dense visual SLAM for RGB-D cameras". In: *Proc. Int. Conference on Intelligent Robots and Systems (IROS)*, 2013.

[Klei 07] G. Klein and D. Murray. "Parallel tracking and mapping for small AR workspaces". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[Klei 10] G. Klein and D. W. Murray. "Simulating Low-Cost Cameras for Augmented Reality Compositing". *Trans. Visualization and Computer Graphics (TVCG)*, Vol. 16, No. 3, pp. 369–380, 2010.

[Klin 88] G. J. Klinker, S. A. Shafer, and T. Kanade. "The measurement of highlights in color images". *Int. Journal of Computer Vision (IJCV)*, Vol. 2, No. 1, pp. 7–32, 1988.

[Knec 10] M. Knecht, C. Traxler, O. Mattausch, W. Purgathofer, and M. Wimmer. "Differential instant radiosity for mixed reality". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2010.

[Knec 12] M. Knecht, C. Traxler, O. Mattausch, and M. Wimmer. "Reciprocal Shading for Mixed Reality". *Computers and Graphics (C&G)*, Vol. 36, pp. 846–856, 2012.

[Knei 11] L. Kneip, D. Scaramuzza, and R. Siegwart. "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation". In: *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[Knor 14] S. B. Knorr and D. Kurz. "Real-Time Illumination Estimation from Faces for Coherent Rendering". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2014.

[Knor 16] S. B. Knorr and D. Kurz. "Leveraging the User's Face for Absolute Scale Estimation in Handheld Monocular SLAM". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2016.

[Koc 13] E. Koc and S. Balcisoy. "Estimation of Environmental Lighting from Known Geometries for Mobile Augmented Reality". In: *Proc. Int. Conference on Cyberworlds (CW)*, 2013.

[Lalo 14] J.-F. Lalonde and I. Matthews. "Lighting estimation in outdoor image collections". In: *Proc. Int. Conference on 3D Vision (3DV)*, 2014.

[Lamb 92] J. H. Lambert. *Photometrie: Photometria, sive de mensura et gradibus luminis, colorum et umbrae (1760)*. Vol. 31, W. Engelmann, 1892.

[Lee 05] K.-C. Lee, J. Ho, and D. J. Kriegman. "Acquiring Linear Subspaces for Face Recognition under Variable Lighting". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 27, No. 5, pp. 684–698, 2005.

[Lee 09] T. Lee and T. Höllerer. "Multithreaded hybrid feature tracking for markerless augmented reality". *Trans. Visualization and Computer Graphics (TVCG)*, Vol. 15, No. 3, pp. 355–368, 2009.

[LeGe 16] C. LeGendre, X. Yu, D. Liu, J. Busch, A. Jones, S. Pattanaik, and P. Debevec. "Practical multispectral lighting reproduction". *Trans. on Graphics (TOG)*, Vol. 35, No. 4, p. 32, 2016.

[Lema 07] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix. "Vision-based SLAM: Stereo and monocular approaches". *Int. Journal of Computer Vision (IJCV)*, Vol. 74, No. 3, pp. 343–364, 2007.

[Lepe 09] V. Lepetit, F. Moreno-Noguer, and P. Fua. "Epnp: An accurate o (n) solution to the pnp problem". *Int. Journal of Computer Vision (IJCV)*, Vol. 81, No. 2, pp. 155–166, 2009.

[Lieb 11] S. Lieberknecht, A. Huber, S. Ilic, and S. Benhimane. "RGB-D camera-based parallel tracking and meshing". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[Liu 04] X. Liu, P.-P. Sloan, H.-Y. Shum, and J. Snyder. "All-Frequency Precomputed Radiance Transfer for Glossy Objects". In: *Proc. Eurographics (EG)*, 2004.

[Liu 07] H. Liu, H. Darabi, P. Banerjee, and J. Liu. "Survey of wireless indoor positioning techniques and systems". *Trans. Systems, Man, and Cybernetics (SMC), Part C (Applications and Reviews)*, Vol. 37, No. 6, pp. 1067–1080, 2007.

[Liu 09] Y. Liu, X. Qin, S. Xu, E. Nakamae, and Q. Peng. "Light source estimation of outdoor scenes for mixed reality". *The Visual Computer*, Vol. 25, No. 5-7, pp. 637–646, 2009.

[Livi 97] M. A. Livingston *et al.* "Magnetic tracker calibration for improved augmented reality registration". *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 5, pp. 532–546, 1997.

[Lowe 04] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". *Int. Journal of Computer Vision (IJCV)*, Vol. 60, No. 2, pp. 91–110, 2004.

[Lowe 99] D. G. Lowe. "Object recognition from local scale-invariant features". In: *Proc. Int. Conference on Computer Vision (ICCV)*, 1999.

[Mads 08] C. B. Madsen and M. Nielsen. "Towards probe-less augmented reality: a position paper". In: *Proc. Int. Conference on Computer Graphics Theory and Applications (GRAPP)*, 2008.

