Intrinsic Images and their Applications in Intelligent Systems

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dipl.-Inform.
Thomas Michael Nestmeyer
aus Nürtingen

Tübingen
2021

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	08.10.2021
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Hendrik P. A. Lensch
2. Berichterstatter:	Dr. Peter V. Gehler

Abstract

This thesis tackles multiple aspects of intelligent systems and especially researches their interconnection through their vision system. We will specifically look into intrinsic images, the decomposition of images into their intrinsic layers such as reflectance and shading. This is an essential component to enable a robust intelligent system which is continuously running a loop of perceiving and acting. First, we examine the part of acting, which is studied in the field of robotics and there we consider a practical application, namely the decentralized exploration of multiple locations by a connected network of robots. This is a relevant task for example in a disaster scenario like an earthquake to search for survivors. We will see how dynamic priority scaling of traveling forces enables the simultaneous and therefore efficient exploration without losing connectivity and how effective visiting of all targets can be guaranteed. In extensive simulations and experiments with real robots, we show its applicability.

Afterwards we continue with the part on perceiving the environment, which is studied in computer vision. We will compile the necessary prelimiaries for instance in machine learning to then be able to understand the later described Convolutional Neural Network (CNN) approach with which we will predict intrinsic images in a generative manner. The actual separation is mathematically ill-posed and data are hard to come by, so we first tackle it by learning from existing sparse human relative reflectance judgments and present a direct CNN prediction model exposing fast inference. Continuing on, we look at a novel way of introducing the prior knowledge of sparse reflectance as edge-aware filtering to improve on its results.

Another big goal of intelligent systems is Augmented Reality (AR). To enable telepresence when conferencing with another participant in AR, one needs to project the sender's face into the receiver's augmented reality glasses and match their scene lighting. This requires an understanding of arbitrarily (re-)lighting human faces. The separation into reflectance and shading under the Lambertian assumption alone is not enough in that case, because it leads to flat looking faces under strong directional light since all the specularities and cast shadows are missing. Realizing this, we create a novel relighting dataset which provides additional intrinsic layers that we then later on learn to predict. We describe a structured approach that generates relit portraits through a mix of rendering which utilizes the predicted intrinsic layers of albedo and normals and additional non-Lambertian ingredients. This approach proves to be beneficial over a completely unstructured model or a merely rigid Lambertian rendering.

In summary, the overall goal of the thesis is to research intelligent systems and to provide one more innovative piece in the puzzle towards general artificial intelligence. Because one quickly realizes the importance of computer vision for this endeavor, and in there specifically the need to understand the 3D world through their 2D projections into images, we thoroughly investigate the field of intrinsic images in this thesis and improve the intrinsic decomposition of arbitrary images to enable smarter intelligent systems. We demonstrate the utilization of such a decomposition in the task of relighting, where the intrinsic structure is shown to improve results.

Kurzfassung

Diese Arbeit befasst sich mit mehreren Aspekten intelligenter Systeme und erforscht vor allem deren Zusammenspiel durch das Bildverarbeitungssystem. Wir werden uns speziell mit intrinsischen Bildern beschäftigen, also der Zerlegung von Bildern in ihre intrinsischen Schichten wie Reflexionsgrad und Schattierung. Dies ist eine wesentliche Komponente, um ein robustes intelligentes System zu ermöglichen, das kontinuierlich eine Schleife aus Wahrnehmen und Handeln durchläuft. Zunächst betrachten wir den Teil des Handelns, der im Bereich der Robotik untersucht wird und betrachten dort eine praktische Anwendung, nämlich die dezentrale Erkundung mehrerer Orte durch ein verbundenes Roboter-Netzwerk. Dies ist eine relevante Aufgabe zum Beispiel in einem Katastrophenszenario wie einem Erdbeben, um nach Überlebenden zu suchen. Wir werden sehen, wie eine dynamische Skalierung der Regelungskräfte basierend auf Prioritätenrollen die gleichzeitige und damit effiziente Exploration ohne Verlust der Konnektivität ermöglicht und wie das effektive Aufsuchen aller Ziele garantiert werden kann. In umfangreichen Simulationen und Experimenten mit realen Robotern belegen wir die Anwendbarkeit unserer Methode.

Anschließend fahren wir mit dem Teil zur Wahrnehmung der Umgebung fort, der im Bereich Computer Vision untersucht wird. Wir erarbeiten die notwendigen Voraussetzungen im maschinellen Lernen, um dann den später beschriebenen Convolutional Neural Network (CNN)-Ansatz verstehen zu können, mit dem wir auf generative Weise intrinsische Bilder vorhersagen. Die eigentliche Trennung ist mathematisch nicht lösbar und Referenzdaten sind schwer aufzunehmen. Wir gehen das Problem also zuerst an, indem wir von bestehenden spärlichen menschlichen relativen Reflexionsurteilen lernen und ein direktes CNN-Vorhersagemodell präsentieren, das schnelle Inferenz liefert. Im weiteren Verlauf betrachten wir eine neuartige Möglichkeit, das Vorwissen der spärlichen Veränderung des Reflexionsgrads als Filterungsoperation einzuführen, um die Ergebnisse der Prädiktion zu verbessern.

Ein weiteres großes Ziel von intelligenten Systemen ist die Umsetzbarkeit erweiterter Realität (“Augmented Reality”, AR). Um glaubhafte Telepräsenz in einer AR-Telekonferenz mit anderen Teilnehmern zu ermöglichen, muss man das Gesicht des Senders in die AR-Brille des Empfängers projizieren und dessen Beleuchtung der Szene anpassen. Dies erfordert ein Verständnis für die (Um-)Beleuchtung beliebiger menschlicher Gesichter. Die Trennung in Reflexionsgrad und Schattierung unter der Lambert’schen Annahme allein reicht in diesem Fall nicht aus, denn sie führt bei stark gerichtetem Licht zu flach wirkenden Gesichtern, da alle Lichtspiegelungen und Schlagschatten fehlen. Um dies zu lösen, erstellen wir einen neuen Datensatz, der zusätzliche intrinsische Schichten liefert, die wir dann später lernen vorherzusagen. Wir beschreiben einen strukturierten Ansatz zur Umbeleuchtung von Porträts durch eine Mischung aus Rendern, das die vorhergesagten intrinsischen Schichten von Albedo und Normalen nutzt, und zusätzliche nicht-Lambert'sche Bestandteile betrachtet. Dieser Ansatz erweist sich als vorteilhaft gegenüber einem völlig unstrukturierten Modell oder einem rein starren Lambert'schen Rendern.

Zusammenfassend lässt sich sagen, dass das übergeordnete Ziel dieser Arbeit darin besteht, intelligente Systeme zu erforschen und ein weiteres innovatives Puzzlestück auf dem Weg zur allgemeinen künstlichen Intelligenz zu liefern. Da man schnell erkennt, wie wichtig Computer Vision für dieses Vorhaben ist, und dort speziell die Notwendigkeit, die 3D-Welt durch ihre 2D-Projektionen in Bilder zu verstehen, untersuchen wir in dieser Arbeit gründlich das Gebiet der intrinsischen Bilder und verbessern die intrinsische Dekomposition beliebiger Bilder, um intelligentere Systeme zu ermöglichen. Wir demonstrieren die Verwendung einer solchen Dekomposition am Beispiel der Umbeleuchtung, wo die intrinsische Struktur nachweislich die Ergebnisse maßgeblich verbessert.

Acknowledgments

This work would not have been possible without many people that contributed in a variety of different ways.

First and foremost, my thanks go out to Peter Gehler for taking me on as a PhD student in the Perceiving Systems department of the Max Planck Institute for Intelligent Systems, and for guiding me along the long and exhausting way of a PhD. He taught me the importance of Machine Learning outside the field of Deep Learning, like Probabilistic Graphical Models, to get a broad education which helps seeing things in perspective. Furthermore, I would like to thank Hendrik Lensch to serve as the official supervisor from University. He provided a welcoming environment in which I learnt a lot about light stages, RAW captures, and (actual) HDR which later on proved to be very relevant knowledge.

A big appreciation also goes out to to Michael Black and the Max Planck Society in general for providing me with a stipend which enabled me to work on this topic in the first place without the need to be distracted by financing my studies. He always made sure that we have the best possible network of academics around us and provided us with an excellent environment to study in and concentrate on everyone's work in general. From him, I learnt how to find and refine the nuggets of what our works are really about. Outside all the technical environment it is important to mention the role of Melanie Feldhofer without whom the department would not work so frictionless. It filled me with great sorrow when I heard about the sad news about Rocko. He was always by our sides and served to be a psychologist when acceptance results did not turn out as expected or when one simply felt to be overtaken by all the pressure and uncertainty about our own future.

For the earlier work on Robotics, I want to send my gratitude especially to Antonio Franchi, who sparked the trust in myself and encouraged me to do a PhD in the first place. For the later time at the Facebook Reality Labs in Pittsburgh, I want to thank Iain Matthews who tought me a lot about technical thoroughness, camera and LED technology, and generally perseverance. He also introduced me to Jean-François Lalonde who was very supportive on our project on relighting faces and I always enjoyed our productive discussions.

The whole Perceiving Systems Department gave me a great time working on my PhD studies, but in some of the members and guests of the Perceiving Systems group, I found friends for life, amongst which are Varun Jampani, Christoph Lassner, Fatma Güney, Laura Sevilla Lara, Sergey Prokudin and Angjoo Kanazawa (in the order of when I have met them).

As to the even longer standing friends who make me enjoy life, I want to start by mentioning Andreas Lehrmann who always makes me test extremes. We climbed the highest mountains, swam along kilometers of cliffs to find the best snorkeling spot and with whom I finished one of my biggest achievements in life, a long distance triathlon. You are not only an Ironman, but a dear friend for life. As for Martin Riedel, even though you live far away in varying places, whenever we talk, I enjoy the time and remember our many hours spent on the race bike, the laid-back evenings and I will never forget our amazing hiking trip through Hardangervidda in Norway. With Sebastian Boegel, I enjoyed a wonderful time in Tübingen's highest flat and for regularly hosting and catering me every Sunday and enjoying "Tatort" together, I want to thank Barbara Rakitsch and Meike Sprecher.

A very special acknowledgment goes out to my wife Carina. She accompanies me through good and bad times, accepts me with all my quirks, shares some of the biggest laughs with me and never stops believing in me. Going into all the details I am thankful for would need way too much space and some are best kept between us. Thank you for being there for me and our little family.

I am also deeply grateful to my family who have provided constant and unconditional support in my life and especially to my parents who enabled me to enjoy such a thorough education in the first place.

Lastly, on a small side note, thanks to cryptocurrencies for helping me reach financial freedom and in general thanks to all the people I met throughout my life, and who had good or bad intentions, for making me the person that I am now. It was a long journey, but it was worth it.

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
AP	Average Precision
AR	Augmented Reality
BF	Bilateral Filter
BRDF	Bidirectional Reflectance Distribution Function
CIE	International Commission on Illumination (French: Commission Internationale de l'Éclairage)
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Field
DSSIM	Dissimilarity measure of Structural SIMilarity
ELU	Exponential Linear Unit
GF	Guided Filter
GAN	Generative Adversarial Network
GNU	GNU's Not Unix
GPU	Graphics Processing Unit
HDR	High Dynamic Range
IIW	Intrinsic Images in the Wild
LED	Light-Emitting Diode
LPIPS	Learned Perceptual Image Patch Similarity
MIT	Massachusetts Institute of Technology
MLP	Multi-Layer Perceptron
PMS	Photometric Stereo
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
ROS	Robot Operating System
SAW	Shading Annotations in the Wild
SGD	Stochastic Gradient Descent
SH	Spherical Harmonics
(MS-)SSIM	(Multi-Scale) Structural SIMilarity
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
WHDR	Weighted Human Disagreement Rate

Chapter 1

Introduction

One of the big research goals in recent years is to create intelligent autonomous systems that interact with our world. In this thesis, two different aspects of such intelligent systems are explored, robotics and computer vision.

1.1 Intelligent Systems

In engineering and computer science, a system describes a collection of connected elements or components that are organized for a common purpose. This can either be emobied, in the form of one or multiple robot(s), which involve mechanical parts as connected elements, but just as well an immobile device like a computer, which also consists of multiple connected logical circuits, and where the application running defines the rationale of its internal components. Grasping a definition of an *intelligent* system, on the other hand, is much harder. To quote from the Journal of Intelligence:

> Intelligence is what the intelligence test measures. Seriously
>
> Van der Maas et al. [2014]

While it is likely that tests designed by humans to measure intelligence will mostly be geared towards measuring human intelligence, one such test for artificial intelligence could be the Turing test to decide whether machines can *think*, which by Alan Turing himself was named the imitation game:

> 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think' ... Instead of attempting such a definition, I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words. The new form of the problem can be described in terms of a game which we call the 'imitation game'. It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. ... We now ask the question, 'What will happen when a machine takes the part of A in this

> game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?'
>
> Turing [1950]

So to avoid the hard task of defining artificial intelligence, we could compare it to human intelligence, as Turing suggests. On the other hand, one might ask if

> The question of whether machines can *think* is about as relevant as the question of whether submarines can *swim*.
>
> Dijkstra [1984]

Meaning the problem arises more with the exact interpretation of the words themselves than providing an actual definition for them. Therefore, to answer what (artificial) intelligence is, we might want to look at why it is desirable in the first place. Potentially, a vague description of the outcome of intelligence could actually be its most accurate definition:

> Intelligence tries to maximise future freedom of action and keep options open.
>
> Wissner-Gross and Freer [2013]

This is inspired by realizing that many computer programs we deem 'intelligent' made actions to maximize future options and not to be trapped. In order to do so, the intelligent system actively needs to pass through local minima in order to eventually find globally better opportunities. Recently, this desire for maximizing entropy can for example be seen in the progress of artificial intelligence in the game 'Go' [Silver et al., 2016, 2017, 2018].

Despite such breakthroughs, artificial *general* intelligence, the ability to perform any task a human can (and potentially more), will likely continue to be hard to achieve for a long time. Nonetheless, the research community, with this work included, tries to create intelligent systems for specific tasks in order to step by step piece the puzzle together and to arrive there at some point when building on previous work and therefore standing on the shoulders of giants, considering

> We see more and farther than our predecessors, not because we have keener vision or greater height, but because we are lifted up and borne aloft on their gigantic stature.
>
> Bernard of Chartres, quoted in of Salisbury [1159]

1.2 Motivation and structure of the thesis

Extending the two examples from before, in this thesis we will investigate intelligent systems in *Robotics* (Part I) and *Computer Vision* (Part II) and how they are interconnected through their vision system.

While in Robotics, we try to perceive the world and act accordingly, in our Computer Vision example of *Relighting*, we will construct novel views of exisiting portrait captures under arbitrary lighting. In order to do so, an appropriate representation as intrinsic layers is described in the interconnecting chapters examining how such a separation into intrinsic layers can effectively and efficiently be done.

Robotics. We first look at multi robot systems, where the swarm behavior of the robots decides about the intelligence of the system. Success of multi-robot systems is based on their ability of parallelizing the execution of several small tasks composing a larger complex mission. Therefore, we discuss the problem of planning trajectories of a team of multiple mobile robots to visit a list of targets in sequence under the constraint of continuous connectivity in order to not lose the potential for communication between the robots in Chapter 2. This is for example relevant in a disaster scenario, to find survivors after an earthquake in a damaged house with the danger of a collapse. Continuous connectivity is required because a global network might not be available (anymore) and therefore a local network needs to be created through multi hops.

We will realize, that a robotic system needs to perceive the world through sensors, amongst which a camera system gives strong clues, therefore we will continue to look into the side of visual inference.

Computer Vision. The world can be perceived through a multitude of sensors, e.g., cameras, radar, ultrasound, pressure, proximity, heat, position/velocity/acceleration and commonly a fusion of the measured information of several sensors is used in order to solve the task at hand. After all, humans rely heavily on the vision system. Roughly 90% of our sensory information comes from vision, why half of the neural tissue is devoted to vision and about two-thirds of the electrical activity of the brain is consumed by vision when the eyes are open [Sells and Fixott, 1957]. Since this enables us humans to efficiently navigate in and interact with the world quite well, it suggests to let robots rely heavily on cameras, too. Therefore we will spent the second part of this thesis exploring the vast field of computer vision, which aims to invert the lossy projection of our 3D world into a 2D image thereof. Our specific focus will be the inference of physical properties from images.

In Chapter 3, we will first discuss the preliminaries of computer vision, machine learning and intrinsic images to serve as a technically sound basis for the rest of the thesis. After setting the tone, we will continue by developing a way to predict reflectance in a completely data-driven way, no priors needed, which leads to a very fast intrinsic image decomposition by design in Chapter 4. We will then improve reflectance prediction results by adding a novel way to introduce a well-known prior of piecewise constant reflectance based on filtering in Chapter 5.

This gives us the knowledge to understand the second example for an intelligent system, an augmented reality (AR) setup for relighting. In AR, we use artificial intelligence to, e.g., properly place virtual objects into a scene spotted by a person wearing AR glasses. In the case of a tele-conference in AR for example, a person's portrait is captured in one illumination and needs to be projected into the receiver's scene, which normally exhibits a different illumination. Therefore we need to be able to relight that sender's face to make the scene believable. In order to be equipped with the essential data to train and evaluate our relighting model, we will first discuss how to capture the necessary layers in Chapter 6. Afterwards, tackling the problem of relighting in Chapter 7, we will demonstrate that modeling non-diffuse effects like specularities and cast shadows is important to achieve a lively relighting result. We therefore realize that the separation under the Lambertian assumption from the early chapters is not enough in a fully constrained rendering pipeline. An otherwise purely learned generative process in the form of an unstructured decoder makes its own errors, why we will therefore go for a structured generative approach augmenting the constrained Lambertian separation with an unconstrained non-Lambertian correction term.

1.3 Research questions

This thesis' chapters are therefore organized into answering the following research questions:

Chapter 1 What are intelligent systems?

Chapter 2 How to efficiently and autonmously steer multiple robots to solve a common task in parallel without loosing communication among them?

Chapter 3 What is needed to understand the current progress in machine learning and computer vision?

Chapter 4 Can intrinsic images be learnt from sparse human annotations in the form of pairwise relative reflectance judgments only?

Chapter 5 Which alternative ways can be innovated to introduce the prior knowledge of piecewise constant reflectance into general intrinsic image algorithms?

Chapter 6 How does a capture setup and resulting processing need to look like to provide the recovery of a multitude of intrinsic layers?

Chapter 7 What level of structure is beneficial when using intrinsic layers for relighting under strong directional light?

In summary, the overall goal of the thesis is to improve intrinsic image research to enable better perception and therefore action planning in intelligent systems. Furthermore, we demonstrate the application of such an intrinsic decomposition explicitly by improving a neural network generator for relighting to be used in, e.g., augmented reality.

Part I

Robotics

Chapter 2

Decentralized Simultaneous Multi-target Exploration using a Connected Network of Multiple Robots

In this chapter, we will introduce a novel decentralized control strategy for a multi-robot system that enables parallel multi-target exploration while ensuring a time-varying connected topology in cluttered 3D environments. Flexible continuous connectivity is guaranteed by building upon the connectivity maintenance method of Robuffo Giordano et al. [2013], in which limited range, line-of-sight visibility, and collision avoidance are taken into account at the same time. Its main points are summarized in Appendix A to be self-contained. Completeness of the decentralized multi-target exploration algorithm is guaranteed by dynamically assigning the robots with different motion behaviors during the exploration task. One major group is subject to a suitable downscaling of the main traveling force based on the traveling efficiency of the current leader and the direction alignment between traveling and connectivity force. This supports the leader in always reaching its current target and, on a larger time horizon, that the whole team realizes the overall task in finite time. Extensive Monte Carlo simulations with a group of several quadrotor UAVs show the scalability and effectiveness of the proposed method and experiments validate its practicability.

Contributions Gradually building the work through Nestmeyer et al. [2013b,c] lead to my diploma thesis [Nestmeyer, 2012]. The work presented in the following continues building thereon and extends it in several ways. The method is modified to achieve a more streamlined presentation using a state machine. Extended details on the decentralized election of the 'prime traveler' are given and a detailed completeness proof for the autonomous exploration is provided. While implementation details were thoroughly discussed in Nestmeyer [2012] already, results at that time were purely gathered from simulations. These initial simulations are extended here to show large scale applicability of the method. Furthermore, experiments with real robots are conducted and thoroughly evaluated to proof the suitability of the proposed method in practice. The resulting work, presented in the following, is published as Nestmeyer et al. [2017] in the journal "Autonomous Robots".

2.1 Introduction

Success of multi-robot systems is based on their ability of parallelizing the execution of several small tasks composing a larger complex mission such as, for instance, the inspection of a certain number of locations either generated off- or online during the robot motion (e.g., exploration, data collection, surveillance, large-scale medical supply or search and rescue [Howard et al., 2006, Franchi et al., 2009, Pasqualetti et al., 2012, Murphy et al., 2008, Faigl and Hollinger, 2014]). In all these cases, a fundamental difference between a group of many single robots and a multi-robot system is the ability to communicate (either explicitly or implicitly) in order to then cooperate together towards a common objective. Another distinctive characteristic in multi-robot systems is the absence of central planning units, as well as all-to-all communication infrastructures, leading to a *decentralized* approach for algorithmic design and implementations [Lynch, 1997]. While communication of a robot with every other robot in the group (via multiple hops) would still be possible as long as the group stays connected, in a decentralized approach each robot is only assumed to be able to communicate with the robots in its 1-hop neighborhood (i.e., typically the ones spatially close by). This brings the advantage of scalability in communication and computation complexity when considering groups of many robots.

The possibility for every robot to share information (via, possibly, multiple hops/iterations) with any other robot in the group is a basic requirement for typical multi-robot algorithms and, as well-known, it is directly related to the connectivity of the underlying *graph* modeling inter-robot interactions. Graph connectivity is a prerequisite to properly fuse the information collected by each robot, e.g., for mapping, localization, and for deciding the next actions to be taken. Additionally, many distributed algorithms like consensus [Olfati-Saber and Murray, 2004] and flooding [Lim and Kim, 2001] require a connected graph for their successful convergence. Preserving graph connectivity during the robot motion is, thus, a fundamental requirement; however, connectivity maintenance may not be a trivial task in many situations, e.g., because of limited capabilities of onboard sensing/communication devices which can be hindered by constraints such as occlusions or maximum range. Given the cardinal role of communication for the successful operation of a multi-robot team, it is then not surprising that a substantial effort has been spent over the last years for devising strategies able to preserve graph connectivity despite constraints in the inter-robot sensing/communication possibilities, see, e.g., Antonelli et al. [2005], Stump et al. [2008, 2011], Pei and Mutka [2012], Robuffo Giordano et al. [2013]. In general, *fixed topology* methods represent conservative strategies that achieve connectivity maintenance by restraining any pairwise link of the interaction graph to be broken during the task execution. A different possibility is to aim for *periodical connectivity* strategies, where each robot can remain separated from the group during some period of time for then rejoining when necessary. *Continuous connectivity* methods instead try to obtain maximum flexibility (links can be continuously broken and restored unlike in the fixed topology cases) while preserving at any time the fundamental ability for any two nodes in the group to share information via a (possibly multi-hop) path (unlike in periodical connectivity methods).

With respect to this state-of-the-art, the problem tackled in this chapter is the design of a *multi-target* exploration/visiting strategy for a team of mobile robots in a cluttered environment able to

1. allow visiting multiple targets at once (for increasing the efficiency of the exploration), while
2. always guaranteeing connectivity maintenance of the group despite some typical sensing/communication constraints representative of real-world situations,
3. without requiring presence of central nodes or processing units (thus, developing a fully *decentralized* architecture), and
4. without requiring that all the targets are known at the beginning of the task (thus, considering *online target generation*).

2.2 Related Work

Designing a decentralized strategy that combines multi-target exploration and continuous connectivity maintenance is not trivial as these two goals impose often antithetical constraints. Several attempts have indeed been presented in the previous literature: a *fixed-topology* and centralized method is presented in Antonelli et al. [2005], which, using a virtual chain of mobile antennas, is able to maintain the communication link between a ground station and a single mobile robot visiting a given sequence of target points. The method is further refined in Antonelli et al. [2006]. A similar problem is addressed in Stump et al. [2008] by resorting to a partially centralized method where a linear programming problem is solved at every step of motion in order to mix the derivative of the second smallest eigenvalue of a weighted Laplacian (also known as algebraic connectivity, or Fiedler eigenvalue) and the k-connectivity of the system. A line-of-sight communication model is considered in Stump et al. [2011], where a centralized approach, based on polygonal decomposition of the known environment, is used to address the problem of deploying a group of roving robots while achieving *periodical connectivity*. The case of periodical connectivity is also considered in Pasqualetti et al. [2012] and Hollinger and Singh [2012]. The first paper optimally solves the problem of patrolling a set of points to be visited as often as possible. The second presents a heuristic algorithm exploiting the concept of implicit coordination. *Continuous connectivity* between a group of robots exploring an unknown 2D environment and a single base station is considered in Pei et al. [2010]. The proposed exploration methodology, similar to the one presented in Franchi et al. [2009], is integrated with a centralized algorithm running on the base station and solving a variant of the Steiner Minimum Tree Problem. An extension of this approach to heterogeneous teams is presented in Pei and Mutka [2012].

Zavlanos and Pappas [2007] exploit a potential field approach to keep the second smallest eigenvalue of the Laplacian positive. The method is tested with ground robots in an empty environment and assumes that each robot has access to the whole formation for computing the connectivity eigenvalue and the associated potential field. It is therefore not scalable, because the strength of all links has to be broadcasted to all robots in the group. Continuous connectivity achieved by suitable mission planning is described in Mosteo et al. [2008], although this work does not allow for parallel exploration. Another method providing flexible connectivity based on a spring-damper system, but not able to handle significant obstacles, is reported in Tardioli et al. [2010].

A decentralized strategy addressing the problem of continuous connectivity maintenance for a multi-robot team is considered in Robuffo Giordano et al. [2013]. In this latter work, the introduction of a sensor-based weighted Laplacian allows to distributively and analytically compute the anti-gradient of a generalized Fiedler eigenvalue. The connectivity maintenance action is further embedded with additional constraints and requirements such as inter-robot and obstacle collision avoidance, and a stability guarantee of the whole system, when perturbed by external control inputs for steering the whole formation, is also provided. Finally, apart for Robuffo Giordano et al. [2013], all the previously mentioned continuous connectivity methods have only been applied to 2D-environment models.

In this work, we leverage upon the general decentralized strategy for connectivity maintenance of Robuffo Giordano et al. [2013] for proposing a solution to the aforementioned problem of decentralized *multi-target exploration* while coping with the (possibly opposing) constraints of continuous connectivity maintenance in a cluttered 3D environment. The main contributions of this work and features of the proposed algorithm can then be summarized as follows:

i) decentralized and continuous maintenance of connectivity,

ii) guarantee of collision avoidance with obstacles and among robots,

iii) possibility to take into account non-trivial sensing/communication models, including maximum range and line-of-sight visibility in 3D,

iv) stability of the overall multi-robot dynamical system,

v) decentralized exploration capability,

vi) possibility for more than one robot to visit different targets at the same time,

vii) online path planning without the need for any (centralized) pre-planning phase,

viii) applicability to both 2D and 3D cluttered environments, and finally

ix) completeness of the multi-target exploration (i.e., all robots are guaranteed to reach all their targets in a finite time).

The items *i) - iv)* have already been tackled in Robuffo Giordano et al. [2013] and are here taken as a basis for our work. On the other hand, the combination of *i) - iv)* with the items *v) - ix)* is a novel contribution: to the best of our knowledge, our work is the first attempt to propose a decentralized multi-target exploration algorithm possessing all the mentioned features altogether.

The rest of the chapter is organized as follows: Sec. 2.3 provides a formal description of the problem under consideration. The proposed algorithm is then thoroughly illustrated in Sec. 2.4. In Sec. 2.5, we report the results of extensive Monte Carlo simulations and experiments with real quadrotors, and Sec. 2.6 concludes the chapter.

2.3 System Model and Problem Setting

We consider a group of N robots operating in a 3D obstacle-populated environment and denote with $q_i \in \mathbb{R}^3$ the position of a reference point of the i-th robot, $i = 1, \ldots, N$, in an inertial world frame. We also let $\mathcal{O}$ be the set of obstacle points in the environment. Each robot i is assumed to be endowed with an omnidirectional sensor able to measure the relative position $q_j - q_i$ of another robot j provided that:

1. $\|q_j - q_i\| < R_s$, where $R_s > 0$ is the maximum sensing range of the sensor, and
2. $\min_{\varsigma \in [0,1], o \in \mathcal{O}} \|q_i + \varsigma(q_j - q_i) - o\| \geq R_o$, i.e., the line segment connecting q_i to q_j is at least at distance $R_o > 0$ away from any obstacle point.

These two conditions account for two common characteristics of exteroceptive sensors, namely, presence of a limited sensing range R_s, and the need for a non-occluded line-of-sight visibility[1]. We further assume that if the i-th robot can measure $q_j - q_i$ then it can also communicate with the j-th robot with negligible delays, that is, the sensing and communication graphs are taken coincident. This assumption is justified by the fact that communication typically relies on wireless technology, thus with a broader range than sensing and without the need for direct visibility to operate. The neighbors of the i-th robot are denoted with $\mathcal{N}_i(t)$, i.e., the (time-varying) set of robots whose relative position can be measured by the i-th robot at time t.

Each robot i is also endowed with a sensor that measures the relative position $o - q_i$ of every obstacle point $o \in \mathcal{O}$ such that $\|o - q_i\| < R_m$, where $R_m > 0$ is the maximum sensing range of this sensor.

Consider the time-varying (undirected) *interaction graph* defined as $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, \ldots, N\}$ and $\mathcal{E}(t) = \{(i, j) \,|\, j \in \mathcal{N}_i(t)\}$. Preserving connectivity of $\mathcal{G}(t)$ for all t allows every robot to communicate *at any time* with any other robot in the network by means of a suitable multi-hop routing strategy, although due to efficiency and scalability reasons, it is always preferred to use one-hop communication when possible.

[1] More complex sensing models could also be taken into account, see Robuffo Giordano et al. [2013] for a discussion in this sense.

As previously stated, decentralized continuous connectivity maintenance is guaranteed by exploiting the method described in Robuffo Giordano et al. [2013], which is based on a gradient-descent action that keeps positive the second smallest eigenvalue λ_2 of the *sensor-based* weighted graph Laplacian [Fiedler, 1973] (see Appendix A for a formal definition).

Each i-th robot is finally endowed with a local motion controller able to let q_i track any arbitrary desired $\bar{\mathcal{C}}^2$ trajectory $q_i(t)$ with a sufficiently small tracking error. This is again a well-justified assumption for almost all mobile robotic platforms of interest, and its validity will also be supported by the experimental results of Sec. 2.5.2. Following the control framework introduced in Robuffo Giordano et al. [2013], the dynamics of q_i is then modeled as the following second order system

$$\Sigma : \begin{cases} M_i\dot{v}_i - f_i^B - f_i^\lambda = f_i \\ \dot{q}_i = v_i \end{cases} \qquad i = 1, \ldots, N \tag{2.1}$$

where $v_i \in \mathbb{R}^3$ is the robot velocity, $M_i \in \mathbb{R}^{3\times 3}$ is its positive definite inertia matrix, and:

1. $f_i^B = -B_i v_i \in \mathbb{R}^3$ is the *damping force* (with $B_i \in \mathbb{R}^{3\times 3}$ being a positive definite damping matrix) meant to represent both typical friction phenomena (e.g., wind/atmosphere drag in the case of aerial robots) and/or a stabilizing control action;

2. $f_i^\lambda \in \mathbb{R}^3$ is the *generalized connectivity force* whose decentralized computation and properties are thoroughly described in Robuffo Giordano et al. [2013] (a short recap is provided in Appendix A);

3. $f_i \in \mathbb{R}^3$ is the *traveling force* used to actually steer the robot motion in order to execute the given task. An appropriate design of f_i is the main goal of this work. As will be clear in the following, special care must be taken in the design of f_i to avoid, for instance, deadlock situations in which the robot group 'gets stuck'.

The following fact, shown in Robuffo Giordano et al. [2013] and recalled in Appendix A, holds:

Fact 1. *As long as f_i keeps bounded, the action of the generalized connectivity force f_i^λ will always ensure obstacle and inter-robot collision avoidance* and *continuous connectivity maintenance for the graph $\mathcal{G}(t)$ despite the various sensing/communication constraints (in the worst case, by completely dominating the bounded f_i).*

To summarize, each robot has

1. an accurate enough measurement of its own location,

2. an omnidirectional sensor which is able to measure relative positions of other robots and obstacles in its close proximity,

3. negligible (compared to the time scale of the robot motion) communication delays with all robots that it can sense/communicate with,

4. the ability to accurately track a smooth path with a force controller.

2.3.1 Multi-target Exploration Problem

We consider the broad class of problems in which each robot runs a black-boxed algorithm that produces *online*[2] a continually adjustable list of targets that have to be visited by the robot in the presented order. We refer to this algorithm as the *target generator* of the i-th robot, and we also assume that the portion of the map needed to reach the next location from the current position q_i is known to robot i. The target generator may represent a large variety of algorithms, such as pursuit-evasion [Durham et al., 2012], patrolling [Pasqualetti et al., 2012], exploration/mapping [Franchi et al., 2009, Burgard et al., 2005], mobile-ad-hoc-networking [Antonelli et al., 2005], and active localization [Jensfelt and Kristensen, 2001]. It might be a cooperative algorithm, or each robot could have a target generator with objectives that are independent from the other target generators. Another possibilty is to appoint a human supervisor as the target generator.

Depending on the particular application, the locations in the lists provided online by the target generators may, e.g., represent:

1. view-points from where to perform the sensorial acquisitions,

2. coordinates of objects that have to be picked up or dropped down,

3. positions of some base stations located in the environment.

We formally denote with $(z_i^1, \dots, z_i^{m_i}) \in \mathbb{R}^{3\times m_i}$ the list of m_i locations provided by the i-th target generator. Additionally, we consider the possibility, for the target generator, to specify a time duration $\Delta t_i^k < \infty$ for which the i-th robot is required to stay close to the point z_i^k, with $k = 1, \dots, m_i$. This quantity may represent, with reference to the previous examples, the time

1. needed to perform a full sensorial acquisition,

2. necessary to pick up/drop down an object,

3. required to upload/download some information from a base station,

[2] By *online*, we mean that the targets are generated at runtime, thus precluding the presence of a preliminary phase in which the robots may *plan in advance* the multi-target exploration action. Indeed, if all the targets are known beforehand, one could still apply our method but other planning strategies might potentially lead to better solutions.

and can also possibly be adjusted at runtime during the execution of the respective task.

Finally, we also introduce the concept of a *cruise speed* $v_i^{\text{cruise}} > 0$ that should be maintained by the i-th robot during the transfer phase from a point to the next one.

Given these modeling assumptions, the addressed problem can be formulated as follows:

Problem 1. *Given a sequence of targets $z_i^1, \dots, z_i^{m_i}$ (presented online) for every robot $i = 1, \dots, N$, together with the corresponding sequence of time durations $\Delta t_i^1, \dots, \Delta t_i^{m_i}$ and a radius R_z, design, for every $i = 1, \dots, N$, a decentralized feedback control law f_i (i.e., a function using only information locally and* 1*-hop available to the i-th robot) for the system described in Eq.* (2.1) *which is bounded and such that, for the closed-loop trajectory $q_i\left(t, f_{i,[0,t)}\right)$, there exists a time sequence $0 < t_i^1 < \ldots < t_i^{m_i} < \infty$ such that for all $k = 1 \ldots m_i$, robot i remains for the duration Δt_i^k within a ball of radius R_z centered at z_i^k, formally $\forall t \in [t_i^k, t_i^k + \Delta t_i^k] : \|q_i(t) - z_i^k\| < R_z$.*

2.4 Decentralized Algorithm

In this section, we describe the proposed distributed algorithm aimed at generating a traveling force f_i that solves Problem 1. We note that the design of such an autonomous distributed algorithm requires special care: When added to the generalized connectivity force in Eq. (2.1), the traveling force f_i should fully exploit the group capabilities to concurrently visit the targets of all robots whenever possible and, at the same time, should not lead to '*local minima*', where the robots get stuck, due to the simultaneous presence of the hard connectivity constraint. While Robuffo Giordano et al. [2013] already gave an exact description of f_i^B and f_i^λ, an application of f_i was kept open. The main focus of this work is to define f_i in such a way that the above mentioned challenges are properly addressed.

In order to provide an overview of the several variables used in Secs. 2.3 and 2.4, we included Table 2.1 for the reader's convenience.

2.4.1 Notation and Algorithm Overview

As in any distributed design, several instances of the proposed algorithms run separately on each robot and locally exchange information with the 'neighboring' instances via communication. Each instance is split into two concurrent routines: a *planning algorithm* and a *motion control algorithm* whose pseudocodes are given in Algorithm 1 and Algorithm 2, respectively. The planning algorithm acts at a higher level and performs the following actions:

- it processes the targets provided by the target generator,
- it generates the desired path to the current target, and

Table 2.1: Meaning of the variable names.

variable	meaning
N	number of robots
q_i	position of i-th robot
v_i	velocity of i-th robot
$\mathcal{O}$	set of obstacle points
R_s	maximum sensing range
R_o	minimum distance to obstacle
R_c	minimum inter-robot distance
$\mathcal{N}_i$	neighbors of i-th robot
$\mathcal{G}$	interaction graph
λ_2	second smallest eigenvalue of the sensor-based weighted graph Laplacian
f_i^λ	generalized connectivity force
f_i^B	damping force
f_i	traveling force
z_i^k	k-th target of i-th robot
Δt_i^k	amount of time to stay close to target z_i^k
R_z	maximum distance to target when anchored
v_i^{cruise}	maximum cruise speed
γ_i	path to current target, starting from position of robot at time of computation
q_i^γ	closest point of path from current position
d_i^γ	length of remaining path
R_γ	distance to path at which it should be re-planned
α_Λ	weighting of position vs. velocity error
e_i	absolute tracking error of i-th robot along path
(x_c, x_M)	tracking error bounds for the traveling efficiency
Λ_i	traveling efficiency of i-th robot (i.e., tracking error nonlinearly scaled to $[0,1]$ based on x_c, x_M)
$\hat{\Lambda}_p^i$	estimation of the traveling efficiency of the 'prime traveler' by the i-th robot
Θ_i	force direction alignment between connectivity and traveling force of i-th robot
σ	weighting between the force direction alignment and the 'prime traveler' traveling efficiency
ρ_i	downscaling factor of a 'secondary traveler', dependent on $\hat{\Lambda}_p^i$, Θ_i and σ

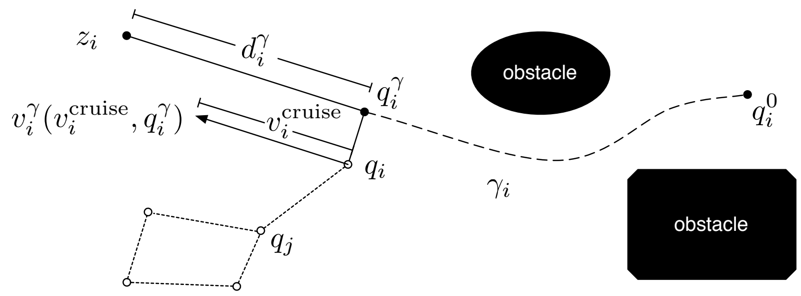


Figure 2.1: Position q_i and path γ_i followed by a traveler from the point q_i^0 to the current target z_i. The solid part of the path represents the *remaining path* which starts at the closest point on the path q_i^γ and whose length is denoted by d_i^γ.

- it selects an appropriate motion control behavior (see later).

The motion control algorithm acts at a lower level by specifying the traveling force f_i as a function of the behavior and the planned path selected by the planning algorithm[3].

The two algorithms have access to the same variables which are formally introduced as follows (see Fig. 2.1 for a graphical representation of some of these variables): the variable $\mathsf{targetQueue}_i$ is filled online by the target generator and contains a list of future targets to be visited by the i-th robot. During the overall running time of the algorithms, the target generator of robot i has access to the whole list $\mathsf{targetQueue}_i$ (which can also be changed online if needed). The current target for the i-th robot (i.e., the last target extracted from the first entry in $\mathsf{targetQueue}_i$) is denoted with z_i. Variable γ_i is a $\bar{\mathcal{C}}^2$ geometric path that leads from the current position q_i of the i-th robot to the target z_i. In our implementation, we used B-splines [Biagiotti and Melchiorri, 2008] in order to get a parameterized smooth path, but any other $\bar{\mathcal{C}}^2$ path would be appropriate. If the robot is not traveling towards any target, then γ_i is set to `null`.

With reference to Fig. 2.1, we also denote with q_i^γ the closest point of the path γ_i to q_i, i.e., the solution of $\arg\min_{p\in\gamma_i} \|p - q_i\|$. In case of multiple solutions, we choose the one with the largest arc-length, i.e., the one nearest to the target along the path. Therefore, the closest point q_i^γ can be considered as unique in the following. The portion of the path γ_i from q_i^γ to z_i is referred to as the *remaining path*, and its length is denoted with d_i^γ.

The motion behavior of the i-th robot is determined by the variable state_i that can take four possible values:

- `connector`,
- `prime-traveler`,
- `secondary-traveler`,

[3] The two routines can run at two different frequencies, typically slower for the planning loop and faster for the motion control loop.

- `anchor`.

The following provides a qualitative illustration of these motion behaviors, while a functional description is given in the next sections:

- *'connector'*: a robot in this state is not assigned any target by the target generator and therefore, its only goal is to help keeping the graph $\mathcal{G}$ connected. For this reason f_i is set to zero and hence the robot is subject solely to the damping and generalized connectivity force f_i^λ in (2.1);
- *'prime traveler'*: a robot in this state travels towards its current target z_i along the path γ_i thanks to the force f_i. At the same time, the robot distributively broadcasts to every other robot a non-negative real number, denoted with Λ_i, that represents its *traveling efficiency*, i.e., a measure of how well it is able to follow its desired path while being influenced by the other robots in the group via the generalized connectivity force f_i^λ (which is described in more detail later). It is essential for the algorithm that only one 'prime traveler' exists in the group at any time. Every other robot with an assigned target needs to be a 'secondary traveler' or 'anchor'. This feature will allow one robot (the 'prime traveler') to reach its target with a high priority, while the other robots will only be allowed to reach their own targets as long as this action does not hinder the 'prime traveler' goal.
- *'secondary traveler'*: a robot in this state travels towards its current target z_i along the path γ_i thanks to the force f_i. The robot keeps an internal estimation $\hat{\Lambda}_p^i$ of the traveling efficiency of the current 'prime traveler', and it scales down the intensity of its traveling force f_i by an *adaptive gain* ρ_i whenever the action of f_i is 'too conflicting' w.r.t. that of f_i^λ, or the 'prime traveler' $\hat{\Lambda}_p^i$ drops lower than a given threshold.
- *'anchor'*: a robot in this state has reached the proximity of the target z_i. The force f_i is then exploited in order to keep q_i within a circle of radius R_z centered at z_i (i.e., the robot is 'anchored' to the target), while waiting for the associated time Δt_i to elapse.

In order to obtain a better intuition of the roles of the robots, we suggest the reader to watch the "Empty Space" video available at `https://homepages.laas.fr/afranchi/robotics/?q=node/144`.

To summarize this qualitative description, these behaviors are designed in such a way that the single 'prime traveler' approaches its target with the highest priority, the 'secondary travelers' approach their targets as long as they have enough spatial freedom by the generalized connectivity force, the 'anchors' stay close to the target until their task is completed, and the 'connectors' help the 'secondary travelers' in providing as much spatial freedom as possible while preserving the connectivity of the graph.

Whenever a robot moves, it may indirectly exert a certain generalized connectivity force on all its neighbors because of the properties of f_i^λ (i.e., for retaining generalized connectivity of the graph $\mathcal{G}$ (see Robuffo Giordano et al. [2013] and Appendix A). This connectivity action can possibly conflict with the traveling force f_i, and also prevent, in the worst case, fulfilment of the multi-target exploration task (e.g., the group falls in a local minimum because two robots start traveling in opposite directions over too large distances, thus threatening connectivity maintenance).

Since the 'connectors' implement $f_i = 0$ by definition, they cannot directly hinder the 'prime traveler' motion. In other words, a group made by all 'connectors' and one 'prime traveler' would always allow the 'prime traveler' to reach its target. Presence of 'anchors' can instead block the 'prime traveler' because of the anchoring force which prevents them to move away from their targets. Nevertheless the anchoring phase can only last for a finite time Δt_i^k after which the 'anchor' changes state and is again free to move.

No such mechanism is instead present for the 'secondary travelers' which would constantly attempt to move along their paths with a ρ_i set to 1. As explained, if many robots are simultaneously traveling in arbitrary directions inside a cluttered environment, while also maintaining connectivity of $\mathcal{G}$, the overall group motion can potentially (and quite easily) fall into a local minimum. The idea behind the gain ρ_i is to then adaptively scale down the traveling force f_i of the 'secondary travelers' whenever either

1. the direction f_i deviates too much from the connectivity force f_i^λ, or

2. the 'prime traveler' motion is nevertheless too obstructed by the actions of the other 'secondary travelers' in the group.

Consequently, this gain $\rho_i \in [0, 1]$ is chosen so that the current 'prime traveler' can always reach its target, no matter the motion planned by the 'secondary travelers' in the group. A formal description of this concept will be given in Sec. 2.4.7.

2.4.2 Start-up phase

The Procedure 'Start-up for Robot *i*' performs the distributed initialization of the planning and motion control algorithms. Its pseudocode is quickly commented in the following.

At the beginning, if targetQueue$_i$ is empty, the path γ_i is set to `null` and state$_i$ is initialized to `connector` (line 3). Otherwise the first target from targetQueue$_i$ is extracted and saved in z_i. Then, the robot i computes a $\bar{\mathcal{C}}^2$ shortest and obstacle-free path γ_i that connects its current position q_i with z_i (line 6). This path is generated with a two-step optimization method: first, the known portion of the map is discretized into an equally spaced grid in 3D with a cell size of R_{grid}. A cell is marked as occupied whenever an obstacle lies inside a radius of R_{grid} around the cell. On this grid, a shortest path is found via A^*. Then, the waypoints obtained from A^* are approximated with a B-spline [Biagiotti and Melchiorri, 2008] in order to remove corners from the path. We note that, depending on the smoothing parameter, this approximation is not guaranteed to

Procedure Start-up for Robot i

```
1  if targetQueue_i is empty then
2      $\gamma_i \leftarrow$ null
3      state_i $\leftarrow$ connector
4  else
5      Extract first target from targetQueue_i and save it as $z_i$
6      $\gamma_i \leftarrow$ Shortest obstacle-free path from $q_i$ to $z_i$
7      Enroll in the list of Candidates to take part in the first distributed 'prime traveler' election
8      if $i = \arg\min_{j \in Candidates} d_j^\gamma$ then
9          state_i $\leftarrow$ prime-traveler
10     else
11         state_i $\leftarrow$ secondary-traveler
12 $\hat{\Lambda}_p^i \leftarrow 0$
13 Run Algorithm 1 and Algorithm 2 in parallel
```

leave enough clearance from surrounding obstacles. Obstacle avoidance is nevertheless ensured thanks to the presence of the generalized connectivity force that prevents any possible collisions by (possibly) locally adjusting the planned path when needed. As an alternative, one could also rely on the method proposed in Masone et al. [2012] for directly generating a smooth path with enough clearance from obstacles.

Subsequently, the robot takes part in the distributed election of the first 'prime traveler' (see Sec. 2.4.3). Depending on the outcome of this election, state$_i$ is set either to `prime-traveler` or `secondary-traveler`.

At the end of the initialization procedure, the estimate $\hat{\Lambda}_p^i$ of the traveling efficiency of the current 'prime traveler' is initialized to zero (line 12) for all robots, and the planning and motion control algorithms are both started (line 13).

2.4.3 Election of the 'prime traveler'

In a general election of a new 'prime traveler', the current 'prime traveler' triggers the election process (line 14 of Algorithm 1), to which every 'secondary traveler' replies with its index and remaining path length, in order to be taken into the list of candidates (line 23). Since this election is a low-frequency event, we chose to implement it via a simple flooding algorithm [Lim and Kim, 2001]. Although this solution complies with the requirement of being decentralized, one could also resort to'smarter' distributed techniques such as [Lynch, 1997]. The 'prime traveler' then waits for $2(N-1)$ steps to collect these replies, being $2(N-1)$ the maximum number of steps needed to reach every robot with flooding and obtain a reply. The winner of this election is then the robot with the shortest remaining path length d_i^γ, i.e., the robot solving $\arg\min_{j \in \text{Candidates}} d_j^\gamma$. In the unlikely event of two (or more) robots having exactly the same remaining path length, the one with the lower index is elected. During the whole election process, the 'prime

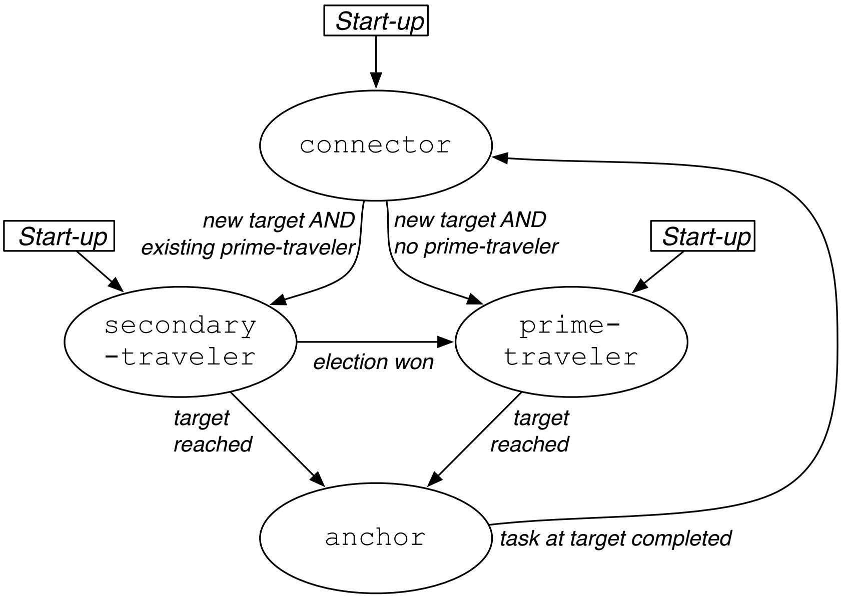


Figure 2.2: State machine of the Algorithm 1.

traveler' keeps its role and only upon decision it abdicates by switching into the 'anchor' state. After announcing the winner, no 'prime traveler' exists in the short time interval (at most $N-1$ steps) until the announcement reaches the winning 'secondary traveler'. This winning robot then switches into the 'prime traveler' behavior. This mechanism makes sure that at most one 'prime traveler' exists at any given time.

The *first* election in the Start-up phase (see Sec. 2.4.2 and line 7 in Procedure 'Start-up for Robot *i*') is handled slightly differently. Instead of the current 'prime traveler' organizing the election, robot 1 is always assigned the role of host and, instead of the only 'secondary travelers' replying, every robot with an assigned target replies with its index and remaining path length (including robot 1 if it has an assigned target).

2.4.4 Planning Algorithm

In this section, we describe in detail the execution of Algorithm 1 running on the i-th robot, whose logical flow is provided in Figure 2.2 as a graphical representation. The algorithm consists of a continuous loop where different decisions are taken according to the value of state_i and according to the following different behaviors:

case connector. If state_i is set to `connector` then $\mathsf{targetQueue}_i$ is checked. In case of an empty queue, state_i remains `connector`, otherwise the next target is extracted from the queue and saved in z_i (line 5). Then the i-th robot computes a $\bar{\mathcal{C}}^2$ shortest and obstacle-free path γ_i connecting the current robot position q_i with z_i (line 6) implementing what was previously described in the start-up procedure. Finally, the robot

Algorithm 1: Planning for Robot *i*

```
while true do
    switch $state_i$ do
        case connector do
            if $targetQueue_i$ is not empty then
                Extract the next target from $targetQueue_i$ and save it as $z_i$
                $\gamma_i \leftarrow$ Shortest obstacle-free path from $q_i$ to $z_i$
                if There is no 'prime traveler' in the group then
                    $state_i \leftarrow$ prime-traveler
                else
                    $state_i \leftarrow$ secondary-traveler
        case prime-traveler do
            if $\|q_i - z_i\| < R_z$ then
                $\gamma_i \leftarrow$ null
                Permit 'prime traveler' candidacy within timeout
                $state_i \leftarrow$ anchor
        case secondary-traveler do
            if $\|q_i - q_i^\gamma\| > R_\gamma$ then
                $\gamma_i \leftarrow$ Shortest obstacle-free path from $q_i$ to $z_i$
            if $\|q_i - z_i\| < R_z$ then
                $\gamma_i \leftarrow$ null
                $state_i \leftarrow$ anchor
            else if 'prime traveler' candidacy is allowed then
                Enroll in the list of Candidates to take part in the distributed 'prime
                  traveler' election
                if $i = \arg\min_{j \in \text{Candidates}} d_j^\gamma$ then
                    $state_i \leftarrow$ prime-traveler
        case anchor do
            if task at target $z_i$ is completed then
                $state_i \leftarrow$ connector
```

changes the value of state_i in order to track γ_i. In particular, if no 'prime traveler' is present in the group, then state_i is set to `prime-traveler` (line 8). Otherwise, state_i is set to `secondary-traveler` (line 10)[4].

case `prime-traveler`. When state_i is set to `prime-traveler` (line 11) and the current position q_i is closer than R_z to the target z_i (line 12), the following actions are performed:

- the path γ_i is reset to `null` (line 13),
- a new distributed 'prime traveler' election as described in Sec. 2.4.3 is announced (line 14),
- the robot abdicates the role of 'prime traveler' and state_i is set to `anchor` (line 15).