[Mads 10] C. Madsen and B. Lal. "Probeless Illumination Estimation for Outdoor Augmented Reality". In: S. Maad, Ed., *Augmented Reality*, Chap. 2, pp. 15–30, InTech, 2010.

[Mama 98] P. Mamassian, D. C. Knill, and D. Kersten. "The perception of cast shadows". *Trends in cognitive sciences*, Vol. 2, No. 8, pp. 288–295, 1998.

[Mars 97] S. R. Marschner and D. P. Greenberg. "Inverse Lighting for Photography". In: *Proc. Color Imaging Conference: Color Sciene, Systems and Applications (CIC)*, 1997.

[Meht 15] S. U. Mehta, K. Kim, D. Pajak, K. Pulli, J. Kautz, and R. Ramamoorthi. "Filtering environment illumination for interactive physically-based rendering in mixed reality". In: *Proc. Eurographics (EG)*, 2015.

[Mei 11] X. Mei, H. Ling, and D. W. Jacobs. "Illumination recovery from image with cast shadows via sparse representation". *Trans. on Image Processing*, Vol. 20, No. 8, pp. 2366–2377, 2011.

[Meil 13] M. Meilland, C. Barat, and A. Comport. "3D High Dynamic Range Dense Visual SLAM and Its Application to Real-time Object Re-lighting". In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

[Meka 16] A. Meka, M. Zollhöfer, C. Richardt, and C. Theobalt. "Live Intrinsic Video". *Trans. on Graphics (TOG)*, Vol. 35, No. 4, 2016.

[Meta 15] Metaio GmbH. "MetaioSDK". `http://www.metaio.com/sdk` (Offline), 2015.

[Miro 13] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald. "SignalSLAM: Simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals". In: *Proc. Int. Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013.

[Murp 09] E. Murphy-Chutorian and M. M. Trivedi. "Head pose estimation in computer vision: A survey". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 31, No. 4, pp. 607–626, 2009.

[Naka 86] E. Nakamae, K. Harada, T. Ishizaki, and T. Nishita. "A Montage Method: The Overlaying of The Computer Generated Images onto a Background Photograph". In: *Proc. SIGGRAPH*, 1986.

[Neve 12] N. Neverova, D. Muselet, and A. Trémeau. "Lighting estimation in indoor environments from low-quality images". In: *Proc. European Conference on Computer Vision (ECCV) - Workshops and Demonstrations*, 2012.

[Newc 11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *Proc. Int. Conference on Computer Vision (ICCV)*, 2011.

[Nico 77] F. Nicodemus, J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis. "Geometrical Considerations and Nomenclature for Reflectance". *Final Report National Bureau of Standards*, Vol. 1, 1977.

[Niko 13] T. Nikodỳm, V. Havran, and J. Bittner. "Multiple Live Video Environment Map Sampling". *Journal of WSCG*, Vol. 21, No. 2, pp. 127–136, 2013.

[Nime 95] J. S. Nimeroff, E. Simoncelli, and J. Dorsey. "Efficient Re-rendering of Naturally Illuminated Environments". In: *Photorealistic Rendering Techniques*, pp. 373–388, Springer, 1995.

[Nish 04] K. Nishino and S. K. Nayar. "Eyes for Relighting". In: *Trans. on Graphics (TOG)*, 2004.

[Nist 04] D. Nistér, O. Naroditsky, and J. Bergen. “Visual odometry”. In: *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[Nutz 11] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. “Fusion of IMU and vision for absolute scale estimation in monocular SLAM”. *Journal of Intelligent & Robotic Systems (IROS)*, Vol. 61, No. 1-4, pp. 287–299, 2011.

[Okab 04] T. Okabe, I. Sato, and Y. Sato. “Spherical harmonics vs. haar wavelets: Basis for recovering illumination from cast shadows”. In: *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[Pear 95] K. Pearson. “Note on regression and inheritance in the case of two parents”. *Proc. of the Royal Society of London*, Vol. 58, pp. 240–242, 1895.

[Pess 10] S. Pessoa, G. Moura, J. Lima, V. Teichrieb, and J. Kelner. “Photorealistic rendering for augmented reality: A global illumination and brdf solution”. In: *Proc. Int. Conference on Virtual Reality (VR)*, 2010.

[Phar 10] M. Pharr and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2 Ed., 2010.

[Poly 08] A. D. Polyanin and A. V. Manzhirov. *Handbook of integral equations*. CRC press, 2 Ed., 2008.

[Qing 04] L. Qing, S. Shan, and W. Gao. “Eigen-Harmonics Faces: Face Recognition under Generic Lighting”. In: *Proc. Int. Conference on Automatic Face and Gesture Recognition (FG)*, 2004.