If, otherwise, z_i is still far from the current robot position q_i, then state_i remains unchanged and the robot continues to travel towards its target.

case `secondary-traveler`. When state_i is `secondary-traveler` (line 16) the distance $\|q_i^\gamma - q_i\|$ to the (closest point on the) path is checked (line 17). If this distance is larger than the threshold R_γ, the robot replans a path from its current position q_i (line 18). This re-planning phase is necessary since a 'secondary traveler' could be arbitrarily far from the previously planned γ_i because of the 'dragging action' of the current 'prime traveler'. Section 2.4.6 will elaborate more on this point. Subsequently, if q_i is closer than R_z to the target z_i (line 19), the path γ_i is reset to `null` (line 20) and state_i is set to `anchor` (line 21). Otherwise, if the target is still far away, the robot checks whether the 'prime traveler' abdicated and announced an election of a new 'prime traveler' (line 22). If this was the case, the robot takes part in the election (line 23) as described in Sec. 2.4.3. If the robot wins the election (line 24), state_i is set to `prime-traveler` (line 25) otherwise it remains set to `secondary-traveler`.

case `anchor`. The last case of Algorithm 1 is when state_i is `anchor` (line 26). The robot remains in this state until the task at target z_i is completed (line 27), after which state_i is set to `connector`.

2.4.5 Completeness of the Planning Algorithm

Before illustrating the *motion control algorithm*, we state some important properties that hold during the whole execution of the planning algorithm.

[4] Presence of a 'prime traveler' can be easily assessed in a distributed way by, e.g., flooding [Lim and Kim, 2001] on a low frequency.

Propositon 1. *If there exists at least one target in one of the* $\mathsf{targetQueue}_i$*, then exactly one 'prime traveler' will be elected at the beginning of the operation. Furthermore, this 'prime traveler' will keep its state until being closer than* R_z *to its assigned target. In the meantime no other robot can become 'prime traveler'.*

Proof. The start-up procedure guarantees that, if there exists at least one target in at least one of the $\mathsf{targetQueue}_i$, the group of robots includes exactly one 'prime traveler' and no 'anchor' at the beginning of the task. Any other robot is either a 'connector' or 'secondary traveler' depending on the corresponding availability of targets. During the execution of Algorithm 1, a robot can only switch into 'prime traveler' when being a 'connector' or a 'secondary traveler'. As long as there exists a 'prime traveler' in the group, a 'connector' cannot become a 'prime traveler'. Furthermore, a 'secondary traveler' becomes a 'prime traveler' only if it wins the election announced by the 'prime traveler'. Since the 'prime traveler' allows for this election only when in the vicinity of its target (within the radius R_z), the claim directly follows. □

Using this result, the following proposition shows that Algorithm 1 is actually guaranteed to complete the multi-target exploration in the following sense: when presented with a finite amount of targets, all targets of all robots are guaranteed to be visited in a finite amount of time. In order to show this result, an assumption on the robot motion controller is needed.

Assumption 1. *In a group of robots with exactly one 'prime traveler', the adopted motion controller is such that the 'prime traveler' is able to arrive closer than* R_z *to its target in a finite amount of time regardless of the location of the targets assigned to the other robots.*

In Sec. 2.4.7 we discuss in detail how the motion controller introduced in the next section 2.4.6 meets Assumption 1.

Propositon 2. *Given a finite number of targets and a motion controller fulfilling Assumption 1, the whole multi-target exploration task is completed in a finite amount of time as long as the local tasks at every target can be completed in finite time.*

Proof. In the trivial case of no targets, the multi-target exploration task is immediately completed. Let us then assume presence of at least one target. Proposition 1 guarantees existence of exactly one 'prime traveler' at the beginning of the planning algorithm, and that such a 'prime traveler' will keep its role until reaching its target, an event that, by virtue of Assumption 1, happens in finite time. At this point, assuming as a *worst case* that no 'secondary traveler' has reached and cleared its own target in the meantime, one of the following situations may arise:

1. There is at least one 'secondary traveler'. The 'secondary traveler' closest to its target becomes the new 'prime traveler' in the triggered election, and it then starts traveling towards its newly assigned target until reaching it in finite time (Proposition 1 and Assumption 1)

2. There is no 'secondary traveler' and no other 'anchor' besides the former 'prime traveler'. In this case, no other robot has targets in its queue as, otherwise, at least one 'secondary traveler' would exist. Therefore, after completing its task at the target location (a finite duration), the former 'prime traveler' and now 'anchor' becomes 'connector' and, in case of additional targets present for this robot, it switches back into being a 'prime traveler' and travels towards the new targets in a finite amount of time as in case 1.

3. There is no 'secondary traveler', but at least one other 'anchor'. This situation can be split again into two sub-cases:

   a) there exists at least one 'anchor' with a future target in its queue. Then, after a finite time, this 'anchor' becomes 'secondary traveler' and case 1 holds;

   b) there is no 'anchor' with a future target in its queue. Then, after a finite time, all 'anchors' have completed their local tasks and case 2 holds.

In all cases, therefore, one target is visited in finite time by the current 'prime traveler'. Repeating this loop finitely many times, for all the (finite number of) targets, allows to conclude that *all* targets will be visited in a finite amount of time, thus showing the completeness of the planning algorithm.

If a 'secondary traveler' already reaches its target while the 'prime traveler' is active, the aforementioned worst case assumption is not valid anymore. But since, in this case, the target of the 'secondary traveler' is already cleared, the total number of iterations is even smaller than in the previous worst case, thus still resulting in a finite completion time. □

2.4.6 Motion Control Algorithm

With reference to Algorithm 2, we now describe the motion control algorithm that runs in parallel to the planning algorithm on the i-th robot, and whose goal is to determine a traveling force f_i that can meet Assumption 1. The algorithm consists of a continuous loop, as before, in which the force f_i is computed according to the behavior encoded in the variable state_i determined by Algorithm 1:

case `connector`. If state_i is set to `connector`, the estimate $\hat{\Lambda}_p^i$ of the traveling efficiency of the current 'prime traveler' is updated with a consensus-like algorithm (line 4) that will be described in the next Sec. 2.4.7. The traveling force f_i is in this case simply set to 0 (line 5). It is worth mentioning that $f_i = 0$ does not mean the i-th robot will not move, since a 'connector' is still dragged by the other travelers via the generalized connectivity force (according to Eq. (2.1), the 'connectors' are still subject to f_i^λ and f_i^B).

Algorithm 2: Motion Control for Robot *i*

```
1  while true do
2    switch $state_i$ do
3      case connector do
4        Update $\hat{\Lambda}_p^i$ using $\dot{\hat{\Lambda}}_p^i = k_\Lambda \sum_{j \in \mathcal{N}_i} (\hat{\Lambda}_p^j - \hat{\Lambda}_p^i)$
5        $f_i \leftarrow 0$
6      case prime-traveler do
7        $\hat{\Lambda}_p^i \leftarrow \Lambda_i$ using Eq. (2.9)
8        $f_i \leftarrow f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}})$, using Eq. (2.3)
9      case secondary-traveler do
10       Update $\hat{\Lambda}_p^i$ using $\dot{\hat{\Lambda}}_p^i = k_\Lambda \sum_{j \in \mathcal{N}_i} (\hat{\Lambda}_p^j - \hat{\Lambda}_p^i)$
11       $f_i \leftarrow \rho_i f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}})$, using Eqs. (2.3) and (2.15)
12     case anchor do
13       Update $\hat{\Lambda}_p^i$ using $\dot{\hat{\Lambda}}_p^i = k_\Lambda \sum_{j \in \mathcal{N}_i} (\hat{\Lambda}_p^j - \hat{\Lambda}_p^i)$
14       $f_i \leftarrow f_{\text{anchor}}(q_i, z_i, R_z)$, as per Eq. (2.5)
```

case `prime-traveler`. If state$_i$ is set to `prime-traveler`, the estimate $\hat{\Lambda}_p^i$ is set to the true traveling efficiency Λ_i, defined by Eq. (2.9) (line 7). Afterwards (line 8) the robot sets

$$f_i = f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}}), \tag{2.2}$$

where $f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}}) \in \mathbb{R}^3$ is a proportional, derivative and feedforward controller meant to travel along γ_i at a given cruise speed v_i^{cruise}:

$$\begin{aligned} f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}}) = & \, a_i^\gamma(v_i^{\text{cruise}}, q_i^\gamma) \\ & + k_v(v_i^\gamma(v_i^{\text{cruise}}, q_i^\gamma) - \dot{q}_i) \\ & + k_p(q_i^\gamma - q_i). \end{aligned} \tag{2.3}$$

Here, k_p and k_v are positive gains, q_i^γ is the point on γ_i closest to q_i (see Fig. 2.1), $v_i^\gamma(v_i^{\text{cruise}}, q_i^\gamma)$ is the velocity vector of a virtual point traveling along γ_i and passing at q_i^γ with tangential speed v_i^{cruise}, and $a_i^\gamma(v_i^{\text{cruise}}, q_i^\gamma)$ is the acceleration vector of the same point. It is straightforward to analytically compute both the velocity and the acceleration from v_i^{cruise}, given the spline representation of the curve [Biagiotti and Melchiorri, 2008].

case `secondary-traveler`. If state$_i$ is set to `secondary-traveler`, the estimate $\hat{\Lambda}_p^i$ is updated with a consensus-like protocol (line 10). Then (line 11) the robot

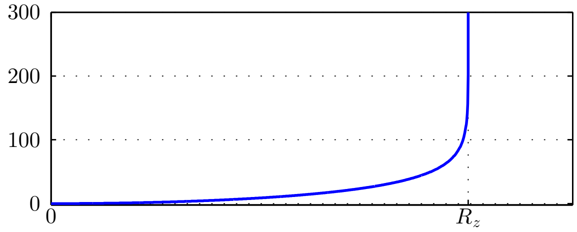


Figure 2.3: Shape of the function $V^{R_z}_{\text{anchor}}(\ell_i)$ defined in Eq. (2.6) that is 0 on the target z_i itself ($\ell_i = 0$) and grows unbounded at the border of a sphere with radius R_z.

sets

$$f_i = \rho_i f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}}), \tag{2.4}$$

where f_{travel} is defined as in Eq. (2.3) and $\rho_i \in [0,1]$ is an adaptive gain meant to scale down the intensity of the action of $f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}})$ whenever

1. its alignment is too conflicting with the generalized connectivity force f_i^λ or
2. the 'prime traveler' is not able to efficiently travel along its path because its reached speed is too low compared to its desired cruise speed.

Section 2.4.7 is dedicated to provide details on choosing an effective ρ_i.

case `anchor`. If state_i is set to `anchor`, the estimate $\hat{\Lambda}_p^i$ is again updated using a consensus-like protocol (line 13). Then (line 14) the force f_i is set as

$$f_i = f_{\text{anchor}}(q_i, z_i, R_z) = -\frac{\partial V^{R_z}_{\text{anchor}}(\|q_i - z_i\|)}{\partial q_i} \tag{2.5}$$

where $V^{R_z}_{\text{anchor}} : [0, R_z) \to [0, \infty)$ is a monotonically increasing potential function of the distance $\ell_i = \|q_i - z_i\|$ between the robot position q_i and the target z_i, and such that $V^{R_z}_{\text{anchor}}(0) = 0$ and $\lim_{\ell_i \nearrow R_z} V^{R_z}_{\text{anchor}}(\ell_i) = \infty$. Under the action of $f_{\text{anchor}}(q_i, z_i, R_z)$ the position q_i is then guaranteed to remain confined within a sphere of radius R_z centered at z_i until the local task at the target location is completed. In our simulations and experiments we employed

$$V^{R_z}_{\text{anchor}}(\ell_i) = -k_z \frac{2R_z}{\pi} \ln\left(\cos\left(\frac{\ell_i \pi}{2R_z}\right)\right) \tag{2.6}$$

where k_z is an arbitrary positive constant. The shape of this function is shown in Fig. 2.3 and the associated f_{anchor} is

$$f_{\text{anchor}}(q_i, z_i, R_z) = -k_z \tan\left(\frac{\ell_i \pi}{2R_z}\right) \frac{q_i - z_i}{\ell_i}. \tag{2.7}$$

2.4.7 Traveling Efficiency, Force Alignment and Adaptive Gain

We now describe how the estimation of the *traveling efficiency* Λ_i of all robots and the *adaptive gain* ρ_i of a 'secondary traveler', used in Algorithm 2, are actually computed. Remember that the idea behind the gain ρ_i is to adaptively scale down the traveling force f_i of a 'secondary traveler' whenever

1. the alignment of f_i and the generalized connectivity force f_i^λ is too different, or
2. the traveling efficiency of the 'prime traveler' is too low.

Therefore the design of ρ_i aims at guaranteeing that the current 'prime traveler' can always reach its target, whatever the motion planned by the other robots in the group are, and thus in fact enforces Assumption 1.

We recall that we provide a compendium of all important variables in Table 2.1.

In order to implement the desired behavior we introduce two functions:

$$\Theta : \mathbb{R}^3 \times \mathbb{R}^3 \to [0,1]$$
$$\Lambda : \mathbb{R}_0^+ \times K^* \to [0,1]$$

where $K^* = \{(x_c, x_M) \in \mathbb{R}^2 \,|\, 0 \le x_c < x_M\}$, defined as:

$$\Theta(x,y) = \begin{cases} \frac{1}{2}\left(1 + \frac{x^T y}{\|x\|\|y\|}\right) & x \neq 0, y \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

$$\Lambda(x,x_c,x_M) = \begin{cases} 1 & x \in [0,x_c] \\ \frac{1}{2} + \frac{\cos\left(\frac{x-x_c}{x_M-x_c}\pi\right)}{2} & x \in (x_c,x_M) \\ 0 & x \in [x_M,\infty). \end{cases} \tag{2.9}$$

Function $\Theta(x,y)$ represents a 'measure' of the direction alignment of the two non-zero 3D vectors x and y. In particular, $\Theta(x,y)$ is 1 if x and y are parallel with the same direction, $\frac{1}{2}$ if they are orthogonal, and 0 if they are parallel with opposite direction. Note that $\Theta(x,y)$ is equivalent to $\frac{1}{2}(1+\cos\theta)$ with θ being the angle between vectors x and y.

Function $\Lambda(x,x_c,x_M)$ 'measures' how small x is. If $x \le x_c$ then x is considered 'small enough' and, therefore, $\Lambda = 1$. If $x \in (x_c,x_M)$ then Λ strictly monotonically varies from 1 to 0. If $x \ge x_M$, then $\Lambda = 0$. The shape of Λ is depicted in Fig. 2.4.

Having introduced these functions, we now define the *force direction alignment* of the i-th robot as

$$\Theta_i = \Theta(f_i^\lambda, f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}})), \tag{2.10}$$

and note that Θ_i can be locally computed by the i-th robot. The quantity Θ_i thus represents an index in $[0,1]$ measuring the degree of conflict among the directions of the generalized connectivity force and the traveling force.

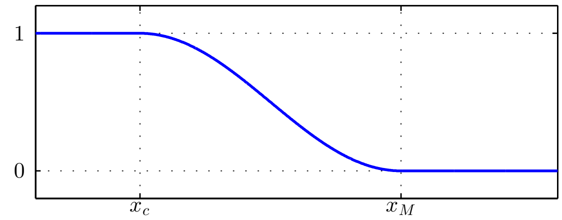


Figure 2.4: Sketch of the function $\Lambda(x,x_c,x_M)$ for fixed x_c and x_M.

When $\gamma_i \neq \texttt{null}$, we also define the absolute tracking error as

$$e_i = (1-\alpha_\Lambda)\|v_i^\gamma(v_i^{\text{cruise}}, q_i^\gamma) - v_i\| + \alpha_\Lambda \|q_i^\gamma - q_i\|, \tag{2.11}$$

with $\alpha_\Lambda \in [0,1]$ being a constant parameter modulating the importance of the velocity tracking error w.r.t. the position tracking error. The *traveling efficiency* is then defined as

$$\Lambda_i = \Lambda\left(e_i, x_c, x_M\right), \tag{2.12}$$

where $0 \leq x_c < x_M < \infty$ are two user-defined thresholds representing the point at which the traveling efficiency Λ_i starts to decrease and the maximum tolerated error after which the traveling efficiency vanishes. In this way it is possible to evaluate how well a traveler can follow its desired planned path according to a suitable combination of velocity and position accuracy. It is important to note that the value $\Lambda_i = 1$ does not imply an exact tracking of the path, but it still allows a small tracking tolerance (dependent on the parameter x_c). Similarly, the value $\Lambda_i = 0$ does not imply a complete loss of path tracking, but it represents the possibility of a tracking error higher than a maximum threshold (dependent on x_M).

In order to meet Assumption 1, we are only interested in the traveling efficiency of the current 'prime traveler' for monitoring whether (and how much) its exploration task is held back by the presence/motion of the 'secondary travelers'. From now on we then denote this value as Λ_p, where

$$p = i \quad s.t. \;\; \mathsf{state}_i = \texttt{prime-traveler}.$$

This quantity is not in general locally available to every robot in the group, and therefore a simple decentralized algorithm is used for its propagation to avoid a flooding step. Among many possible choices we opted for using the following well-known consensus-based propagation [Olfati-Saber and Murray, 2003]:

$$\begin{aligned} \dot{\hat{\Lambda}}_p^i &= k_\Lambda \sum_{j \in \mathcal{N}_i} (\hat{\Lambda}_p^j - \hat{\Lambda}_p^i) && \text{if } i \neq p \\ \hat{\Lambda}_p^i &= \Lambda_i && \text{if } i = p. \end{aligned} \tag{2.13}$$

This distributed estimator lets $\hat{\Lambda}_p^i$ track Λ_p for all i that hold $\mathsf{state}_i \neq \texttt{prime-traveler}$ with an accuracy depending on the chosen gain k_Λ. Notice that, for a constant Λ_p, the convergence of this estimation scheme is exact. Furthermore, since $\Lambda_p \in [0,1]$, $\hat{\Lambda}_p^i$ is then saturated so as to remain in the allowed interval despite the possible transient oscillations of the estimator. Instead of this simple consensus, one could also resort to a PI average consensus estimator [Freeman et al., 2006] to cope with presence of a time-varying signal. However, for simplicity we relied on a simple consensus law with less parameters to be tuned, and with, nevertheless, a satisfying performance as extensively shown in our simulation and experimental results.

Hence, every 'secondary traveler' can locally compute Θ_i and build an estimation $\hat{\Lambda}_p^i$ of Λ_p. In order to consolidate these two quantities into a single value, we define the function $\rho : [0,1] \times [0,1] \times [1,\infty) \to [0,1]$ as:

$$\rho(x,y,\sigma) = (1-x)y^\sigma + x(1-(1-y)^\sigma), \tag{2.14}$$

where $1 \leq \sigma < \infty$ is a constant parameter. Gain ρ_i is then obtained from Θ_i and $\hat{\Lambda}_p^i$ as

$$\rho_i = \rho(\Theta_i, \hat{\Lambda}_p^i, \sigma) \tag{2.15}$$

with $1 \leq \sigma < \infty$ being a tunable parameter.

The reasons motivating this design of gain ρ_i are as follows: ρ_i is a smooth function of Θ_i and $\hat{\Lambda}_p^i$ possessing the following desired properties (see also Fig. 2.5)

1. $\hat{\Lambda}_p^i = 1 \Rightarrow \rho_i = 1$: if the traveling efficiency of the 'prime traveler' is 1 then every 'secondary traveler' sets $f_i = f_{\text{travel}}(q_i, \gamma_i, v_i^{\text{cruise}})$;
2. $\hat{\Lambda}_p^i = 0 \Rightarrow \rho_i = 0$: if the traveling efficiency of the 'prime traveler' is 0 then every 'secondary traveler' sets $f_i = 0$;
3. ρ_i monotonically increases w.r.t. $\hat{\Lambda}_p^i$ for any Θ_i and σ in their domains;
4. ρ_i constantly increases w.r.t. Θ_i for any $\hat{\Lambda}_p^i \in (0,1)$ and $\sigma > 1$;
5. if $\sigma = 1$ then $\rho_i = \hat{\Lambda}_p^i$ for any $\Theta_i \in [0,1]$
6. if $\sigma \to \infty$ then $\rho_i \to \Theta_i$ for any $\hat{\Lambda}_p^i \in (0,1)$.

Summarizing, gain ρ_i mixes the information of both the force direction alignment and the traveling efficiency of the 'prime traveler', with more emphasis on the first or the second term depending on the value of the parameter σ. Nevertheless, the traveling efficiency $\hat{\Lambda}_p^i$ is always predominant at its boundary values (0 and 1) regardless of the value of σ. This means that, whenever the estimated travel efficiency of the 'prime traveler' is $\hat{\Lambda}_p^i = 0$ and robot i is a 'secondary traveler', its traveling force is scaled to zero and, therefore, robot i only becomes subject to the connectivity and damping force. Therefore,

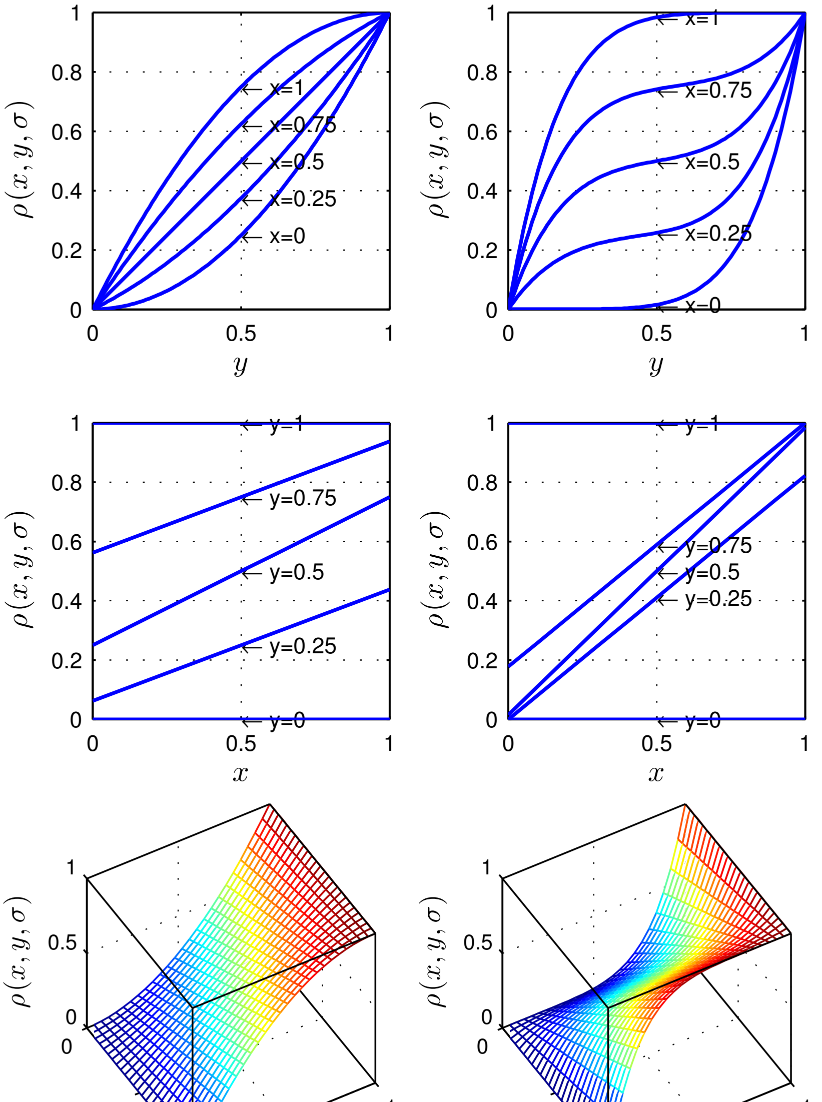


Figure 2.5: Function ρ for $\sigma = 2$ (left side) and $\sigma = 6$ (right side). The motion controller exploits this function by plugging the force direction alignment in the x argument, and the estimate of the traveling efficiency of the current 'prime traveler' in the y argument.

in this situation the motion of all 'secondary travelers' results dominated by the 'prime traveler', which is then able to execute its planned path towards its target location. On the other hand, when $\hat{\Lambda}_p^i = 1$, the 'prime traveler' has a sufficiently high traveling efficiency despite the 'secondary traveler' motions. Therefore, every 'secondary traveler' is free to travel along its own planned path regardless of the direction alignment between traveling and connectivity force.

We conclude noting that the main goal of the machinery defined in Secs. 2.4.6 and 2.4.7 is to ensure that the motion controller meets the requirements defined in Assumption 1. Although some of the steps involved in the design of the traveling force f_i have a 'heuristic' nature, the proposed algorithm is quite effective in solving the multi-target exploration task (in a decentralized way) under the constraint of connectivity maintenance, as proven by the several simulation and experimental results reported in the next section.

2.5 Simulations and experiments

In this section, we report the results of an extensive simulative and experimental campaign meant to illustrate and validate the proposed method. The videos of the simulations and experiments can be watched on `https://homepages.laas.fr/afranchi/robotics/?q=node/144`.

All the simulation (and experimental) results were run in 3D environments, although only a 2D perspective is reported in the videos for the simulated cases (therefore, robots that may look as 'colliding' are actually flying at different heights, since their generalized connectivity force prevents any possible inter-robot collision).

As robotic platform in both simulations and experiments we used small quadrotor UAVs (Unmanned Aerial Vehicles) with a diameter of 0.5 m. This choice is motivated by the versatility and construction simplicity of these platforms, and also because of the good match with our assumption of being able to track any sufficiently smooth linear trajectory in 3D space.

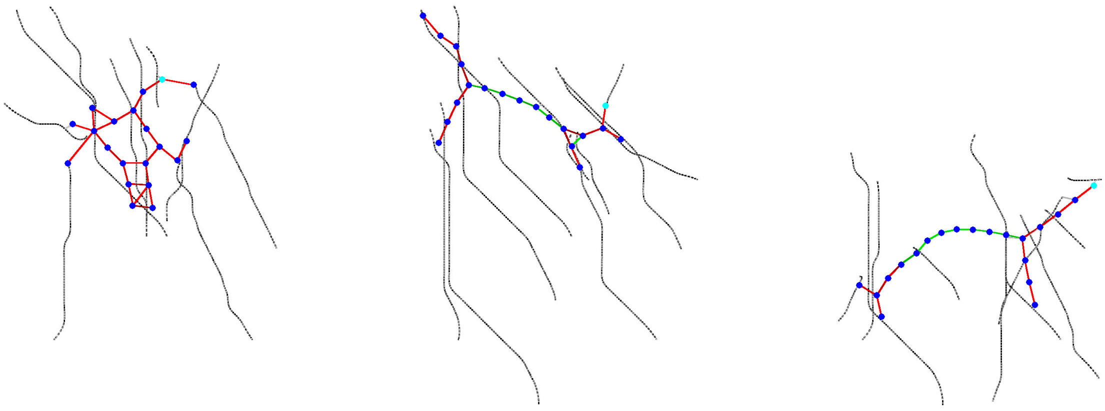
We further made use of the SwarmSimX environment [Lächele et al., 2012], a physically realistic simulation software. The simulated quadrotors are highly detailed models of the real quadrotors later employed in the experiments. The physical behavior of the robots itself and their interaction with the environment is simulated in real-time using PhysX[5].

For the experiments, we opted for a highly customized version of the MK-Quadro[6]. We implemented a software on the onboard microcontroller able to control the orientation of the robot by relying on the integrated inertial measurement unit. The desired orientation is provided via a serial connection by a position controller implemented within the ROS framework[7] that can run on any generic GNU-Linux machine. The machine can

[5] `http://www.geforce.com/hardware/technology/physx`

[6] `http://www.mikrokopter.com`

[7] `http://www.ros.org`



(a) (b) (c)

Figure 2.6: Snapshots of a simulation with 20 UAVs in empty space in three different consecutive time instants. The dotted black curves represent the planned path γ_i to the current target for each robot i (if it has a current target). Blue dots are the robots, the turquoise dot is the current 'prime traveler'. Line segments represent the presence of a connection link between a pair of robots with the following color coding: green – well connected, red – close to disconnection. The robots are able to concurrently explore the given targets and continuously maintain the connectivity of the interaction graph.

be either mounted onboard or acting as a base-station. In the latter case a wireless serial connection with XBees[8] is used. We opted for the separate base station in order to extend the flight time thanks to the reduction of the onboard weight. The current UAV position used by the controller is retrieved from a motion capturing system[9], while obstacles are defined statically before the task execution.

To abstract from simulations and experiments, we used the TeleKyb software framework, which is thoroughly described in Grabe et al. [2013]. Finally, the desired trajectory (consisting of position, velocity and acceleration) is generated by our decentralized control algorithm implemented using Simulink[10] running in real-time at 1 kHz.

2.5.1 Monte Carlo Simulations

The proposed method has been extensively evaluated through randomized experiments in three significantly different scenarios. The first scenario is an obstacle free 3D space and three snapshots of the evolution of the proposed algorithm are presented in Fig. 2.6. The second, a more complex, scenario includes a part of a town and is reported in Fig. 2.7.

[8] http://www.digi.com/lp/xbee

[9] http://www.vicon.com

[10] http://www.mathworks.com/products/simulink/

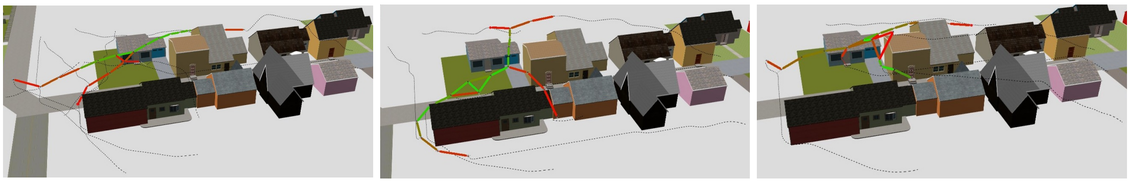


Figure 2.7: Three snapshots of consecutive time instants of a simulation in the town environment. Graphical notation similar to Fig. 2.6

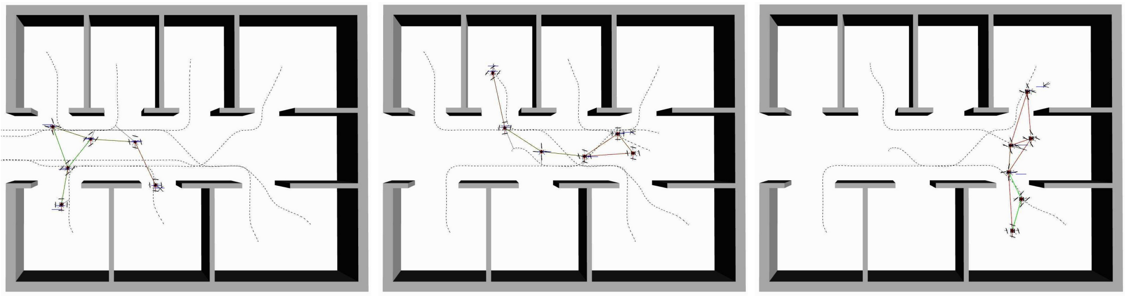


Figure 2.8: Snapshots of a simulation in the office-like environment in three consecutive time instants. Graphical notation similar to Fig. 2.6

The third is an office-like environment shown in Fig. 2.8. The size of the environments is 50 m × 70 m for both the empty space and the town, and about 10 m × 15 m for the office. Since the first two environments are outdoor scenarios and the office-like environment is indoor, two different sets of parameters were employed in the simulations. The values of the main parameters are listed in Table 2.2.

The number of robots varied from 10 to 35. In every trial 3 targets are sequentially assigned to 5 robots and 2 targets are sequentially assigned to other 5 robots, for a total of 25 targets per trial. The remaining robots are given no targets (i.e., they act always as 'connectors').

The configuration of the given targets is randomized across the different trials. The same random configurations are repeated for every different number of robots in order to allow for a fair comparison among the results. In the following we refer to the robots with at least one target assigned during a trial as 'explorers'.

To summarize, we simulated a total number of 1800 trials arranged in the following way: in each of the 3 scenes, and for each of the 100 target configurations in each scene, we ran a simulation with 6 different numbers of robots, namely 10, 15, 20, 25, 30, 35. We encourage the reader to also watch the video[11] where some representative simulative trials are shown.

In Fig. 2.9 we show the evolutions of the statistical percentiles of:

- the overall completion time,

[11] available at `https://homepages.laas.fr/afranchi/robotics/?q=node/144`

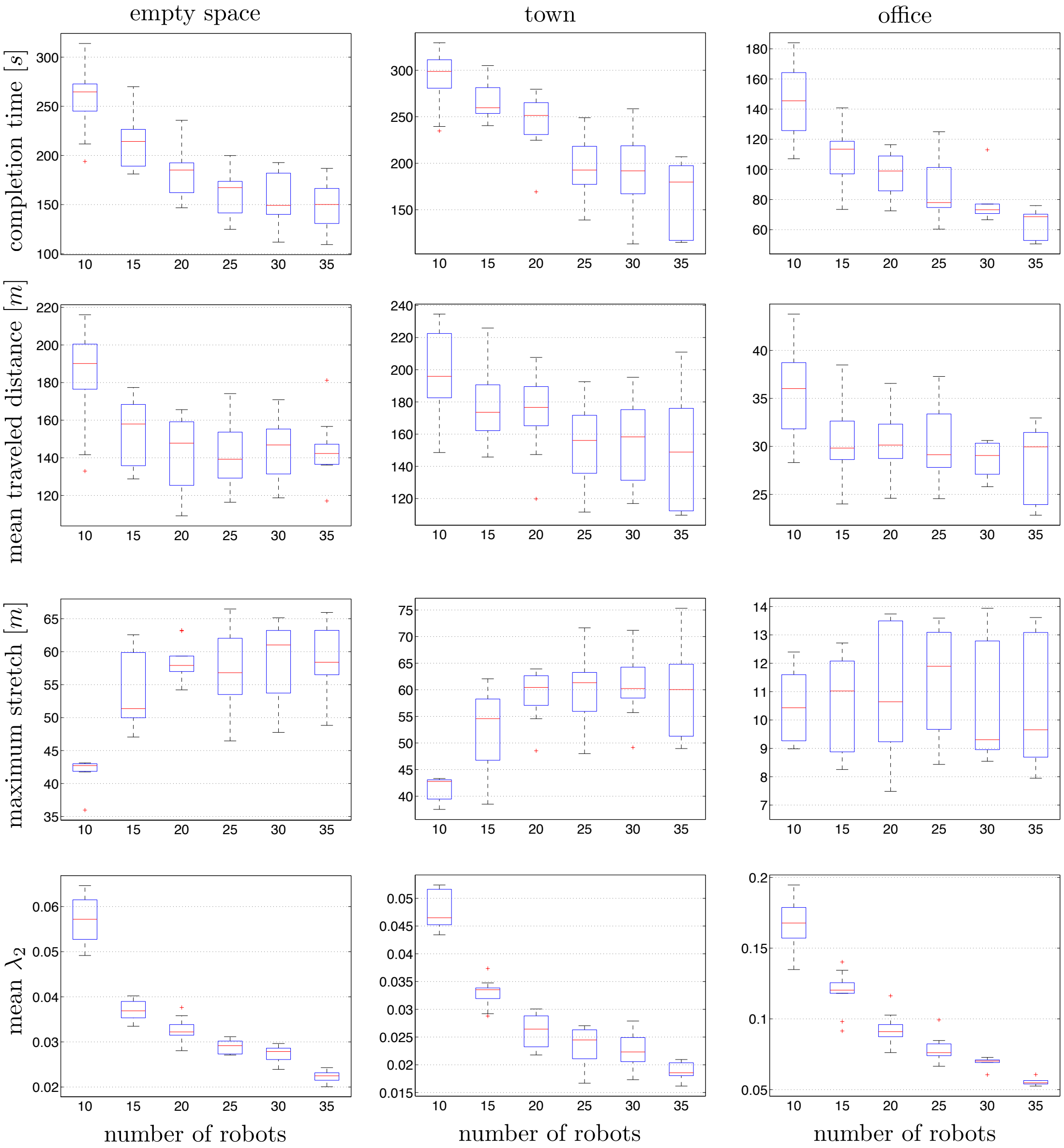


Figure 2.9: Statistics of the completion times (first row), mean traveled distance of the traveling robots (second row), the maximum Euclidian distance between two traveling robots (third row) and mean λ_2 (forth row) versus the number of robots in the environments empty space (left column), town (middle column) and office (right column).

Table 2.2: Main parameters of the algorithm used in the 1800 randomized simulative trials in the different scenarios.

parameter	empty space and town	office
(R'_s, R_s)	(2.5 m, 6 m)	(1.1 m, 2.5 m)
(R_o, R'_o)	(0.75 m, 1.75 m)	(0.25 m, 0.6 m)
(R_c, R'_c)	(1 m, 2.5 m)	(0.8 m, 1.1 m)
$(\lambda_2^{\min}, \lambda_2^{\text{null}})$	$(0,1)$	$(0,1)$
R_{grid}	0.75 m	0.25 m
σ	3	3
v_i^{cruise}	3 m/s	1 m/s for all i
(x_c, x_M)	$(0.1, 0.6)v_i^{\text{cruise}}$	$(0.1, 0.6)v_i^{\text{cruise}}$
Δt_i^k	3 s for all i and k	3 s for all i and k
R_z	1.8 m	1 m

- the mean traveled distance of the 10 ‘explorers’,
- the maximum Euclidean distance between two ‘explorers’
- the average of $\lambda_2(t)$ over time along the whole trial (we recall that the larger the λ_2 the more connected is the group of robots, refer to Appendix A),

when the number of robots varies from 10 (i.e., no ‘connectors’) to 35 (i.e., 25 ‘connectors’). Each column refers to one of the 3 different scenarios.

An improvement with the increasing number of ‘connectors’ in all scenarios is obvious. The mean completion time (first row) roughly halves when comparing 0 to 25 ‘connectors’. Adding more than 25 connectors will likely produce only minor improvement compared to the higher cost of having more robots, since the trend becomes basically flat. For this reason we did not perform simulations with a larger number of robots.

In the second row (mean traveled distance) one can see how, by already adding a few robots, a reduced mean traveled distance is obtained. This can be explained by the fact that the ‘connectors’ make the ‘explorers’ less disturbed by other ‘explorers’ with, therefore, more freedom to avoid unnecessary detours in reaching their targets.

Another measure of the reduced task completion time is the maximum stretch among the ‘explorers’ (i.e., the maximum Euclidean distance between any two ‘explorers’, see third row). The more connectors, the more stretch is allowed: ‘connectors’ in fact provide the support needed by the ‘explorers’ for keeping graph $\mathcal{G}$ connected while freely moving towards their targets. Only the office-like environment does not show this trend in the maximum stretch. This is due to the fact that the scene is relatively small and therefore the targets are not enough spread apart, so no bigger stretch is needed.

The increased freedom of the ‘explorers’ is also evident in the plots of the average $\lambda_2(t)$ (fourth row). These plots show how the ‘connectors’ are also useful to let the

Table 2.3: Main parameters used in the experiments.

parameter	value
(R'_s, R_s)	(1.4 m, 2.5 m)
(R_o, R'_o)	(0.5 m, 0.75 m)
(R_c, R'_c)	(1.0 m, 1.4 m)
$(\lambda_2^{\text{min}}, \lambda_2^{\text{null}})$	$(0, 1)$
R_{grid}	0.2 m
σ	3
v_i^{cruise}	0.5 m/s
(x_c, x_M)	$(0.2, 0.7)\, v_i^{\text{cruise}}$
Δt_i^k	3 s for all i and k
R_z	0.75 m

'explorers' move more freely even in small environments. In fact, the larger the amount of 'connectors', the lower the mean λ_2: with more connectors the 'explorers' are more able to simultaneously travel towards their targets, thus bringing the topology of the group closer to less connected topologies (i.e., closer to tree-like topologies where the explorers would be the leaves of the tree). Clearly, this effect is independent of the maximum stretch, in fact the average λ_2 follows this decreasing trend also in the third office-like environment (third column).

2.5.2 Experiments

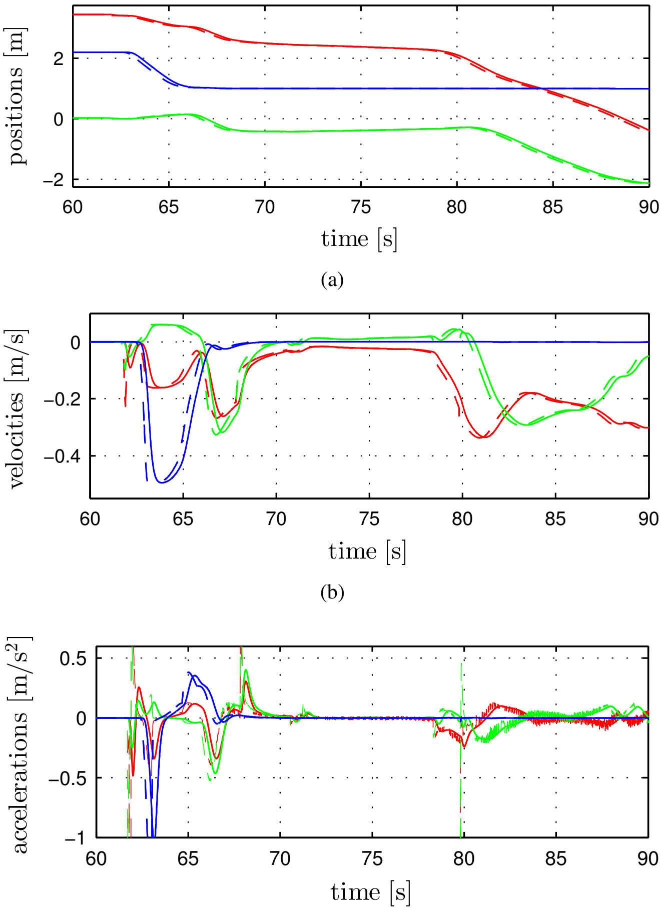
The experiments involved 6 real quadrotors and were meant to test the applicability of the algorithm in a real scenario. The parameters of the algorithm used in the experiments are reported in Table 2.3.

In order to obtain a $\bar{\mathcal{C}}^4$ trajectory smoother than $q_i(t)$ and, thus, better matching the dynamics capabilities of a quadrotor UAV [Mistler et al., 2001], we made use of a fourth order linear filter for each quadrotor:

$$\ddddot{q}_i^f(t) = -k_1 \dddot{q}_i^f(t) - k_2 \ddot{q}_i^f(t) - k_3 \dot{q}_i^f(t) + k_4 (q_i(t) - q_i^f(t)) \tag{2.16}$$

that tracks the position of the original trajectory $q_i(t)$, while keeping the velocity, acceleration, and jerk low in the filtered trajectory. The tunable gains were chosen as $k_1 = 44$, $k_2 = 707$, $k_3 = 5090$, $k_4 = 13692$ for placing the (real negative) poles at approximately -12, -13, -14, -15, then resulting in a settling time of about 0.3 s within a band of 5%.

The resulting trajectory $q_i^f(t)$ is then provided in place of $q_i(t)$ as input trajectory for the robot i as defined in Eq. (2.1), since it results very close to $q_i(t)$ as shown in Fig. 2.10a. However, at the same time, it provides a much smoother reference position signal to the



(a)

(b)

(c)

Figure 2.10: Position, velocity and acceleration of ‘explorer’ 1 during a representative period of the experiment, where $q_i(t), \dot{q}_i(t), \ddot{q}_i(t)$ are plotted in dash and $q_i^f(t), \dot{q}_i^f(t), \ddot{q}_i^f(t)$ as solid curves. The x, y and z component is plotted in red, green and blue respectively.

quadrotor by filtering off occasional abrupt motions, as can be seen in the velocity and acceleration reported in Figs. 2.10b and 2.10c. Figure 2.11 shows the norms of the UAV errors while tracking the desired trajectory $q_i^f(t)$. The average norm of all the quadrotors tracking errors during the whole experiment is 0.021 m, a few short peaks are above 0.06 m, and the highest peak is about 0.098 m.

For these experiments, we reproduced a scene similar to the office-like environment used in simulation, see Fig. 2.12. The UAVs with IDs ‘2’ and ‘4’ (called ‘explorers’) were given some targets, while the UAVs with IDs ‘1’, ‘3’, ‘5’, and ‘6’ (‘connectors’) had no target, for then a total of 6 quadrotors.

The *‘explorer’ 1* (with ID ‘4’) carries an onboard camera and has two targets in total. Whenever it reaches one of its targets it gives a human operator direct control of the vehicle in the surrounding area of the target. With the help of the onboard camera, the

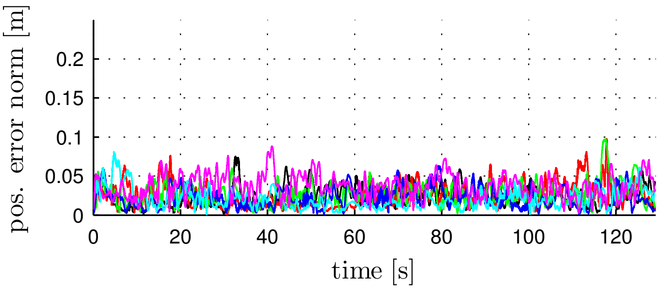


Figure 2.11: Plots of the 6 norms of the position error between $q_i(t)$ and the corresponding real quadrotor trajectory, for $i = 1, \ldots, 6$. The average error norm is 0.021 m.

human operator has the task of searching for an object in the environment. This could potentially be handled remotely as in Nestmeyer et al. [2013a], where rapid haptic feedback of the current swarm configuration improves awareness of the tele-operator. When the object is found by the human operator, the task at the target is considered completed, and the UAV switches back to autonomous control. In order to allow full human control of 'explorer' 1 in the anchoring behavior, the UAV is temporarily decoupled from the point q_4, which is instead kept close to the target by the action of f_{anchor} (as desired). The *'explorer' 2* (with ID '2') is instead fully autonomous and is assigned with a total of 4 targets. At the first target location, the task is to pick up an object to then be released at the second target location. The same task is subsequently repeated with targets 3 and 4. We note, however, that the pick and place action is only virtually performed since the employed quadrotors are not equipped with an onboard gripper. We also stress that all these operations are performed concurrently while keeping the topology of the group connected at all times.

A video of the experiment can be found on `https://homepages.laas.fr/afranchi/robotics/?q=node/144`.

Table 2.4 reports and describes all the relevant events taking place during an experiment in a chronological order.

Figure 2.12 shows the top-view of the 'explorer' paths for five representative time periods: $T_1 = [0, 25]$ s in Fig. 2.12a, $T_2 = [25, 60]$ s in Fig. 2.12b, $T_3 = [60, 80]$ s in Fig. 2.12c, $T_4 = [80, 120]$ s in Fig. 2.12d, and finally $T_5 = [120, 129]$ s in Fig. 2.12e. Every plot shows the (connected) graph topology of the group at the beginning of the time interval (dashed black lines) and the paths of the 2 'explorers' (solid lines, blue for the 'explorer' 1 and red for the 'explorer' 2). The initial positions of the robots are shown with colored circles and are labeled with the IDs of the corresponding robots. The two small blue squares represent the two desired target locations of the 'explorer' 1. The two green squares and the two red squares represent the two pick positions and release positions of 'explorer' 2, respectively. Finally, the vertical walls of the environment are shown in gray. Figure 2.12f on the other hand shows the z-coordinate of all the six quadrotors in order to understand the 3D motion in the 2D projections of Figs. 2.12a to 2.12e.

Table 2.4: Chronological list of important events in the experiment.

Fig. 2.12	Time	Events
(a)	0 s	The experiment starts. Both 'explorers' are assigned a target and since 'explorer' 2 is closer to its goal, it becomes 'prime traveler', while 'explorer' 1 is 'secondary traveler'.
	22 s	'Explorer' 2 arrives at its first target, where it should pick up an object. Therefore 'explorer' 2 goes into 'anchor' and 'explorer' 1 becomes 'prime traveler'.
(b)	29 s	'Explorer' 2 has completed the pick-up action and receives the point to release the object as a new target. Since 'explorer' 1 is still 'prime traveler', 'explorer' 2 becomes 'secondary traveler'.
	35 s	'Explorer' 1 arrives at its target, where the human operator takes control of the UAV and use its camera to find a yellow picture on the wall. 'Explorer' 2 then becomes 'prime traveler'.
	56 s	'Explorer' 2 arrives at the target where it needs to release the object.
(c)	63 s	'Explorer' 2 has completed the releasing action and receives the next pick-up location. 'Explorer' 1 is still under the control of the human operator and therefore in an 'anchor' state, so 'explorer' 2 directly becomes 'prime traveler'.
	65 s	The human operator finds the picture on the wall, 'explorer' 1 becomes autonomous again and starts to move towards its next target as 'secondary traveler', since 'explorer' 2 is 'prime traveler'.
	78 s	'Explorer' 2 arrives at the location where to pick up the second object and goes to the 'anchor' state. Hence 'explorer' 1 becomes 'prime traveler'.
(d)	85 s	'Explorer' 2 has completed the pick-up action and starts moving towards the releasing location as 'secondary traveler'.
	100 s	'explorer' 1 arrives at its target, goes to 'anchor' state and is thus under control of the human operator, therefore 'explorer' 2 becomes 'prime traveler'.
	119 s	'Explorer' 2 arrives at its final target and switches into 'anchor'.
(e)	123 s	The human operator finds the searched object and 'explorer' 1 becomes 'connector' since it has no new target location.
	126 s	'Explorer' 2 has completed the releasing action and becomes a 'connector' since it has also no new target.
	129 s	No UAV has a next target and the experiment ends.

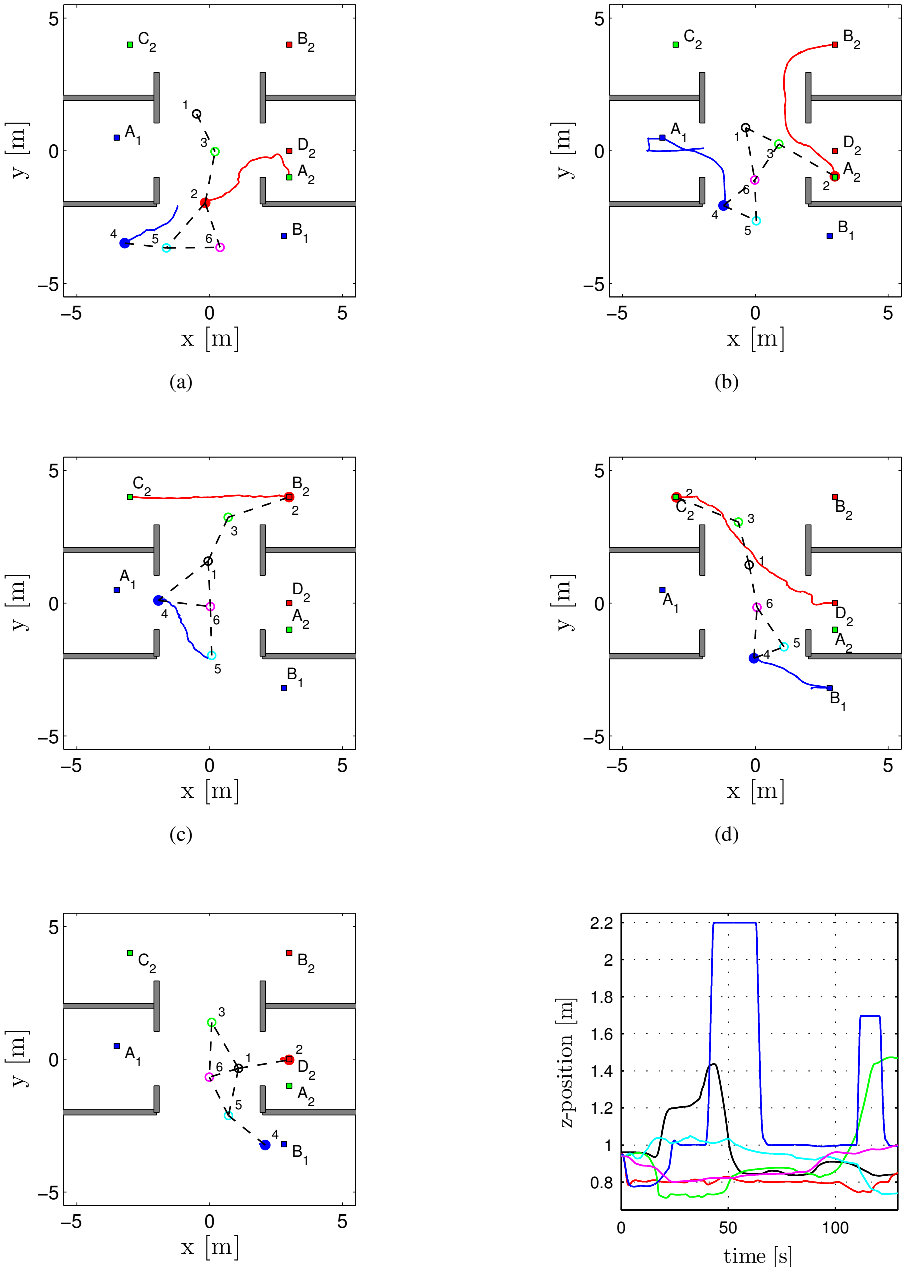


Figure 2.12: (a)-(e) Top view of the 3D paths of the ‘explorers’ (solid blue and red curves) during the experiment in five representative time intervals. The interaction graph at the beginning of each interval is shown with black dashed lines. The ID of each robot is shown besides the circle representing the starting position of each robot at the beginning of the corresponding interval. Targets are represented with colored squares and walls are gray. The specific time intervals are: (a) $T_1 = [0, 25]$ s, (b) $T_2 = [25, 60]$ s, (c) $T_3 = [60, 80]$ s, (d) $T_4 = [80, 120]$ s and (e) $T_5 = [120, 129]$ s. (f) z-coordinate of the positions of all six quadrotors to help interpreting the 2D projection reported in the plots (and videos). The large vertical motion of ‘explorer’ 1 (blue) is due to the human operator flying this robot, while the subsequent descent is autonomously performed thanks to the proposed algorithm.