[Rama 01] R. Ramamoorthi and P. Hanrahan. “A signal-processing framework for inverse rendering”. In: *Proc. SIGGRAPH*, 2001.

[Rama 02] R. Ramamoorthi. “Analytic PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object”. *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 24, No. 10, pp. 1322–1333, 2002.

[Rama 06] R. Ramamoorthi. “Modeling Illumination Variation with Spherical Harmonics”. *Face Processing: Advanced Modeling Methods*, pp. 385–424, 2006.

[Rama 88] V. S. Ramachandran. “Perceiving Shape from Shading”. *Scientific American*, Vol. 259, pp. 76–83, 1988.

[Rein 10] E. Reinhard, G. Ward, S. N. Pattanaik, P. E. Debevec, W. Heidrich, and K. Myszkowski. *High Dynamic Range Imaging - Acquisition, Display, and Image-Based Lighting*. Academic Press, 2 Ed., 2010.

[Rich 16] T. Richter-Trummer, J. Park, D. Kalkofen, and D. Schmalstieg. “Instant Mixed Reality Lighting from Casual Scanning”. In: *Proc. Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2016.

[Rost 06] E. Rosten and T. Drummond. "Machine learning for high-speed corner detection". In: *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[Sato 99] I. Sato, Y. Sato, and K. Ikeuchi. "Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene". *Trans. Visualization and Computer Graphics (TVCG)*, Vol. 5, No. 1, pp. 1–12, 1999.

[Sim 01] T. Sim and T. Kanade. "Illuminating the Face". Tech. Rep. CMU-RI-TR-01-31, Carnegie Mellon University, the Robotics Institute, 2001.

[Sloa 02] P.-P. Sloan, J. Kautz, and J. Snyder. "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments". *Trans. on Graphics (TOG)*, Vol. 21, No. 3, pp. 527–536, 2002.

[Sloa 08] P.-P. Sloan. "Stupid Spherical Harmonics (SH) Tricks". In: *Game Developers Conference*, 2008.

[Son 12] W. Son, B. Nam, T. Kim, and H. Hong. "Using environment illumination information in mobile Augmented Reality". In: *Proc. Int. Conference on Consumer Electronics (ICCE)*, 2012.

[Stat 16] Statistisches Bundesamt (Destatis). *Private Haushalte in der Informationsgesellschaft – Nutzung von Informations- und Kommunikationstechnologien - Fachserie 15 Reihe 4 - 2015*. Wiesbaden, 2016.

[Stew 06] H. Stewenius, C. Engels, and D. Nistér. "Recent developments on direct relative orientation". *Journal of Photogrammetry and Remote Sensing (ISPRS)*, Vol. 60, No. 4, pp. 284–294, 2006.

[Stor 00] M. Störring, E. Granum, and H. J. Andersen. "Estimation of the illuminant colour using highlights from human skin". In: *Proc. Int. Conference on Color in Graphics and Image Processing*, 2000.

[Stra 12] H. Strasdat, J. M. Montiel, and A. J. Davison. "Visual SLAM: why filter?". *Image and Vision Computing*, Vol. 30, No. 2, pp. 65–77, 2012.

[Supa 06] P. Supan, I. Stuppacher, and M. Haller. "Image Based Shadowing in Real-Time Augmented Reality". *Int. Journal of Virtual Reality (IJVR)*, Vol. 5, No. 3, pp. 1–7, 2006.

[Tans 13] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. "Live metric 3d reconstruction on mobile phones". In: *Proc. Int. Conference on Computer Vision (ICCV)*, 2013.

[Trig 99] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. "Bundle adjustment—a modern synthesis". In: *Proc. Int. Workshop on Vision Algorithms: Theory and Practice*, 1999.

[Umey 91] S. Umeyama. "Least-squares estimation of transformation parameters between two point patterns". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 13, No. 4, pp. 376–380, 1991.

[Uran 16] Y. Uranishi, M. Imura, and T. Kuroda. "The Rainbow Marker: An AR marker with planar light probe based on structural color pattern matching". In: *Proc. Int. Conference on Virtual Reality (VR)*, 2016.

[Weis 11] S. Weiss and R. Siegwart. "Real-time metric state estimation for modular vision-inertial systems". In: *Proc. Int. Conference on Robotics and Automation (RAS)*, 2011.

[Weyr 06] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen, *et al.* "Analysis of human faces using a measurement-based skin reflectance model". *Trans. on Graphics (TOG)*, Vol. 25, No. 3, pp. 1013–1024, 2006.

[Yao 13] Y. Yao, H. Kawamura, and A. Kojima. "The Hand as a Shading Probe". In: *SIGGRAPH - Posters*, 2013.

[Zhan 00] Z. Zhang. "A flexible new technique for camera calibration". *Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 22, No. 11, pp. 1330–1334, 2000.

[Zhan 03] L. Zhang and D. Samaras. "Face Recognition Under Variable Lighting using Harmonic Image Exemplars". In: *Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.