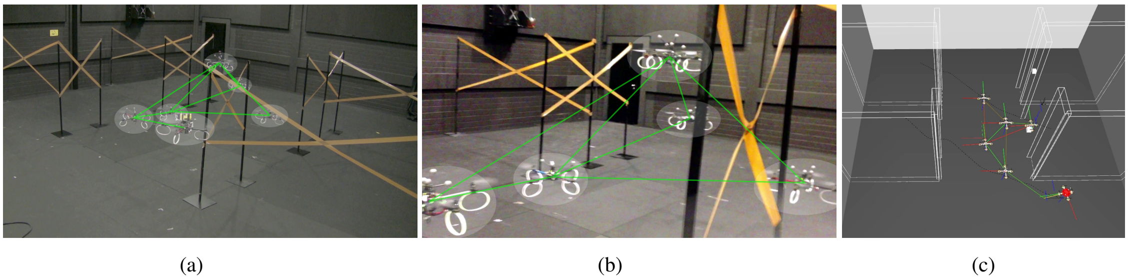

(a) (b) (c)

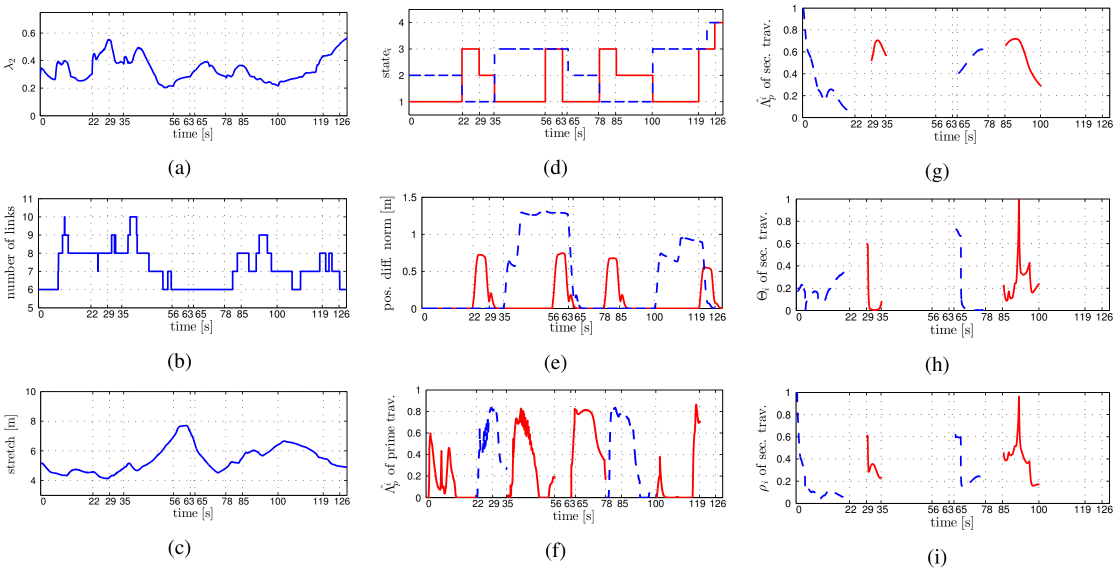
Figure 2.13: Three simultaneous screenshots of the experiment described in the text: (a) shows the side view of the scene from a fixed camera. Connections between UAVs (brightened areas) are overlayed as green lines. (b) shows the view taken from the onboard camera of the 'explorer' 1 using the same highlighting. (c) shows a 3D synthetic reconstruction of the robot positions and connections are shown with a line given in green when the weight is high, red shortly before a connection breaks and as a gradient in between. The robot that is marked with the red sphere is currently decoupled and controlled by the human operator.

Figure 2.13 shows three screenshots of the experiment: the lines between two quadrotors represent the corresponding connecting link as per graph $\mathcal{G}$.

Finally, Fig. 2.14 reports nine plots that capture the behavior of several quantities of interest throughout the whole experiment. As can be seen in Fig. 2.14a, the generalized algebraic connectivity eigenvalue $\lambda_2(t)$ (see Appendix A) remains positive for any $t > 0$, thus implying continuous connectivity of the graph $\mathcal{G}$ as desired. The time-varying number of edges in Fig. 2.14b shows the dynamic reconfiguration of the group topology which ranges between topologies with 5 edges (the minimum for having $\mathcal{G}$ connected) and topologies with up to 10 edges. This plot clearly shows how the adopted connectivity maintenance approach can cope with time-varying graphs. In Fig. 2.14c, we report the stretch of the group, defined as the maximum Euclidean distance between any two robots at a given time t. One can then appreciate how this stretch varies among 3.5 and 7.5 meters thus exploiting at most the allowable ranges of the experimental arena. Notice also how the stretch is in general larger when the number of links (and consequently $\lambda_2(t)$) is smaller. In fact the two peaks at about 60 s and 103 s occur when the robots are forced into a sparsely connected topology because the two 'explorers' have concurrently reached their farthest target pairs, i.e., (A_1, B_2) and (B_1, D_2).

Figure 2.14d shows the 'explorer' states state_2 and state_4 over time, with a dashed blue line and solid red line, respectively. In the plot, the following code is used: $1 =$ 'prime traveler', $2 =$ 'secondary traveler', $3 =$ 'anchor' and $4 =$ 'connector'. For $i = 1,3,5,6$ it is $\mathsf{state}_i = 4$ for all $t \in [0,129]$. Notice that, because of the algorithm design, at most one 'explorer' has $\mathsf{state}_i = 1$ at any given time.

The temporary decoupling of the 'explorers' from the points q_2 and q_4 during their anchoring behavior can be appreciated in Fig. 2.14e, where the Euclidian distance between the real robot position in the trajectory and the corresponding $q_i(t)$ is shown, for $i = 2,4$. 'Explorer' 1 (solid red line) decouples four times in total, in correspondence of the 2 pick-and-place operations, which gives rise to 4 short peaks in the plot. 'Ex-



(a) (d) (g)

(b) (e) (h)

(c) (f) (i)

Figure 2.14: Behavior of different measurements during an experiment: (a) λ_2 always stays greater than zero, thus showing how the group remains connected at all times, (b) the number of links $|\mathcal{E}(t)|$ of the interaction graph $\mathcal{G}(t)$, (c) the stretch of the formation given by the maximum Euclidean distance between any two quadrotors over time, (d) the exploration states with the following meaning: 1: 'prime traveler', 2: 'secondary traveler', 3: 'anchor', 4: 'connector', (e) the position difference between the virtual point of the connectivity maintenance and the commanded position to the quadrotors showing the decoupling as an 'anchor', (f) the traveling efficiency of the current 'prime traveler' (see Eq. (2.12)), (g) the estimation of the traveling efficiency by the 'secondary travelers' (see Eq. (2.13)), (h) the force direction alignment for the 'secondary travelers' (see Eq. (2.10)), (i) the adaptive gain used by the 'secondary travelers' to scale down their traveling force (see Eq. (2.15)).

plorer' 2 (dashed blue line) decouples two times in total, in correspondence of the 2 human-in-the-loop operations, causing 2 long peaks in the plot.

Figure 2.14f shows the traveling efficiency Λ_p of the current 'prime traveler' with a dashed blue line when 'explorer' 1 is the 'prime traveler' and with a solid red line when 'explorer' 2 is the 'prime traveler'. The estimation $\hat{\Lambda}_p^i$ of this value (see Eq. (2.13)) by all robots that are currently not 'prime traveler' is given in Fig. 2.15. We chose $k_\Lambda = 1$ resulting in a relatively slow propagation to show the additional robustness of our algorithm against this parameter (and the simple adopted consensus propagation), but clearly one could easily employ higher gains. To make it easier for the reader to understand the following discussion, we show again in Fig. 2.14g the essential information of this last plot whenever a robot is a 'secondary traveler'. In Figs. 2.14h and 2.14i the force direction alignment Θ_i (see Eq. (2.10)) and the adaptive gain ρ_i (see Eq. (2.15)) of the current 'secondary traveler' are shown. In the latter three plots a dashed blue line indicates when 'explorer' 1 is the 'secondary traveler', and a solid red line when 'explorer' 2 is the 'secondary traveler'.

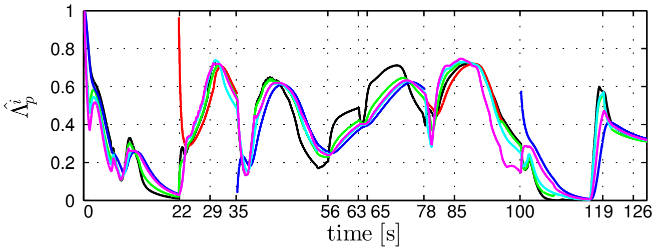


Figure 2.15: Estimation of the 'prime traveler' traveling efficiency of all the six robots, whenever they are currently not a 'prime traveler' (see Eq. (2.13)). The color scheme for the robots is as in Fig. 2.12.

To fully understand the important features of our method, we now give a detailed description of the time interval $[0,22]$ in the Figs. 2.14f to 2.14i. A similar pattern can then be found in the rest of the experiment. In this time interval, the 'explorer' 2 is the 'prime traveler', while 'explorer' 1 is a 'secondary traveler' (and the rest are 'connectors'). Due to the initial transient of its motion controller, the 'prime traveler' starts with $\Lambda_p = 0$ and quickly reaches $\Lambda_p = 0.6$. Shortly after, the traveling efficiency decreases again since 'explorer' 2 reaches the end of the area where it can freely move and, thus, needs to 'pull' the other robots for preserving connectivity of $\mathcal{G}$. This effect is propagated to the 'explorer' 1 as shown in Fig. 2.14g. The force direction alignment between the traveling force and the generalized connectivity force is shown in Fig. 2.14h. Combining these two plots with Eq. (2.15) allows to understand the effect of Fig. 2.14i. As can be seen, the 'secondary traveler' slows down its motion to around 10% for roughly 5 seconds. This enables the 'prime traveler' to travel faster again (see Fig. 2.14f). However, since the 'explorer' 2 needs to move around the wall (see Fig. 2.12a) to reach its target, it needs to 'pull' the other robots even more for preserving connectivity. Therefore, the traveling efficiency becomes zero and, although the direction alignment of the 'secondary traveler' becomes higher, the overall gain ρ_i stays very low: this makes it possible for the 'prime traveler' to eventually reach its target. We recall here that, according to Table 2.3 and Eq. (2.9), $\Lambda_p = 1$ as soon as the 'prime traveler' achieves a speed of at least 80% of its desired cruise speed (so the error is less than 20%), while $\Lambda_p = 0$ means a speed of less than 30% (an error of more than 70%), and *not necessarily* a zero velocity.

2.6 Conclusions

In this chapter, we presented a novel distributed and decentralized control strategy that enables simultaneous multi-target exploration while ensuring a time-varying connected topology in a 3D cluttered environment. We provided a detailed description of our algorithm which effectively exploits presence of *four* dynamic roles for the robots in the group. In particular, a 'connector' is a robot with no active target, an 'anchor' a robot close to its desired location, and all other robots are instead moving towards their targets.

Presence of at most one ‘prime traveler’, holding a leader virtue, is always guaranteed. All other robots (‘secondary travelers’) are bound to adapt their motion plan so as to facilitate the ‘prime traveler’ visiting task. This feature ensures that the ‘prime traveler’ is always able to reach its target, and thus ultimately allows to conclude completeness of the exploration strategy. The scalability and effectiveness of the proposed method was shown by presenting a complete and extensive set of simulative results, as well as an experimental validation with real robots for further demonstrating the practical feasibility of our approach.

2.6.1 Future work

As future development, the control of the ‘connectors’ could be modified to actively help improving the connectivity (e.g., moving towards the center of the group or towards the closest ‘explorer’) and therefore decrease the overall completion time even more. Another extension could include imposing temporal targets that expire before any robot can possibly reach them. In our framework this could be easily achieved by letting the corresponding ‘prime traveler’ or ‘secondary traveler’ switching into a ‘connector’ whenever a target expires, for then automatically starting to explore the next target (if any).

An important direction worth of investigation would also be the possibility to (explicitly) deal with errors or uncertainties in the relative position measurements (w.r.t. robots and obstacles) needed by the algorithm. Indeed, the presented results rely on an accurate measurement of relative robot and obstacle positions obtained by means of an external motion capture system.

Another improvement could address the distributed election of the ‘prime traveler’ as was already discussed in Sec. 2.4.3. Indeed, while the adopted flooding approach does not require presence of a centralized planning unit, it still needs to take into account information from all robots. It would obviously be preferable to only exploit information available to the robot itself and its 1-hop neighbors. This could be achieved by leveraging some (suitable variant of the) consensus algorithm as done for the decentralized propagation of the traveling efficiency of the current ‘prime traveler’. More generally, it might also be beneficial to improve the election of the ‘prime traveler’ by considering other criteria than the Euclidean distance w.r.t. a target which may not always result in an ‘optimal’ group motion (e.g., when obstacles, such as a wall, are present between the next ‘prime traveler’ and the target). The election could for instance choose the robot with the highest chance of reducing even further the completion time, e.g., based on the current motion of the group or direction of the majority of current targets of all ‘secondary travelers’.

Finally, it would be interesting to obtain an analytical upper bound of the total exploration time for our approach, although, in our opinion, deriving such a bound is unfortunately not so straightforward. Clearly, the considered multi-target exploration scenario has some analogies with the multiple traveling salesman problem [Bektas, 2006], where a certain number N of agents are asked to find a set of N shortest routes through a set of m cities and return back to the start. Nevertheless, an analysis based on the multiple traveling salesman problem would not easily extend to our case because of the constraint of continuous connectivity maintenance.

2.6.2 Autonomous Vision System

The 'explorer' 1 in Sec. 2.5.2, which carries an onboard camera, was operated by a human while being an 'anchor'. It would certainly be desirable to achieve a fully independent setup, without having a human as intermediary. Hence, the robot in our example would need to fully autonomously find the desired object through perceiving its camera stream, to achieve a truely intelligent system.

A major benefit of the human visual system is that we perceive the world mostly invariant to illumination [Marr, 1982]. Inspired by this light independent description of the scene, many artificial intelligence systems infer intrinsic images in their pipeline, like shadow removal [Kumar, 2011], object recoloring [Beigpour and Van de Weijer, 2011], interactive image and video editing [Bonneel et al., 2014] and autonomous driving [Maddern et al., 2014]. This motivates us to look deeper into the area of computer vision and especially the inference of light-invariant intrinsic images in the next part of my thesis.

Part II

Computer Vision

Chapter 3

Preliminaries

In this chapter, we introduce the fundamental concept of computer vision. Then, we look into the formation of an image through light transport and how humans perceive the world. Afterwards, we clarify what (convolutional) neural networks, currently the strongest contender of machine learning in computer vision, are. This gives us the appropriate preliminaries to fully grasp the ideas of the coming chapters.

3.1 Computer Vision as Inverse Computer Graphics

One of the big goals in computer vision is to invert the image formation process in order to infer properties of the real 3D world from the 2D projection in the image. On the one hand, Computer Graphics tries to simulate the real world through images or video, by taking a 3D model with material definitions and asking the question how to realistically render the scene into a 2D projection. In Computer Vision on the other hand, one tries to inverse the graphics approach by taking a 2D image and accurately recover information about the 3D model. Intrinsic Images as subfield describes the problem of recovering the latent space of range, orientation, reflectance, and incident illumination, according to Barrow and Tenenbaum [1978]. We want to focus first on the recovery of the intrinsic layers *reflectance* (describing the objects' material properties) and *shading* (a combination of surface orientation and illumination) in Chapters 4 and 5 and how this decomposition can be used for relighting in Chapter 7. Therefore, we first need to understand how light interacts with objects in our world.

3.2 Light Transport

3.2.1 Physical properties of light

We first define some notation to have a solid knowledge about the physical concept of light transport as defined by Horn [1986]:

Irradiance The amount of light falling on a surface. It is the power per unit area ($\mathrm{W\,m^{-2}}$ – watts per square meter) incident on the surface.

Radiance The amount of light radiated from a surface. It is the power per unit area per unit solid angle ($\mathrm{W\,m^{-2}\,sr^{-1}}$ – watts per square meter per steradian) emitted from the surface.

Luminance The intensity of light emitted from a surface per unit area in a given direction. Therefore, it is one specific angle of radiance, while the angle of interest is most commonly the angle subtended by the eye or camera.

3.2.2 Physical Measurements and Perceived Sensations

We have to distinguish between the physical properties of light and the color perceived by the brain. While, from a physical perspective, color can be described exactly by measuring the spectral power distribution (the intensity of the visible electro-magnetic radiation), most of the signal gets lost when perceived through the eye. The retina samples color in three broad bands only, roughly corresponding to red, green and blue light. The brain combines the signals from the cones (cells sensitive to those colors) and the rods (cells sensitive to intensity only). Therefore, so-called metamers exist, colors which through the eye appear to be the same, although in fact they have a different spectral composition [Horn, 1986]. This leads to different sensations of perceived color, dependent on the viewer/imaging system.

3.2.3 Human Perception

The 'International Commission of Illumination' (CIE, from French: 'Commission internationale de l'éclairage') defined the following sensations of human perception [Hunt and Pointer, 1987]:

Brightness the human sensation by which an area exhibits more or less light.

Hue the human sensation according to which an area appears to be similar to one, or to proportions of two, of the perceived colors red, yellow, green and blue.

Colourfulness the human sensation according to which an area appears to exhibit more or less of its hue.

Lightness the sensation of an area's brightness relative to a reference white in the scene.

Chroma the colourfulness of an area relative to the brightness of a reference white.

Saturation the colourfulness of an area relative to its brightness.

Object color depends upon the ratios of light reflected from the various parts of the visual field rather than on the absolute amount of light reflected. Still, humans can intuitively figure out the separation of reflectance of objects from the shading of the scene:

> We are never in doubt whether we have before us a white or gray paper even under quite different conditions of illumination: in bright sunshine, overcast sky, in twilight, or by candle light, we have always almost the same sensation.
>
> Mach [1865]

3.2.4 The (General) Rendering Equation

To understand how an image is formed, we first look at how light interacts with a scene before it reaches the eye/camera.

An easy derivation shows that image irradiance is proportional to scene radiance [Horn, 1986]. Furthermore, the radiance of a surface will generally depend on the direction from which it is viewed as well as on the direction from which it is illuminated. It can be described by the rendering equation [Immel et al., 1986, Kajiya, 1986] that determines the color and brightness of a surface point x at time t:

$$L_o(x,\omega_o,\lambda,t) = L_e(x,\omega_o,\lambda,t) + \int_{\omega_i\in\Omega} f_r(x,\omega_i,\omega_o,\lambda,t)L_i(x,\omega_i,\lambda,t)\langle\mathbf{n},\omega_i\rangle\,\mathrm{d}\omega_i \tag{3.1}$$

It depends on the incoming angle ω_i of light with intensity L_i at wavelength λ, reflected towards the angle ω_o of outgoing light L_o and emitted light L_e, the surface normal $\mathbf{n}$ and the amount of reflected light f_r. The dot product in the integral models Lambert's (cosine) law:

> Dependence on the cosine of the incident angle comes directly from the dependence of the irradiance on that factor and so can be traced to the foreshortening of the surface as seen from the light source.
>
> Lambert [1760]

This general model allows us to describe even complex effects like subsurface scattering by letting the point x emit light, when it actually exits the material from a subsurface reflection which was transmitted at another surface location of the object.

3.2.5 Simplified Rendering Equation

Most objects do not emit light themselves or exhibit the phenomenon of subsurface scattering. It is the amount of light at a given wavelength λ in the (visible) spectrum that is reflected at a specific point x on the object in which we are then interested in. We can then describe the equation with a bidirectional reflectance distribution function (BRDF) and simplify to:

$$L_o(\omega_o) = \int_{\omega_i\in\Omega} f(\omega_i,\omega_o)L_i(\omega_i)\langle\mathbf{n},\omega_i\rangle\,\mathrm{d}\omega_i \tag{3.2}$$

For the BRDF, the Helmholtz reciprocity condition holds:

$$f(\omega_i, \omega_o) = f(\omega_o, \omega_i) \tag{3.3}$$

Many everyday surfaces have a reflectance function that combines two components, one matte and one specular [Marr, 1982]. In more detail, an ordinary BRDF has the following shape:

> The surface orientation that maximizes the diffuse reflection component is typically one for which the surface normal points at the light source. The surface orientation that maximizes the glossy component, on the other hand, is usually one for which the surface normal points about halfway between the light source and the viewer. Correspondingly, the reflectance map can have two maxima. Typically, the global maximum is at the glossy peak.
>
> Horn [1986]

3.2.6 Material Properties

In 3D, any direction ω can be described with two-dimensional spherical coordinates, i.e., their polar angle θ and azimuth ϕ, so the BRDF is a four-dimensional function dependent on $\omega_i = (\theta_i, \phi_i)$ and $\omega_o = (\theta_o, \phi_o)$. Materials are then classified by the dimensionality of the subspace they can be reduced to.

Lambertian In the simplest case of Lambertian materials, the BRDF equates to a constant and the apparent brightness of a Lambertian surface to an observer is the same regardless of the observer's angle of view [Ikeuchi, 2014]. This is the case for Spectralon [Goldstein et al., 1999], which is designed specifically for its properties. At least mostly, this also holds for paper, unfinished wood, snow and charcoal.

Isotropic Isotropic materials behave invariant to rotation and the BRDF is reduced to the three dimensions $\theta_i, \theta_o, \phi_{\text{diff}} = \phi_o - \phi_i$. For example plastic is a typical isotropic material.

Anisotropic The most complex class are anisotropic materials, like brushed metal, velvet and satin. They need the full potential of the 4D BRDF to describe its reflectance properties.

3.3 (Convolutional) Neural Networks

For a long time, computer vision systems were created by using prior knowledge to directly model the desired task. But, as Polanyi [1966] stated, the problem with this approach is that “we can know more than we can tell”. This fact, often called *Polanyi’s Paradox*, means that many of the tasks we perform rely on tacit, intuitive knowledge that is difficult to codify and automate. Facial recognition is only one out of many examples where tacit knowledge is obvious:

> We know a person’s face, and can recognize it among a thousand, indeed a million. Yet we usually cannot tell how we recognize a face we know, so most of this cannot be put into words. When you see a face, you are not conscious about your knowledge of the individual features (eye, nose, mouth), but you see and recognize the face as a whole.
>
> Lam [2000]

Polanyi’s Paradox shows the desire to establish machine learning systems that directly learn from examples of a particular problem with their correct answers that stem from expert human judgment. This approach is called *supervised learning*.

The year 2012, with the work of Krizhevsky et al. [2012], marked a change in the usability of supervised learning systems in computer vision heavily relying on Convolutional Neural Networks (CNNs), made possible by the combination of software and computational power through the use of GPUs for the paramount matrix multiplications. From this point on, many areas in the field of computer vision saw big improvements in performance. The benefit of using CNNs and applying the holy grail of ‘deep learning’ is that they can make better use of much larger datasets compared to other supervised learning methods.

In order to understand this approach, we will now summarize the definition of a (convolutional) neural network. For a more thorough introduction, refer to Goodfellow et al. [2016], amongst many others.

3.3.1 Multi layer perceptron

A neuron in an artificial neural network is loosely inspired by a neuron in the brain. It computes the mathematically simple

$$o_j = \varphi\left(\sum_{i=1}^{n} x_i w_{ij} + b_j\right),$$

a linear reweighting with w_{ij} of all the input signals x_i and a bias b_j, followed by a non-linear activation function φ (see Fig. 3.1) in a purely feedforward fashion. A small network consisting of a collection of several of those neurons connecting an input to an output layer, is called a perceptron [Rosenblatt, 1958].

The dilemma with such a simple network is that it can only compute linearly separable problems and therefore not even a very basic predicate function such as `XOR` [Minsky

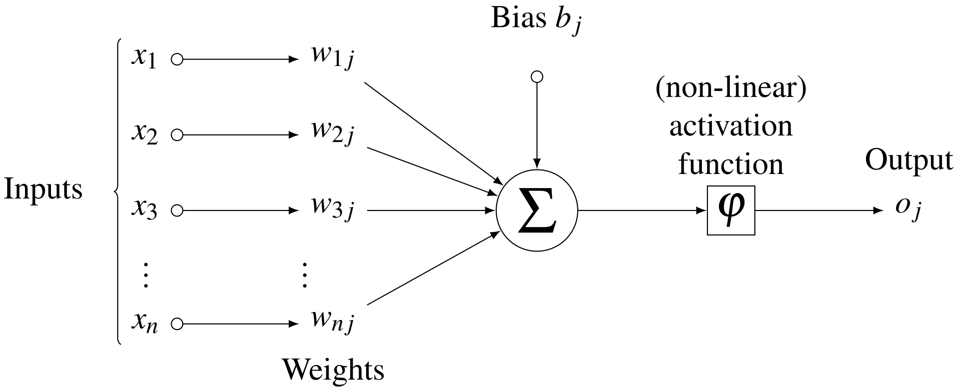


Figure 3.1: **Artificial neuron.** Model of the j-th artificial neuron in a perceptron.

et al., 1969]. This is why multilayer perceptrons (MLPs) were introduced, which consist of several hidden layers (see Fig. 3.2 for an example) of that same concept.

In fact, according to the universal approximation theorem [Hornik, 1991], a MLP can theoretically approximate any function arbitrarily well provided that the number of units is sufficiently large.

This is the foundation of neural networks, but there are three things to take into account to lead to the break through of neural networks in computer vision: the backpropagation algorithm to learn the weights, an appropriate activation function and the use of convolutions. We will discuss those topics in the next sections.

3.3.2 Backpropagation and Stochastic Gradient Descent

In the previous section, we defined the structure of the neural network, now we need to determine appropriate weights so that the overall net solves the desired task best, which in turn is measured by a loss/error/cost function E. We find the network parameters w_{ij} that result in the lowest loss by gradient descent on the loss function.

As in Fig. 3.1, we have

$$\mathrm{net}_j = \sum_{i=1}^{n} x_i w_{ij} \tag{3.4}$$

$$o_j = \varphi\left(\mathrm{net}_j\right), \tag{3.5}$$

therefore, to update the weights, we can use the chain rule to determine the gradient

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \mathrm{net}_j} \frac{\mathrm{net}_j}{\partial w_{ij}}. \tag{3.6}$$

For performance reasons, gradient descent is applied to mini-batches, which is referred to as stochastic gradient descent (SGD) [Robbins and Monro, 1951]. The high dimen-

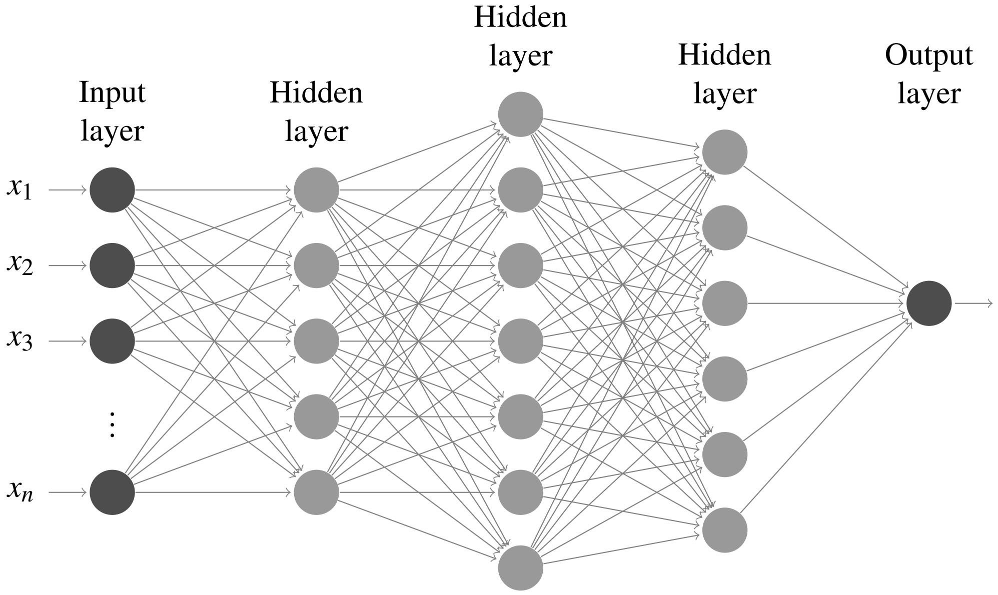


Figure 3.2: **Multilayer perceptron.** One example of a multilayer perceptron with three hidden layers. The bias terms are omitted for clarity.

sional error surface generally has many local minima, hence finding the right step size in SGD that leads to a global minimum is an active research topic. AdaDelta [Zeiler, 2012] and Adam [Kingma and Ba, 2015] are two popular recent approaches to automatically adapt the learning rate.

3.3.3 Choice of activation function

In the original definition of the perceptron, the activation function is determined by a step function, to give a binary output, which is supposed to resemble the firing of a neuron [Rosenblatt, 1958]. An obvious extension is to use functions with continuous range in order to also solve regression problems. Their shape, as shown in Fig. 3.3, is often motivated by the step function, so in earlier years, the most used ones were the *sigmoid/logistic function*

$$S(x) = \frac{1}{1 - \exp(-x)} \tag{3.7}$$

and *hyperbolic tangent*

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{3.8}$$

(see Figs. 3.3a and 3.3b for their visualization). Recent works deviated from that concept, after the *Rectified Linear Unit* [Nair and Hinton, 2010]

$$\mathrm{ReLU}(x) = \max(0, x) \tag{3.9}$$

was suggested (see Fig. 3.3c), which was one of the main ingredients that helped in the breakthrough of neural networks in computer vision [Krizhevsky et al., 2012]. It is a common belief that the introduced sparsity by a ReLU is key to good performance, but there is evidence that incorporating a non-zero slope for the negative part in rectified activation units could consistently improve results [Xu et al., 2015]. Therefore, many derived functions inspired by the ReLU exist nowadays:

- Parametric ReLU [He et al., 2015]:

$$\mathrm{PReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha \cdot x & \text{if } x \leq 0 \end{cases} \tag{3.10}$$

- Concatenated ReLU [Shang et al., 2016]:

$$\mathrm{CReLU}(x) = (\max(0, x), \max(0, -x)) \tag{3.11}$$

- Exponential Linear Unit [Clevert et al., 2015]:

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha \cdot (\exp(x) - 1) & \text{if } x \leq 0 \end{cases} \tag{3.12}$$

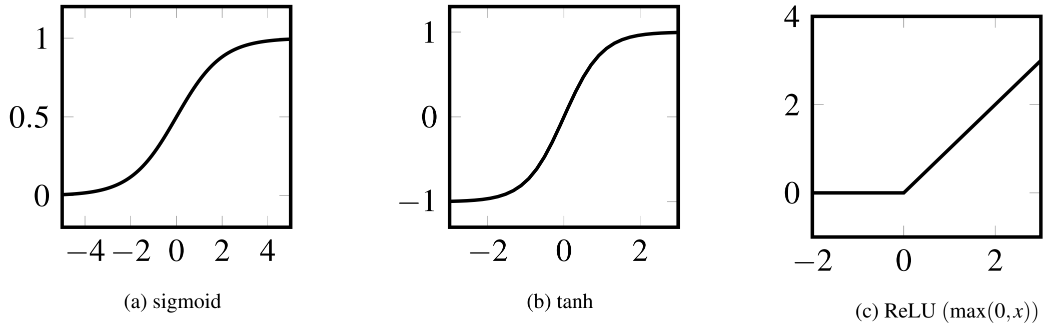
- Scaled ELU [Klambauer et al., 2017]:

$$\mathrm{SELU}(x) = \lambda \cdot \mathrm{ELU}(x) \tag{3.13}$$

Many of those activation functions prove to be beneficial for the specific task they were suggested for. Nonetheless, the simple ReLU is the predominantly employed nonlinearity [Rawat and Wang, 2017, Gu et al., 2018] that generally works well.

3.3.4 Convolutions

In images, each pixel can be represented with one neuron. While a multi layer perceptron, as described above, connects all neurons between two adjacent layers with learnable weights, in images the spatial structure of pixels can be exploited. Pixels are normally related to those in their proximity and less with pixels far apart in the image. In addition, objects have the same appearance in images, no matter where they are located in the



(a) sigmoid

(b) tanh

(c) ReLU $(\max(0,x))$

Figure 3.3: **Activation functions.** Three commonly employed non-linear activation functions.

image. This translational invariance and sparsity in connections leads to using convolutions [Fukushima, 1980, LeCun et al., 1999], where the same filter is applied in a sliding window fashion over the whole neural network layer. This greatly reduces the amount of parameters in the network, which leads to faster training and execution.

Several other approaches exist to learn the dependency structure across pixels, e.g., through learned bilateral filtering to achieve sparse high dimensional filters [Jampani et al., 2016, Gadde et al., 2016], in the form of non-parametric structured output networks [Lehrmann and Sigal, 2017] or in video data by utilizing temporal correspondences [He et al., 2018], nonetheless, convolutions keep being used as the standard solution in computer vision [Rawat and Wang, 2017, Gu et al., 2018].

Now, we have the prerequisites to dive deeper into learning to predict intrinsic images which we will talk about in the coming chapters.

Chapter 4

Direct CNN Prediction of Reflectance

In this chapter we want to focus on the recovery of the intrinsic layers reflectance and shading, under the Lambertian assumption.

Separating an image into reflectance and shading layers poses a challenge for learning approaches because no large corpus of precise and realistic ground truth decompositions exists. The Intrinsic Images in the Wild (IIW) dataset [Bell et al., 2014] provides a sparse set of relative human reflectance judgments, which serves as a standard benchmark for intrinsic images. A number of methods use IIW to learn statistical dependencies between the images and their reflectance layer. Although learning plays an important role for high performance, we show that a standard signal processing technique achieves performance on par with state-of-the-art. We propose a loss function for CNN learning of dense reflectance predictions. Our results show a simple pixel-wise decision, without any context or prior knowledge, is sufficient to provide a strong baseline on IIW. This sets a competitive baseline which only two other approaches surpass.

Contributions This chapter is partially published as [Nestmeyer and Gehler, 2017, Sec. 1 – 4] with additional material to provide a more thorough treatment.

4.1 Introduction

Almost 40 years ago, the seminal paper of Barrow and Tenenbaum conjectured that

> A robust visual system should be organized around a noncognitive, nonpurposive level of processing that attempts to recover an intrinsic description of the scene.
>
> Barrow and Tenenbaum [1978]

Their work motivates the task of decomposing an image into constituent layers such as surface reflectance, surface orientation, distance and incident illumination. Ever since, significant progress has been made on this problem, but the recovery of these physical properties of visual scenes or videos remains an open challenge.

> The central problem in recovering intrinsic scene characteristics is that the information is confounded in the original light-intensity image: a single intensity value encodes all the characteristics of the corresponding scene point.
>
> Barrow and Tenenbaum [1978]

A successful model needs to resolve the ill-posedness of the problem and cope with the variety of image appearances.

A possible line of attack are supervised learning methods which have been used with great success for a wide range of computer vision applications. Standing out for superior performance combined with favorable runtimes is the class of Convolutional Neural Network (CNNs), a dominant contender for many vision problems. CNNs are mostly falling into the category of purposive models, guided by task specific goals such as image classification or recognition. The obvious question is whether CNNs will fare equally well on the problem of intrinsic image decompositions.

Several works have included CNN methods in systems that recover reflectance and shading layers Narihira et al. [2015a,b], Zhou et al. [2015], Zoran et al. [2015]. However, prior work uses CNNs mostly in combination with additional methods, such as Conditional Random Fields (CRFs), to achieve a dense image decomposition. An advantage of CRF models is their ability to encode prior information about the problem. In the pre-CNN time, intrinsic image methods were dominated by CRF models with carefully designed priors on reflectance, shading, and their combination. In this and the next chapter, we attempt to answer the question whether prior terms are necessary when human annotation in the form of weak labels is available.

Acquiring accurate training data for intrinsic images is a challenge. The MIT intrinsic dataset [Grosse et al., 2009] with 20 images and 10 (single color) light configurations was a first attempt to empirically validate intrinsic estimation techniques. It has served this purpose well, but lacks realism and diversity. Recently, Beigpour et al. [2015] proposed an extension to multi illuminants, but without overcoming the limitations on extent. Another possible route to generate datasets is the use of computer graphics rendering engines. This has been explored by Beigpour et al. [2013] who created a dataset of synthetic scenes rendered using the Blender open source rendering engine [Blender, 2021]. This led to a dataset of 32 single objects and 36 scene compositions which is still limited in terms of detail and diversity. The MPI-Sintel dataset has been created using the open source movie Sintel [Butler et al., 2012] to serve as a benchmark for several problems such as optical flow estimation. While MPI-Sintel is more varied and complex, the type of scenes and visual appearance is still very different from real world data.

A significant attempt to overcome the lack of empirical data is the dataset of "Intrinsic Images in the Wild" (IIW) [Bell et al., 2014]. This dataset contains 5230 photos of mostly indoor scenes which have been annotated with a sparse set of relative reflectance judgments. From a small set of image locations, human judgments on pairs of neighboring locations have been collected, which provide whether one point is of darker or similar material reflectance. Although humans can be fooled with artificial setups [Adelson, 2000], the perception of relative material reflectance is sufficient to provide mostly consistent label information for this large corpus of images (see Bell et al. [2014] for an analysis). Along with the dataset, Bell et al. also formulate the *weighted human disagreement rate (WHDR)*, a performance metric that we will discuss in detail in Sec. 4.3.1. The IIW dataset allows to empirically validate intrinsic image estimation and the judgments have also been used to train models for intrinsic image decompositions [Narihira et al., 2015b, Zhou et al., 2015, Zoran et al., 2015].

In this and the next chapter, we develop two intrinsic image models: a CNN approach with appropriate loss function and a filtering technique to include strong prior knowledge about reflectance properties. We first design a CNN method that, in contrast to previous work, does not include prior information on shading smoothness [Land and McCann, 1971], reflectance [Omer and Werman, 2004, Gehler et al., 2011, Shen and Yeo, 2011, Bi et al., 2015], or combinations [Barron and Malik, 2015]. We design a loss function that enables end-to-end learning from the pairwise judgments. This leads to an interesting result: a simple multi-layer perceptron with no image context, just based on the pixels alone provides competitive performance, better or on par with current learning and non-learning models. We then develop a method from the other extreme, a dense filtering operation based on joint bilateral and guided filtering. This technique simplifies the processing pipeline of Bi et al. [2015] and makes it possible to apply to any reflectance prediction. Our experiments show drastically improved state-of-the-art performance on IIW. Besides presenting the empirically best performing algorithm, our results reveal interesting observations about the current state of intrinsic image estimation. In summary, we believe that for intrinsic image estimation, it is the inclusion of prior knowledge through regularization, CRFs, or filtering that still drives the performance. To rely solely on learning approaches, the amount of available annotation may still be insufficient.

4.2 Related Work

Until recently there was a lack of empirical data to validate intrinsic images algorithms. Therefore, most of the literature revolved around the design of suitable priors. The recent work of Barron and Malik [2015] is a prominent example of a method that carefully trades the use of prior information with interesting representations that enable a detailed decomposition into several layers. Priors in Barron and Malik [2015] include terms on smoothness, parsimony and absolute values of reflectance, smoothness, surface isotropy and occluding contour priors on shape and a multivariate Gaussian fit to a spherical-harmonic illumination model. This lead to impressive results on the MIT intrinsic dataset [Grosse et al., 2009], but the method is limited to single masked objects in a scene, and problems with complex illumination remain.

The work of Bi et al. [2015] approaches the problem from a filtering perspective. After a filtering step followed by clustering, the pixels are grouped into regions of same reflectance, such that a simple shading term suffices to recover the full intrinsic decomposition. This method produces the best results on the IIW dataset but takes several minutes of processing time. In Chapter 5 we build on this work and propose a filtering technique that can be applied to any other intrinsic image estimation as well. This implements the idea of grouping pixels into sets of constant reflectance. Other works consider additional knowledge in order to recover reflectance and shading, as, *e.g.*, multiple images of the same scene with different lighting [Weiss, 2001, Laffont and Bazin, 2015], an interactive setting with user annotations [Bousseau et al., 2009, Bonneel et al., 2014], or an additional depth layer as input [Chen and Koltun, 2013].

The paper of Bell et al. [2014] introduced the Intrinsic Images in the Wild dataset with human annotations giving relative reflectance judgments that served as the training and test set for different learning based methods. Using this data, the work of Bell et al. [2014] was the first to compare different algorithms on a large corpus of real world scenes.

A first attempt to learn using the data from IIW was made by Narihira et al. [2015b]. The authors used the relative judgment information in a multi-class setup and fine-tune an AlexNet CNN trained on ImageNet. Only the sparse annotation points that are required to compute the WHDR loss are predicted with this network and there is no step that turns them into a dense decomposition. The works of Zhou et al. [2015] and Zoran et al. [2015] are similar, both use a CNN to obtain pairwise judgment predictions, then followed by a step to turn the sparse information into a dense decomposition. Both methods achieve good results on IIW and take several seconds to process an image.

Similar to our work, in the sense that a dense intrinsic decomposition is predicted, is the work of Narihira et al. [2015a]. A CNN is used to directly predict reflectance and shading with the objective function being the difference to ground truth decompositions. Since those are only available for the rendered dataset of MPI-Sintel, the authors report that the learned model does not generalize well to the real world images of IIW. An additional data term in the gradient domain is used by Lettry et al. [2016]. They also propose to use an adversary in order to remove typical generative CNN artifacts by discriminating between generated and ground truth decompositions. Therefore, this approach has the same limitation requiring dense ground truth decompositions and no results on IIW are available. To our knowledge, there is no CNN based method that predicts a dense intrinsic decomposition and works well for images from IIW.

In Table 4.1 we organize the related work along the dimensions that are relevant for the proposed method, whereas in Table 4.2 we compile an overview on which of the related work presents qualitative or quantitative results on which dataset.

Similar in style, the work of Chen et al. [2016] also trains a CNN from relative judgments with a ranking loss to predict pixel-wise labels, but for the application of recovering dense depth estimates. This involved the creation of a dataset with relative depth judgments in the spirit of IIW. However, in contrast to intrinsic images, it is possible to capture accurate ground truth depth for training and testing, making reflectance and shading estimation a more relevant target of learning from sparse pairwise comparisons.

Table 4.1: **Overview of different intrinsic image estimation methods.** For every method we note whether or not it uses a *CNN* method, predicts a *dense decomposition* into intrinsic layers without an additional globalization step (Narihira et al. [2015b] only reports relative estimates for a sparse set of points), requires *no prior terms* based on man-made models and is *trained on IIW*.

Method	CNN	dense de-composition	no prior terms	trained on IIW
Shen and Yeo [2011]	✘	✔	✘	n/a
Garces et al. [2012]	✘	✔	✘	n/a
Zhao et al. [2012]	✘	✔	✘	n/a
Bonneel et al. [2014]	✘	✔	✘	n/a
Bell et al. [2014]	✘	✔	✘	n/a
Barron and Malik [2015]	✘	✔	✘	n/a
Bi et al. [2015]	✘	✔	✘	n/a
Narihira et al. [2015b]	✔	✘	✔	✔
Zhou et al. [2015]	✔	✔	✘	✔
Zoran et al. [2015]	✔	✔	✘	✔
Narihira et al. [2015a]	✔	✔	✔	✘
Lettry et al. [2016]	✔	✔	✔	✘
Our method	✔	✔	✔	✔

[IIW: Intrinsic Images in the Wild [Bell et al., 2014]]

Table 4.2: **Overview of evaluated datasets for qualitative and quantitative results by recent related work.** For each related work, we document if it provides qualitative and quantitative results on the relevant datasets.

Paper	qualitative results on			quantitative results on		
	IIW	MIT	MPI	IIW	MIT	MPI
Bell et al. [2014]	✔	✘	✘	✔	✔	✘
Barron and Malik [2015]	✘	✔	✘	✘	✔	✘
Bi et al. [2015]	✔	✘	✘	✔	✘	✘
Narihira et al. [2015b]	✘	✘	✘	✔	✘	✘
Zhou et al. [2015]	✔	✘	✘	✔	✘	✘
Zoran et al. [2015]	✔	✘	✘	✔	✘	✘
Narihira et al. [2015a]	✘	✔	✔	✔	✔	✔
Lettry et al. [2016]	✘	✔	✔	✘	✘	✔

[IIW: Intrinsic Images in the Wild [Bell et al., 2014]; MIT: MIT intrinsic dataset [Grosse et al., 2009]; MPI: MPI-Sintel [Butler et al., 2012]]

4.3 Preliminaries

We work with linear RGB and the Lambertian reflectance assumption, which allows to separate every pixel in image $I \in [0,1]^{3 \times h \times w}$ into a product of reflectance R and shading S, that is the pixel and channel-wise product $I = RS$. Further, we assume achromatic light which reduces the decomposition problem to a per-pixel scalar estimation problem. Namely, given a scalar $r_p \in [0,1]$ for each pixel p, we recover reflectance and shading as

$$R_p = \frac{r_p}{\frac{1}{3}\sum_c I_p^c} \cdot I_p, \qquad S_p = \frac{\frac{1}{3}\sum_c I_p^c}{r_p} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \tag{4.1}$$

where $c \in \{R,G,B\}$ denotes the color channel. Under these assumptions, the problem boils down to estimation of a single scalar per pixel $\mathbf{r} \in \mathbb{R}^{h \times w}$.

The same assumptions are commonplace in the literature and have been used, *e.g.*, in Gehler et al. [2011]. We note that achromatic light is often violated in the IIW dataset, especially in the presence of multiple light sources. As the proposed loss function WHDR only compares relative lightness and no color information, it is invariant to this choice.

4.3.1 A quantitative measure for intrinsic images

Accurate ground truth information in the form of image decompositions in reflectance and shading layers does not exist at scale. To empirically validate the quality of intrinsic image algorithms using the pairs of relative reflectance judgments alone, Bell et al. [2014] introduced the WHDR metric (weighted human disagreement rate). We refer to their work for all details on the data annotation process, but will review the ingredients that we need for our development.

For every image, annotation is given in the form of pairs of image locations (i_1, i_2) for which a human reflectance judgment $J_i \in \{1, 2, E\}$ is provided. The judgment indicates whether point i_1 is darker than i_2 ($J_i = 1$), lighter ($J_i = 2$), or of equal reflectance ($J_i = E$). The confidence w_i of a judgment is defined via the CUBAM score of the two-decision model "points have the same reflectance" and if not "does the darker point have darker reflectance" (see Bell et al. [2014] for further details). The annotation set $\{(i_1, i_2, J_i, w_i)\}_{i=1,\dots,N_I}$ varies in size N_I for every image I in the range from 1 to 1181 with a median of 113.

Given a reflectance prediction R, first a relative classification for the set of annotated points is computed as

$$\hat{J}_\delta(R,i) = \begin{cases} 1 & \text{if } R_{i_2}/R_{i_1} > 1+\delta \\ 2 & \text{if } R_{i_1}/R_{i_2} > 1+\delta \\ E & \text{else,} \end{cases} \tag{4.2}$$

where $\delta \geq 0$ controls when two points are considered different. For large values of δ, two points would need to be farther apart to be judged as darker (resp. lighter).

Given these relative estimates, the WHDR loss is computed as the weighted average of how often the annotation and prediction disagree

$$\mathrm{WHDR}_\delta(J,R) = \frac{\sum_i w_i \cdot \mathbb{1}\left(J_i \neq \hat{J}_\delta(R,i)\right)}{\sum_i w_i}, \tag{4.3}$$

and is regularly given in per cent. Note that this loss does not evaluate the reflectance at all points in the image, but only at those for which labels are available. Therefore, it could also be evaluated on these points alone for an algorithm that does not provide a dense decomposition of the image.

The works of Narihira et al. [2015b], Zhou et al. [2015], Zoran et al. [2015] use these relative annotations to train multi-class classifiers, predicting for every pair of patches its relative reflectance judgment $\{1,2,E\}$ directly. Since this approach does not provide the actual values R of the reflectance layer, further post-processing steps are required to produce a dense prediction. These post-processing steps are separate from the classifiers and motivated by common intrinsic prior terms. We will circumvent any post-processing by directly predicting a dense reflectance map R.

4.4 Direct Reflectance Prediction with a CNN

We propose an objective function that makes direct use of the relative reflectance judgments by humans that the IIW dataset provides. This weak label information has been used in Zhou et al. [2015], Zoran et al. [2015] for CNN training already, treating it however as a multi-class classification problem. While a multi-class loss achieves good performance on pairs of points, this strategy requires an additional globalization step to propagate information to all pixels. Our aim is to directly decompose the entire image with a single forward pass of a CNN, avoiding any need for post-processing.

We will first discuss the loss function that we use and then describe the network architecture and training method.

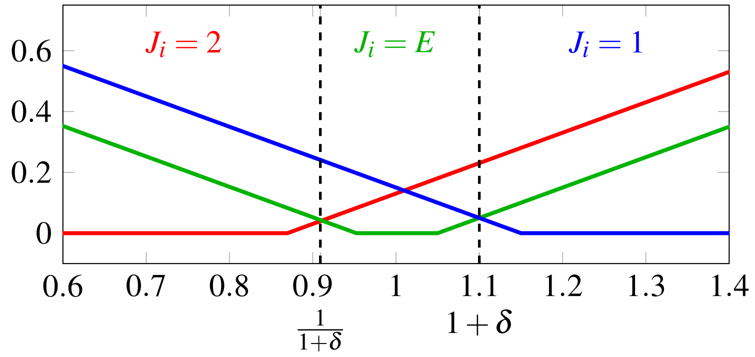


Figure 4.1: **Visualization of the WHDR-Hinge loss.** We visualize the WHDR-Hinge loss dependent on the ratio R_{i_1}/R_{i_2} for $\delta = 0.1$ and $\xi = 0.05$. The value of δ controls where the decision boundary for darker/lighter or equal reflectance lightness is made. With the value ξ, a margin from this boundary is encouraged. For values $\xi > \delta$ the E class will always have a non-zero loss.

4.4.1 WHDR-Hinge loss

We construct a proxy loss for the WHDR that can be used for supervised training. The formulation is an adaption of the ε-insensitive loss for regression Vapnik [1993] for this problem setup. We define

$$\ell_{\delta,\xi}(J,R,i) = \begin{cases} \max\left(0,\, \frac{R_{i_1}}{R_{i_2}} - \frac{1}{1+\delta+\xi}\right) & \text{if } J_i = 1 \\ \max\left(0,\, \frac{1}{1+\delta-\xi} - \frac{R_{i_1}}{R_{i_2}}, \frac{R_{i_1}}{R_{i_2}} - (1+\delta-\xi)\right) & \text{if } J_i = E \\ \max\left(0,\, 1+\delta+\xi - \frac{R_{i_1}}{R_{i_2}}\right) & \text{if } J_i = 2, \end{cases} \tag{4.4}$$

which is visualized in Fig. 4.1. The scalar δ is the threshold of the WDHR$_\delta$ and we introduce the hyper-parameter ξ, which is the margin between the neighbouring classes $1,E$ and $2,E$.

The pipeline of supervised training is simple. A network produces a dense decomposition R, which is then used to arrive at relative judgments for two pixel locations based on the ratio of the predicted R values. The loss in Eq. (4.4) is then weighted and summed over all annotated pixel pairs, similar to Eq. (4.3), and the error is propagated backwards to compute the gradients of the network parameters.

As with the standard hinge-loss commonly used for binary SVM training, the subgradients of the WHDR-hinge loss can be easily computed. For completeness, the derivation is provided in Appendix B.1.

4.4.2 Train and test data set

The IIW dataset does not come with a pre-defined train, validation and test split. We adopt the split suggested by Narihira et al. [2015b] into 80% training and 20% test images, putting the first of every five images sorted by file name in the test set. In order to properly evaluate different models, we additionally split the data into a separate validation set, with the ratios of 70% training, 10% validation and 20% test. We keep the test set of Narihira et al. [2015b], and, inspired from its selection, use from every series of 10 images the seventh in the validation set to keep it disjoint from the test set.

4.4.3 Network architecture of the CNN

As, input, we take the linearized RGB images in the range $[0,1]$, evaluate a series of n convolutional layers with f filters each, acting on a kernel of size k, with a ReLU as nonlinear activation function in between. The padding in the convolutions is chosen based on k, so as to not change the resolution. The output of all nonlinearities is concatenated and convolved with a 1×1 filter to fuse the information of skipped layers. A last sigmoidal activation function bounds the single channel output r, on which the WHDR-Hinge loss, as given in Sec. 4.4.1, operates during training.

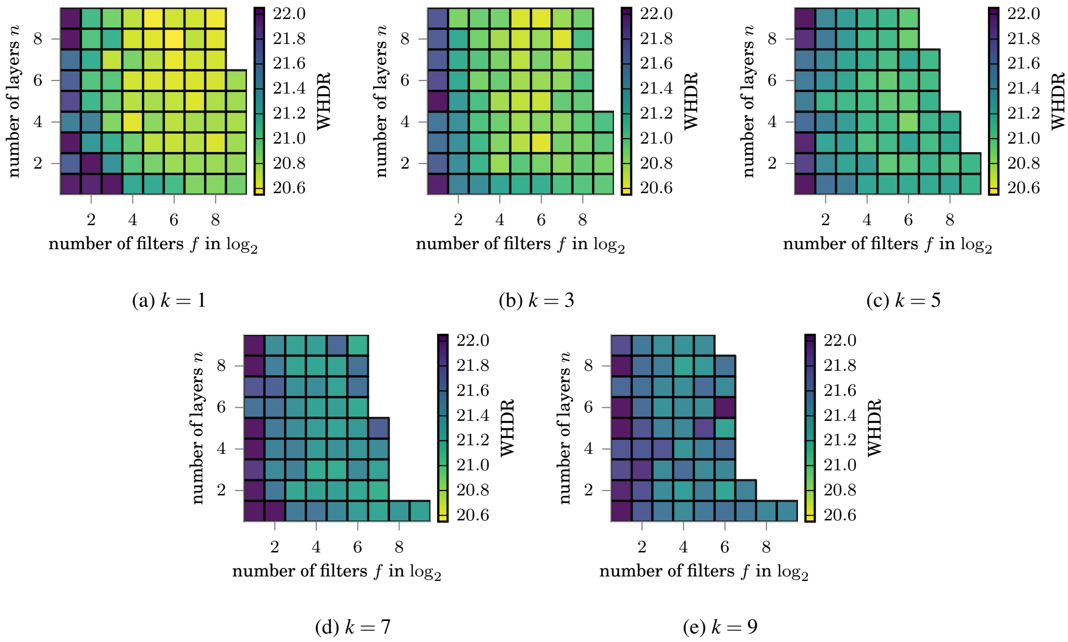
One final layer recovers RGB reflectance R and shading S from the scalar reflectance intensity r, as given in Eq. (4.1), to output the final dense intrinsic image decomposition.

Resolving light intensity. The last nonlinearity in the network acting on r is included since ambiguity about the light intensity in an image cannot be solved. It is only possible to determine reflectance and shading up to a constant $\alpha \in (0,\infty)$, since $I = RS = (\alpha R)\left(\frac{1}{\alpha}S\right)$. Therefore, to keep the reflectance values bounded, we employ a sigmoidal activation function to limit the scalar reflectance intensity to be in the range $[0,1]$.

4.5 Experiments

For all experiments in this chapter, we use the open source deep learning framework caffe [Jia et al., 2014] utilizing the ADAM solver Kingma and Ba [2015] with a learning rate of 0.001, momentum of $\beta_1 = 0.9$ and momentum-2 of $\beta_2 = 0.999$. All training images are resized to a fixed 256×256 pixel resolution to be able to process them in batches.

Data Augmentation. As proposed in Zhou et al. [2015], we tried to augment the comparisons by computing the transitive closure of all comparisons. Instead of pruning the comparisons with low confidence, as done in Zhou et al. [2015], we used all available annotations and set the weight w_i for the augmented comparisons to be the minimum of the confidence of the pair of relations from which it was generated. In case two relations for the same pair of points are generated, we keep the one with higher confidence. In the end we do a consistency check and keep only consistent relations by throwing out the contradicting relation with lower confidence. Despite the much bigger amount of



(a) $k = 1$

(b) $k = 3$

(c) $k = 5$

(d) $k = 7$

(e) $k = 9$

Figure 4.2: **WHDR for different network depths, numbers of filters and kernel sizes in the network.** Mean WHDR on the validation set for different network depths n and number of filters f for the kernel sizes (a) $k = 1$, (b) $k = 3$, (c) $k = 5$, (d) $k = 7$, (e) $k = 9$. Missing data is the result of memory limit on our graphics card.

data (> 20M, a factor of 23.6 times as many comparisons), training on this augmented data did not improve on the resulting WHDR (computed on the original comparisons).

Network Hyper-Parameters. For the network layout described above, we performed an extensive parameter sweep over a varying number of kernel widths $k \in \{1,3,5,7,9\}$, layers $n \in \{1,\ldots,9\}$, and filters $f \in \{2^1,\ldots,2^9\}$. The results on the validation set are shown in Fig. 4.2. The number of layers n has only small influence on the performance above $n \geq 2$, similar with $f \geq 2^4$. While performance does not differ much for for the kernel sizes $k = 1$ and $k = 3$, it seems that bigger kernel sizes overfit more heavily on the training set and therefore have higher mean WHDR on the validation set. An unexpected finding was that 1×1 convolutions work just as well as those with bigger kernels. This means that the network only learns a pixel-wise lookup table, but at the same time, the network performs already better in WHDR than most state-of-the-art methods in the literature. This amounts to a re-scaling of the reflectance intensity at every pixel separately, no context needed. We will discuss this further in Sec. 4.5.1. From this analysis we chose a network of $n = 5$, $f = 2^5$, and $k = 1$ as the basis for all future experiments.

In addition to this basic setup we also played with different network layouts, *e.g.*, without skip connections and with a U-net like architecture [Ronneberger et al., 2015],

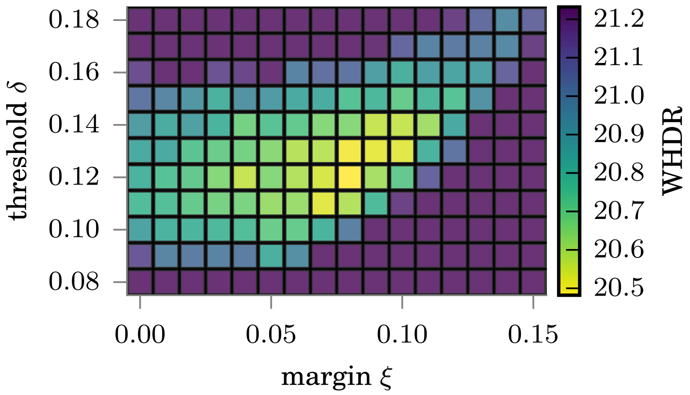


Figure 4.3: **WHDR for different parameters of WHDR-Hinge.** Mean WHDR on the validation set when training with different thresholds δ and margins ξ of the WHDR-Hinge.

tried PReLU [He et al., 2015] nonlinearities in between and dilated convolutions [Yu and Koltun, 2015] to widen the receptive field, but did not find better results. In general we found that simpler networks perform better, what we believe is the outcome of the amount of weakly labeled training data.

Hyperparameters of the WHDR-Hinge loss. To minimize the WHDR rate consistent with $\delta = 0.1$ from Bell et al. [2014] we optimized the loss hyper-parameter δ, ξ on the validation set. The influence is shown in Fig. 4.3 and the final parameters used for training are $\delta = 0.12$ and $\xi = 0.08$.

4.5.1 Discussion of the results

Many methods build on the Retinex assumption [Land and McCann, 1971], which states that strong image gradients are reflectance edges and small gradients are explained by shading. Under the assumption of smooth shading, local gradient estimation would only require a small receptive field, but there is no possibility that a method can resolve shading from a single pixel alone, e.g., see the famous illusion of Adelson [2000] for a counter example. Still, in terms of WHDR, this method performs better than most methods [Shen and Yeo, 2011, Garces et al., 2012, Zhao et al., 2012, Bell et al., 2014, Zhou et al., 2015] on IIW, see the table Table 4.3 for an empirical comparison to a few approaches. A full comparison will be given later, after further improvements in the next chapter, in Fig. 5.2.

To assess the qualitative performance, we compiled a collection of results in Figs. 4.4 and 4.5, an extended comparison can be found in the Appendix in Figs. B.1 to B.3. The images are randomly sampled from the Narihira et al. [2015b] test split. In the spirit of the project page for Bell et al. [2014] we also show grayscale reflectance, especially to highlight the difference between the baseline (const R) and our direct CNN reflectance

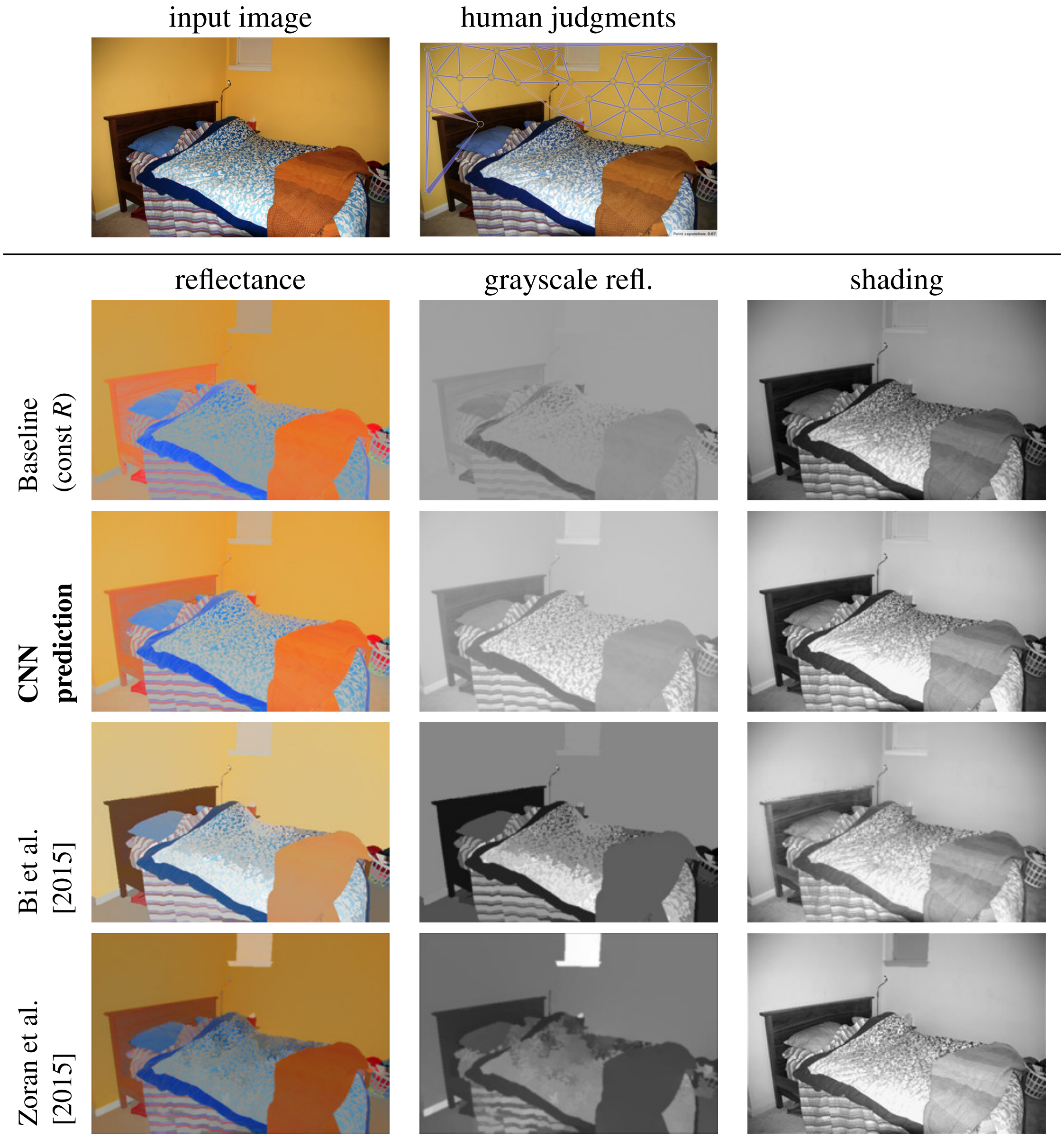


Figure 4.4: **Qualitative comparison on sample image 101684 of IIW.** The first row gives the input image and the evaluated comparisons on it. Comparisons are given as in Bell et al. [2014], where blue is a judgment with high confidence and orange low. The narrow part of the connecting lines is the point which is labeled as darker or they are given as "about the same" when the annotation is a straight line. In the following rows the decompositions into color reflectance in the first column, grayscale reflectance in the second and shading in the third of a subset of methods is shown. All outputs are mapped to sRGB for display.

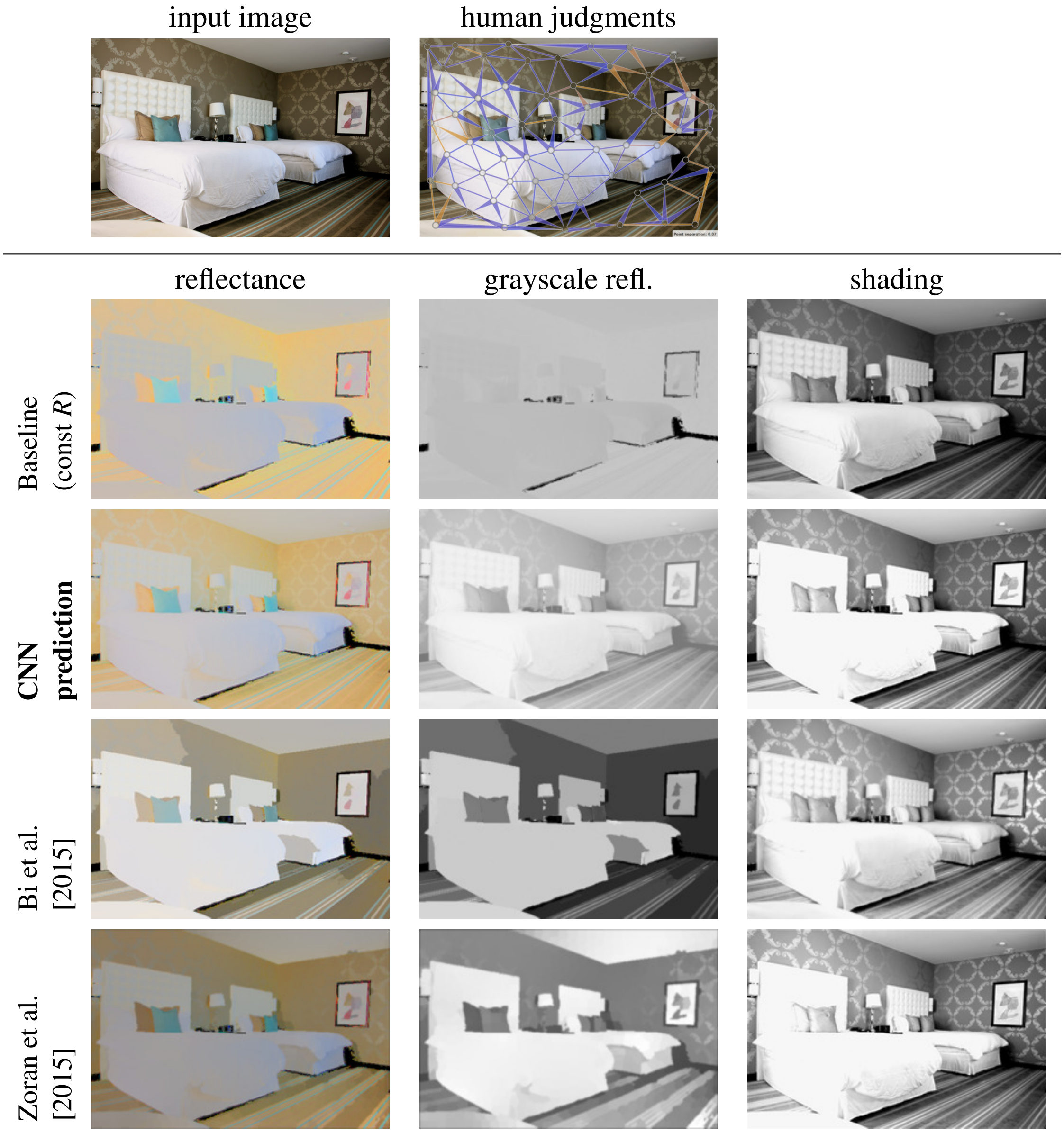
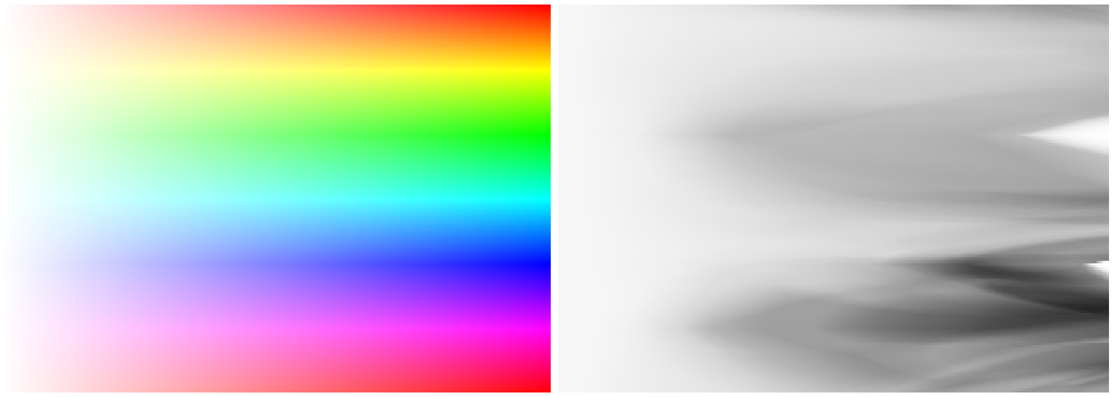


Figure 4.5: **Qualitative comparison on sample image 102147 of IIW.** Extends Fig. 4.4 on IIW ID 102147. All outputs are mapped to sRGB for display.

Table 4.3: **Comparison of intrinsic image approaches on IIW.** We compare the mean WHDR results of relevant intrinsic image approaches on the test set of IIW. An extended comparison will be presented in Fig. 5.2.

Method	Mean WHDR
Retinex [Land and McCann, 1971]	26.9
Bell et al. [2014]	20.6
Zhou et al. [2015]	19.9
Our Direct CNN prediction	19.5
Zoran et al. [2015]	17.9
Bi et al. [2015]	17.7



(a) input image

(b) predicted reflectance

Figure 4.6: **Lookup table for reflectance values generated by the Direct CNN.** Lookup table in HSV space, generated by our direct prediction network, for varying hue and saturation and a constant value/brightness of 255. (a): The input image I, (b): The single channel reflectance intensity r predicted by our Direct CNN.

prediction, which appears to be subtle in the color reflectance, but is not to be overlooked in the grayscale reflectance.

Since there is a direct pixel-to-reflectance relationship, we can visualize a “lookup-table” mapping RGB pixels to reflectance, see Fig. 4.6. A big portion of colors is judged to have more or less the same reflectance intensity as white. Blue is mostly judged being darker than green, which is biologically plausible: Green light contributes the most to the intensity perceived by humans, and blue light the least [Poynton, 2012]. There is a portion of very light reflectance for fully saturated green, even brighter than from white pixels. This might be a result of the Helmholtz-Kohlrausch effect [Corney et al., 2009], according to which humans perceive colored light brighter than white light. This may lead to wrong human reflectance judgments under the circumstances of bright saturated colors.

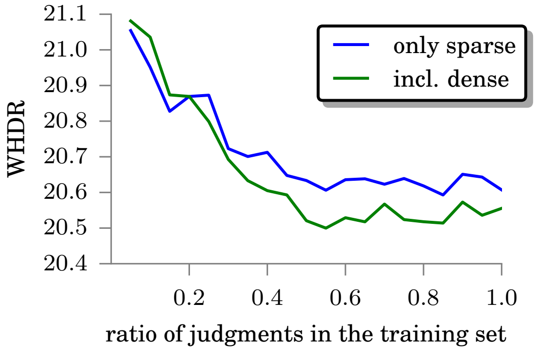


Figure 4.7: **WHDR performance when training with fewer training annotations.** Using a reduced set of human reflectance judgments during the training reduces the mean WHDR validation performance, although it is possible to remove about 50% of the annotation pairs before the WHDR loss starts to decrease. When removing the images with dense judgments from the training set, the performance degrades (see blue line)

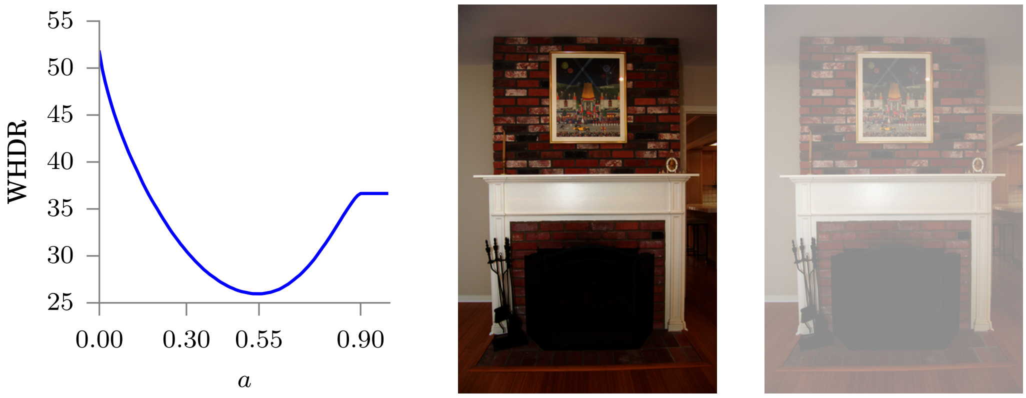
4.5.2 Weak label analysis

We analyzed how much labeled information is needed to obtain good WHDR results. To test this, we reduced the amount of available training data and retrained a fixed network with the parameters $n = 5$, $f = 2^5$, $k = 1$, $\delta = 0.12$, $\xi = 0.08$ from scratch. First we reduce the amount of annotated pairs per image. The result is the green line in Fig. 4.7. We observe that it is possible to remove about 50% of the annotation pairs until the WHDR loss starts to decrease.

Out of the 5230 images in IIW in total, roughly 400 images contain more “dense” annotations. This means they are evaluated at 303 to 1181 pairs (with a median of 916), instead of evaluating 1 to 216 (with a median of 108) comparisons. When removing these images from the training set, the performance degrades (see blue line in Fig. 4.7), as expected.

4.5.3 Rescaling of lower bound

Bell et al. [2014] suggested using the input image directly as a baseline for the reflectance prediction, without any change. Instead, we propose to re-scale the input image from $[0,1]$ into $[a,1]$ for a more elaborate baseline, so that the lower bound of the reflectance prediction has the constant value $a \in [0,1]$. Since WHDR measures reflectance *ratios*, the upper bound can be kept fixed to 1 without loss of generality. On the other hand, linearly scaling the lower bound induces a non-linear change in the reflectance ratios, which influences the WHDR results. For $a = 0$ we have what Bell et al. [2014] named ‘baseline (const S)’, while $a = 1$ is ‘baseline (const R)’. Interestingly, using the parameter $a = 0.55$, which gives the lowest WHDR on the training and validation set as shown in Fig. 4.8, already outperforms Retinex, with a WHDR of 25.7 on the test



(a) (b) (c)

Figure 4.8: **WHDR performance for rescaling approach.** We evaluate the WHDR when rescaling the lower bound of the image and instead of using $[0,1]$, to transform into $[a,1]$ for the reflectance prediction. (a) Mean WHDR (in %) on training and validation set. As reflectance image we take the input image after rescaling it into the range $[a,1]$. (b) An example input image. (c) The same image scaled to have a lower bound of $a = 0.55$, where the mean WHDR is minimal.

set. This low score is due to an in-balance of relative judgments, $2/3$ of which are equal judgments. Reducing the dynamic range and re-scaling to $[0.55,1]$ makes most equal judgments correct and compromises the unequal judgments.

4.6 Conclusion

Until this point, the CNN implements no explicit prior knowledge and it predicts a dense reflectance map on the test set, unaware of the point pairs, where performance will be evaluated on. The output of our CNN is on average in the range $[0.48,0.96]$ and therefore exploits the just described effect of scaling the lower bound. Still, it often leaves small variations in the reflectance image that should be explained via shading gradients, since they fall below the δ threshold for the 'equal' class. It is this fact, that motivates to smooth reflectance values by an additional filtering step to encourage piecewise constant reflectance in the next chapter, where we will see that implementing this prior will lead to generally improved reflectance predictions, not only for our method, but also other state-of-the-art works.

Chapter 5

Reflectance Adaptive Filtering Improves Intrinsic Image Estimation

In the following chapter we concentrate on introducing the prior of piecewise constant reflectance in a novel way through filtering, to improve on our results of the previous chapter and in reflectance predictions in general.

A common prior of intrinsic image estimation is to have only a sparse set of reflectances present in a scene [Omer and Werman, 2004, Gehler et al., 2011, Shen and Yeo, 2011]. We will now describe a new technique based on image filtering that implements this strong prior knowledge about reflectance constancy. This filtering operation can be applied to any intrinsic image algorithm and allows an easy integration into any existing techniques. We found that filtering reflectance estimates improves several previous results achieving a new state-of-the-art on IIW. Our findings suggest that the effect of learning-based approaches may have been over-estimated so far. Explicit prior knowledge is still at least as important to obtain high performance in intrinsic image decompositions.

Contributions This chapter is partially published as [Nestmeyer and Gehler, 2017, Sec. 5 – 6] with some supplemental material and an extended section discussing recent developments of related work since the time of publication.

5.1 Image-aware filtering

A general linear translation-variant filtering process is defined as

$$q_i = \sum_j W_{ij}(I) p_j, \tag{5.1}$$

where the input image p is smoothed under the guidance of an image I to the filtered output image q. Here i, j denote pixels and the sum runs over the entire image. Two examples are the joint bilateral filter and the guided filter, whose weights for Eq. (5.1) we summarize next.

5.1.1 The (joint) bilateral filter

The joint bilateral filter [Petschnigg et al., 2004] is an extension to the bilateral filter [Tomasi and Manduchi, 1998] which uses feature difference in a (potentially different) guidance image to spatially smooth pixels in the input image. It defines the weights as

$$W_{ij}(I) = \frac{1}{K_i} \exp\left(-\frac{|x_i - x_j|^2}{\sigma_s^2} - \frac{|I_i - I_j|^2}{\sigma_r^2} \right), \tag{5.2}$$

with x_i being pixel coordinates. This means that pixels that are both close spatially and in intensity in the guidance image will be smoothed more. The normalization K_i is chosen to ensure $\sum_j W_{ij} = 1$.

5.1.2 The guided filter

The guided filter [He et al., 2010] is a fast alternative to the joint bilateral filter. It is also edge-preserving, and has better behavior near edges. It is based on a locally linear model $\forall i \in \omega_k : q_i = a_k I_i + b_k$, where a_k, b_k are linear coefficients assumed to be constant in the square window ω_k centered at pixel k of size r. The linearity guarantees that q has an edge only if I has an edge, since $\nabla q = a \nabla I$. Solving for the coefficients that minimize the difference between q and p leads to the weights

$$W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \varepsilon} \right) \tag{5.3}$$

where μ_k and σ_k^2 are the mean and variance of I in ω_k, $|\omega_k|$ is the number of pixels in ω_k and ε a constant parameter similar to the range variance σ_r^2 in the bilateral filter. Especially for larger spatial scales, the guided filter benefits from not having the quadratic dependency on the filtering kernel size. We refer to He et al. [2010] for a more thorough discussion.

5.2 Filtering for piecewise constancy

We need to define a guidance image to fully specify the filtering operation. An ideal guidance image would group pixels into regions of constant reflectance. We will refer the filtered image with BF(*method*, *guidance*) for the bilateral filter and GF(*method*, *guidance*) for the guided filter, respectively.

Using a flattened image as guidance. The method of Bi et al. [2015] formulates an optimization problem to group pixels into regions of similar reflectance. This provides a good candidate for a suitable guidance image. The piecewise flattened image is found by minimizing $E = E_l + \alpha E_g + \beta E_a$, with the *local flattening energy*

$$E_l = \sum_i \sum_{j \in N_h(i)} \underbrace{\exp\left(-\frac{\|f_i - f_j\|_2^2}{2\sigma^2}\right)}_{w_{ij}} \|q_i - q_j\|_1, \tag{5.4}$$

where $N_h(i)$ is the $h \times h$ neighborhood of the i-th pixel, q_i is the output RGB vector, $f_i = [\kappa \cdot l_i, a_i, b_i]$, with $[l_i, a_i, b_i]$ being the input vector in CIELab color space and κ, σ are hyper-parameters. A *global sparsity energy* is defined as

$$E_g = \sum_{i \in S_r} \sum_{j \in S_r} w_{ij} \|q_i - q_j\|_1, \tag{5.5}$$

with the same affinity weights w_{ij} as in Eq. (5.4) and S_r being the set of representative pixels which are closest to the average color in their superpixels. To avoid the trivial solution, a *data term for image approximation* is added:

$$E_a = \|q - p\|_2^2. \tag{5.6}$$

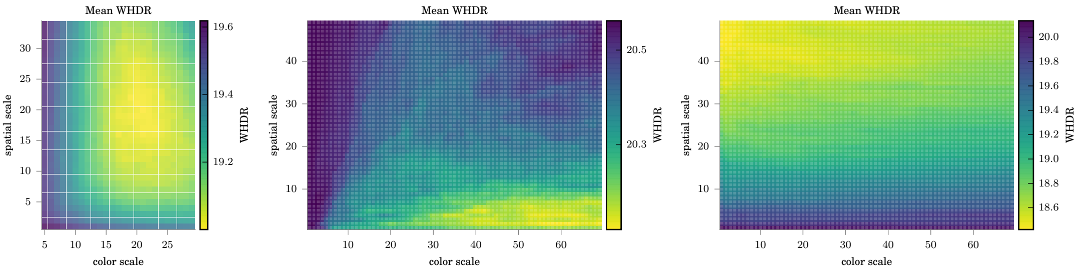
See Bi et al. [2015] for how to solve the resulting optimization problem. We will refer to the result of this L_1-flattening optimization from now on simply as *flat*.

5.3 Experiments

Filtering with the image itself. The conceptually easier choice is to filter using the input image itself as guidance. We applied this to the Direct CNN predictions (referred to as *CNN*) and searched for the spatial- and color scale hyper-parameters of the respective filter on the training and validation set in Fig. 5.1a to find $\sigma_s = 22$, $\sigma_r = 20$ having the lowest mean WHDR. On the test set, this improved the performance from 19.5% to 18.9%.

Guided filtering also improved a bit to 19.2% with $r = 7$ and $\varepsilon = 52$, chosen with the help of Fig. 5.1b.

Filtering using 'flat' as guidance. Using 'flat' as a more elaborate guidance image, we again found the best hyper-parameters on the validation set ($\sigma_r = 15$, $\sigma_s = 28$ for the joint bilateral and $r = 45$, $\varepsilon = 3$ for the guided filter, chosen from Fig. 5.1c). Using the CNN predictions as input,we find the result of BF(CNN, flat) to improve to 18.1% and GF(CNN, flat) further to 17.7%. This is on par with the state-of-the-art (17.67%) at the time of publication of Nestmeyer and Gehler [2017], which is the full pipeline of Bi et al. [2015]: the flat image is clustered, followed by a CRF and another energy minimization step. We note that using the L_1 flattened result directly as reflectance image has only



(a) BF(CNN, CNN)

(b) GF(CNN, CNN)

(c) GF(CNN, flat)

Figure 5.1: **WHDR after filtering with varying color and spatial scale.** Mean WHDR on the training and validation set for filtering (a) BF(CNN, CNN), (b) GF(CNN, CNN), and (c) GF(CNN, flat), with a varying color and spatial scale.

Table 5.1: **Comparison of filtering performance for intrinsic image estimation.** We compare the performance when filtering the intrinsic image estimation methods under varying guidance images. We report the improvement in mean WHDR (in %) over the images in the test split from Narihira et al. [2015b], and for Zoran et al. [2015] results on their respective test set (marked with an asterisk) before and relatively to it after one filtering operation.

Method	F(CNN, CNN)	F(CNN, flat)	F(Zoran et al. [2015], flat)*
unfiltered	19.49	19.49	17.85
bilateral filter (BF)	-0.6	-1.38	-1.47
guided filter (GF)	-0.25	-1.8	-1.98

20.9% WHDR, which shows that there is complementary information in the CNN output and the guidance image.

This use of the flattened image as a guidance in a filter, extends Bi et al. [2015] and allows application to other intrinsic image decompositions. We apply filtering to the second best method Zoran et al. [2015] on IIW. Their work proposes to create a sparse representation of the image by using the centers of a superpixelization. Patches around those centers are extracted and a CNN is used to provide an ordinal relationship via the three-way classification into “darker”, “equal”, and “lighter”. This sparse result is then again densified by solving a constrained quadratic optimization problem to produce a full reflectance image.

The filtering step is mostly dependent on the feature space, therefore we used the same filtering hyper-parameters from above, when smoothing the method of Zoran et al. [2015], since we only had access to their test set and hence could not optimize for the best parameters. Nonetheless, the application of GF(Zoran et al. [2015], flat) using these parameters improves WHDR from 17.85% to 16.38%. Repeated application of the filter further improves the output down to 15.78% after three applications of the guided filter. This result represented the new state-of-the-art by a large margin at the time of publication.

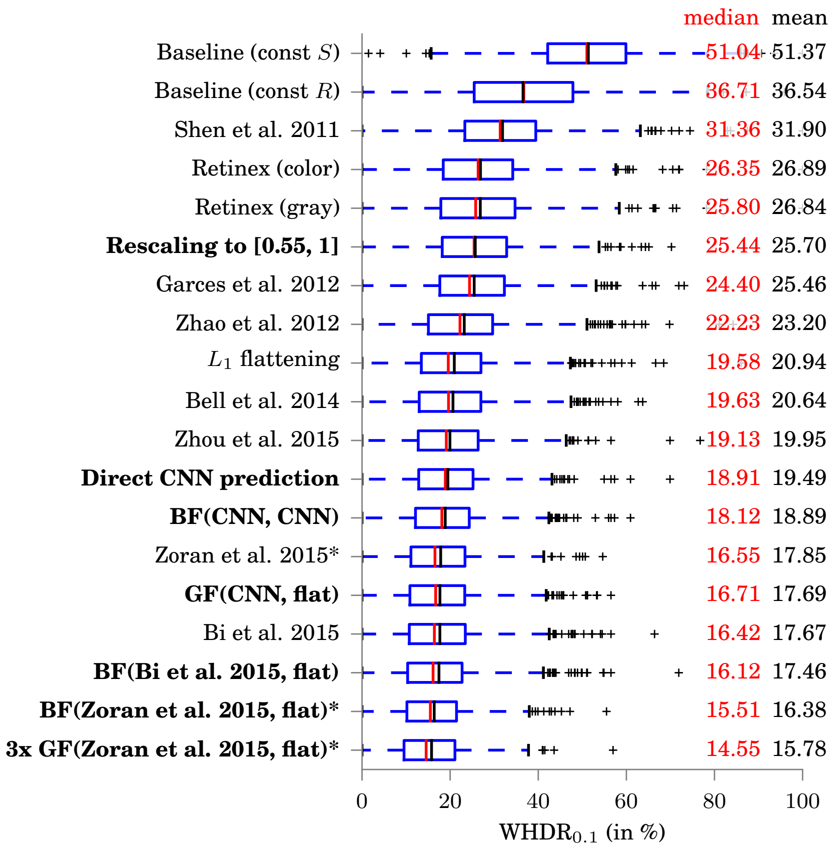
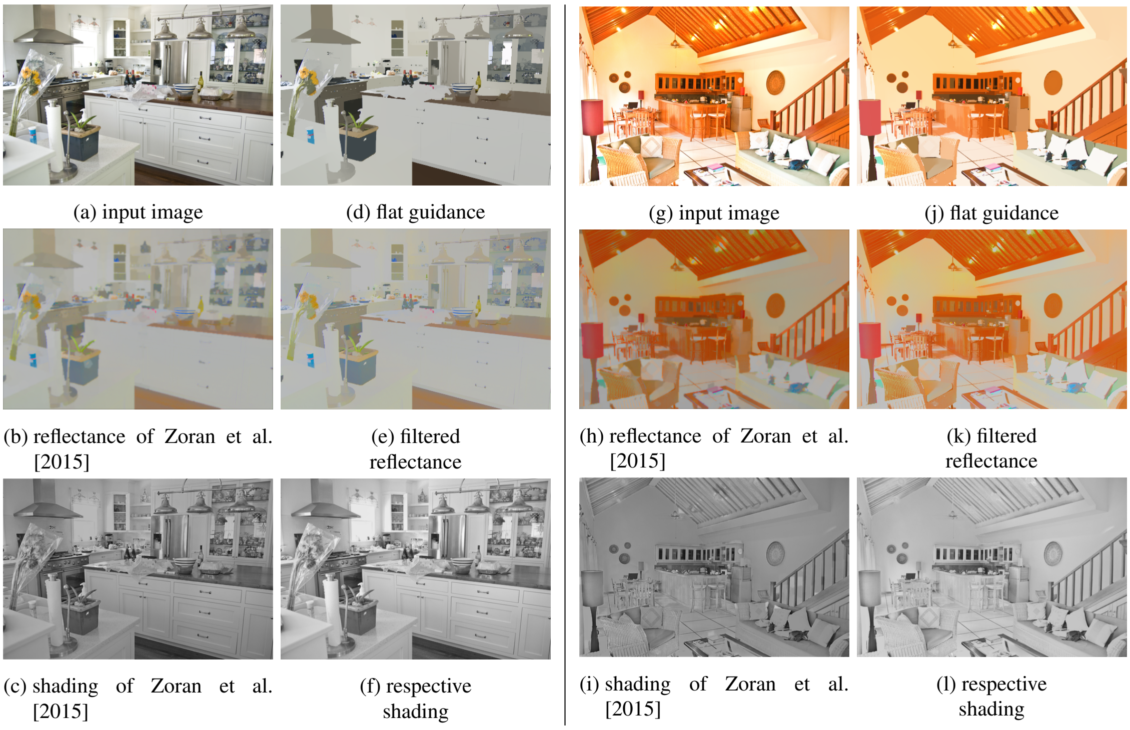


Figure 5.2: **Comparison of performance to related intrinsic image estimation methods.** We compare the WHDR performance on IIW to related intrinsic image estimation methods. Over all images in the test split from Narihira et al. [2015b], we report the statistics of the individual WHDRs on the images. The red line represents the median, the black line the mean. Results of Zoran et al. [2015], are based on reflectance predictions provided by the authors which are generated on a different test split. All methods that are evaluated on this different test set are marked with an asterisk as they are not directly comparable.

5.3.1 Discussion of the results

Quantitative results. We found throughout that guided filtering of reflectances improves performance, see Table 5.1 for a collection of the results. There are some cases where the joint bilateral filter outperforms the guided filter, but in general the latter leads to better performance. Also the guided filter is magnitudes faster. We summarize a comparison with recent methods and state-of-the-art in Fig. 5.2. As mentioned in Sec. 4.5.1, our direct CNN approach from Chapter 4 outperforms all but two state-of-the-art methods, and combined with the additional guided filter leaves one more contender behind. Utilizing our novel filtering technique to improve reflectance predictions of related work on the other hand improves state-of-the-art by a large margin, which shows its generality.



(a) input image

(d) flat guidance

(g) input image

(j) flat guidance

(b) reflectance of Zoran et al. [2015]

(e) filtered reflectance

(h) reflectance of Zoran et al. [2015]

(k) filtered reflectance

(c) shading of Zoran et al. [2015]

(f) respective shading

(i) shading of Zoran et al. [2015]

(l) respective shading

Figure 5.3: **Sample decompositions visualizing the components of the reflectance adaptive filtering approach.** Sample decompositions of the (a) input image with the IIW ID 71341 into (b) reflectance and (c) shading by the method of Zoran et al. [2015]. Filtering this reflectance prediction using the flat guidance image in (d) results in the intrinsic layers (e) and (f) of our final model. (g)-(l) are the same as (a)-(f) for IIW ID 58346.

Qualitative results. The method of Zoran et al. [2015] has staircase effects due to the superpixelization in reflectance. This is removed by our reflectance filtering step, when filtering with the flat image, which not only leads to increased quantitative, but, as expected, also qualitative results. For an impression to assess the qualitative performance, we refer to Fig. 5.3. Results on a larger number of sample images and in comparison to more related work, as well as our Dircet CNN prediction with and without filtering, can be found in Appendix B.2.

5.3.2 Runtime analysis

In Fig. 5.4 we show the runtime of different algorithms against their WHDR. All methods of this and the previous chapter are colored in green. We collected the timing estimates from the respective statements in the corresponding publications. By construction, our direct prediction CNN with only a few filters is fast at test time (180 fps on GPU) but it requires further filtering for better results. The bilateral filter adds around 2 s per image on CPU and the guided filter less than 0.1 s. The bottleneck of the filtering approach is the computation of the L_1 flattened guidance image.

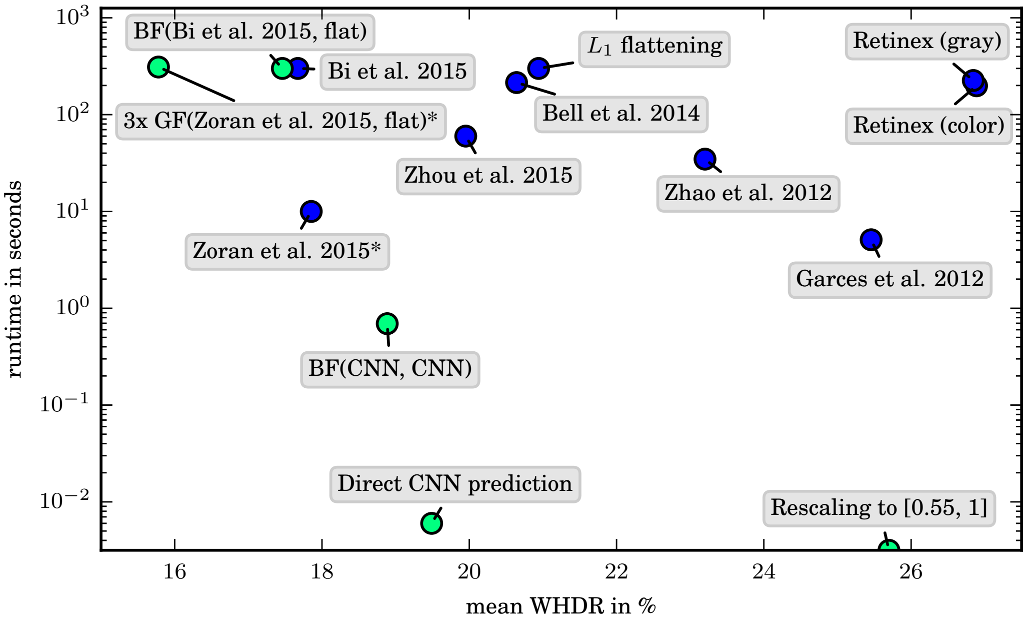


Figure 5.4: **WHDR against runtime for related work.** Mean WHDR of competing methods is evaluated on the Narihira test split [Narihira et al., 2015b] on the decompositions provided by Bell et al. [2014] and the project pages of Bi et al. [2015], Zhou et al. [2015]. Methods with an appended asterisk were evaluated on the test split given in Zoran et al. [2015]. For methods which are evaluated in Bell et al. [2014], we used the reported runtimes on the corresponding project page. Methods developed in this work are plotted in green, previous results are plotted in blue.

5.3.3 Quantitative comparison to related work after publication

The field of intrinsic image decomposition advanced further since the publication of Nestmeyer and Gehler [2017] and we will summarize the main advancements now.

Kovacs et al. [2017] introduced the Shading Annotations in the Wild (SAW) dataset, extending the reflectance judgments in IIW with three categories of annotations: 1. regions of near-constant shading, 2. edges due to discontinuities in shape, and 3. edges due to discontinuities in illumination. This gives an additional valuable measure to evaluate intrinsic image decompositions. Li and Snavely [2018] took the annotations from SAW and introduced a gradient-weighted “challenging” benchmark in which a pixel is weighted less if it is located in an easy region where the input image intensity and the shading intensity are both smooth. Therefore, we will present new methods on both these metrics, WHDR in IIW and average precision (AP) in SAW (if quantitative results were published in any recent work) for the challenging benchmark and collect their results in Table 5.2.

In the same work, Li and Snavely [2018] also created a new large-scale dataset called CGIntrinsics of physically-based rendered images of scenes with full ground truth decompositions to train a U-Net with 2 decoders predicting reflectance and shading in log-domain, that set the new state-of-the-art on SAW. They also applied a filter for post-processing to improve their performance on reflectance estimation, but this leads to a slight deterioration on shading estimation.

In parallel, Fan et al. [2018] heavily built on our work. In a similar fashion, they use a direct CNN prediction for reflectance, where core network structures reflect prior knowledge about the image formation process, but instead of filtering as a post-processing step, they integrate a domain filter guided by a learned edge map into their unified deep architecture for end-to-end learning.

In GLoSH, Zhou et al. [2019b] predict reflectance and surface normals, which are shaded through a spherical harmonics light model with a global and local component that is constrained to non-negative lighting. They also realized that our loss in Eq. (4.4) is not symmetric, therefore added a symmetry term as one of their ingredients and improved their final WHDR by 0.14% in doing so.

Most recently, Luo et al. [2020] created a unified framework called NIID-Net, that jointly estimates normals, reflectance and shading, which performs reasonably well on reflectance estimation and sets the state-of-the-art in shading estimation.

An interesting alternative approach is given by the Fast Fourier Intrinsic Network (FFI-Net) [Qian et al., 2021] which operates in the spectral domain, splitting the input into several spectral bands. Its weights are directly optimized in the spectral domain based on a spectral loss which measures global distance between network prediction and ground truth and they implement multi-scale learning by spectral banding.

The results of those methods along with our results are summarized in Table 5.2. While the state-of-the-art in reflectance prediction in the year 2021 is reduced to a WHDR under 15%, the influence of our work [Nestmeyer and Gehler, 2017] becomes all the more apparent: Our contributions from the last Chapter 4 heavily inspired Fan et al. [2018], setting the new state-of-the-art and the simple to apply filtering as post-processing from this Chapter 5 boosts the performance of Li and Snavely [2018] to be the follow-up.

5.4 Conclusion

In the last two chapters, we have proposed methods that are on opposing ends of employed prior knowledge. This led to both the best results on IIW and valuable insights into the current state of intrinsic image estimation at the time of publication [Nestmeyer and Gehler, 2017]. We presented the first end-to-end CNN method, trained on the WHDR-Hinge loss, that predicts a dense result without any post-processing step. Our finding is that a context-free per-pixel judgment is sufficient for competitive results. We believe that this should set a new lower bar for learning methods on IIW. While this observation may be attributed to an inherent bias in IIW, we have no qualified reason to believe so. We still conjecture that good results correlate with low WHDR numbers, and note that human performance sets a high bar with a median WHDR of only 7.5% [Bell

Table 5.2: **Extended quantitative results.** We extend the quantitative results in Fig. 5.2 with results after publication of Nestmeyer and Gehler [2017] (if results were published in any recent work). 'WHDR' is given as the mean $\text{WHDR}_{0.1}$ in % on IIW. Average precision (AP) on SAW [Kovacs et al., 2017] is given in % under the gradient-weighted challenging benchmark introduced in Li and Snavely [2018]. Our contributions are given in bold, but our published best result as given in Fig. 5.2 (*3x GF(Zoran et al. [2015], flat)*) is evaluated on a different test set and therefore marked with an asterisk because it is not directly comparable. The filtered result *GF(Li and Snavely [2018], flat)* was also provided in Li and Snavely [2018] and improves their unfiltered CGIntrinsics result by 0.7 percentage points. The best results on WHDR and AP are marked in bold each.

Method	WHDR [%]	AP [%]
GF(CNN, flat) from Chapters 4 and 5	17.7	88.64
NIID [Luo et al., 2020]	16.6	**98.40**
3x GF(Zoran et al. [2015], flat)*	15.8	-
FFI [Qian et al., 2021]	15.8	-
CGIntrinsics [Li and Snavely, 2018]	15.5	97.93
GLoSH [Zhou et al., 2019b]	15.2	95.01
GF(Li and Snavely [2018], flat)	14.8	96.57
Fan et al. [2018]	**14.5**	-

et al., 2014]. This has not been attained by any automatic method so far. We further develop a filtering technique to implement the assumption of piecewise constant and sparse reflectance. This extends the work of Bi et al. [2015] and makes it possible to apply their reflectance grouping to other decompositions. We find that, in 2017, the filtered CNN output was on par with the best published learning based methods and further improved the initial result of Zoran et al. [2015] to 15.78% on its testset, which was the lowest WHDR performance for a dense decomposition.

Time has passed since the publication of our work [Nestmeyer and Gehler, 2017], and the leaderboard undoubtedly progressed. Nonetheless, we provide the main ingredients for the two best performing methods of Li and Snavely [2018] who use our filtering to boost their performance in post-processing and Fan et al. [2018] who generally took big inspiration from our work.

In summary, the findings of the last two chapters suggest that it is still the use of strong prior knowledge in intrinsic estimation algorithms that drives empirical performance.

All code, models, and results are available at `https://ps.is.tue.mpg.de/research_projects/reflectance-filtering`.

As we have seen in Chapter 2, robotics is only one field, where intrinsic images play a relevant role in an intelligent system. Another big topic of current research is Augmented Reality (AR), and we will look into how intrinsic images can be used for relighting in AR in the following. In order to posses the necessary data available for training, we will examine in the next chapter, how capturing it in large scale can be accomplished.

Chapter 6

Large scale acquisition of intrinsic layers in a light stage

In Chapter 4 we already discussed the difficulty of getting large scale ground truth for intrinsic images and how the community tries to tackle this by only annotating sparse data or using synthetic data to avoid cumbersome manual labeling with the drawback of reduced realism. Because we realized in Chapters 4 and 5 that the sparse labels from IIW are insufficient for meaningful supervised learning alone, without additional priors, we will now describe how we acquire high quality reconstructions of a much broader set of intrinsic layers than only reflectance and shading for a diverse set of facial expressions captured under various lighting conditions. Namely, these layers consist of albedo, normals, shading, depth, a light visibility mask, a non-diffuse residual and recovered light calibration. We will use this data to advance research on relighting in the next Chapter 7.

This resonates well with the age-old theoretical decomposition of Barrow and Tenenbaum [1978] into consituent layers such as surface reflectance (a combination of diffuse albedo and the non-diffuse residual), surface orientation (normals), distance (depth) and incident illumination (light calibration and light visibility).

Contributions This chapter is an extended version of [Nestmeyer et al., 2020, Sec. 4], providing much more details on building and calibration of the light stage, as well as the capturing process with subsequent intrinsic layer separation.

6.1 Building the light stage

We record our data in a calibrated multi-view light-stage (see Fig. 6.1) consisting of 6 stationary Sony PMW-F55 camcorders and a total of 32 white LED lights. The cameras record linear HDR images at 2048×1080 / 60 fps and are synchronized with a Blackmagic sync generator that also serves as a signal wich is read by an Arduino that triggers the LED lights in sync with the recorded frames. The LEDs are mounted on four different horizontal bars, each offset with a 3D printed extension in varying lengths in order to achieve a good coverage of light directions on the frontal half dome around the face.

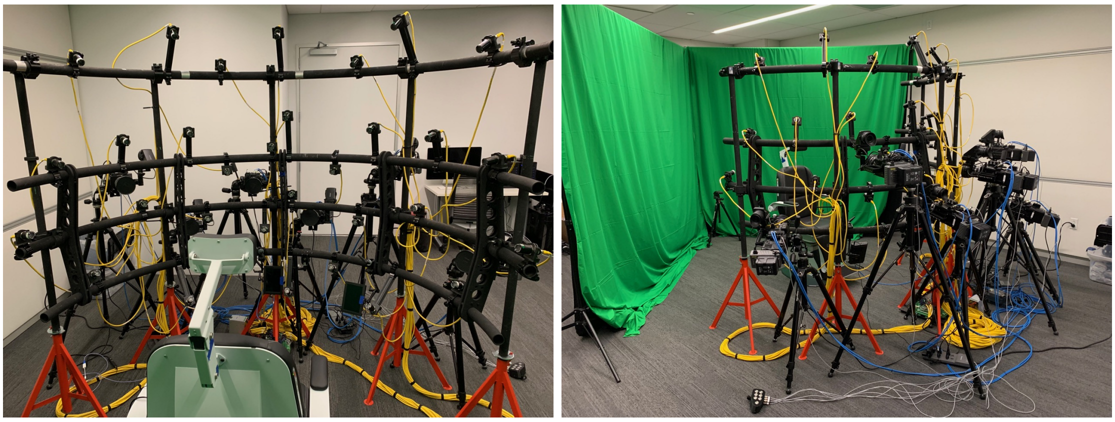


Figure 6.1: **Light stage capturing setup.** The capturing setup in the form of a light stage consists of 6 synchronized Sony PMW-F55 camcorders and 32 white LED lights synchronized to the cameras. The chair with a head rest lets the subjects sit still during capturing a sequence (~ 0.53 s).

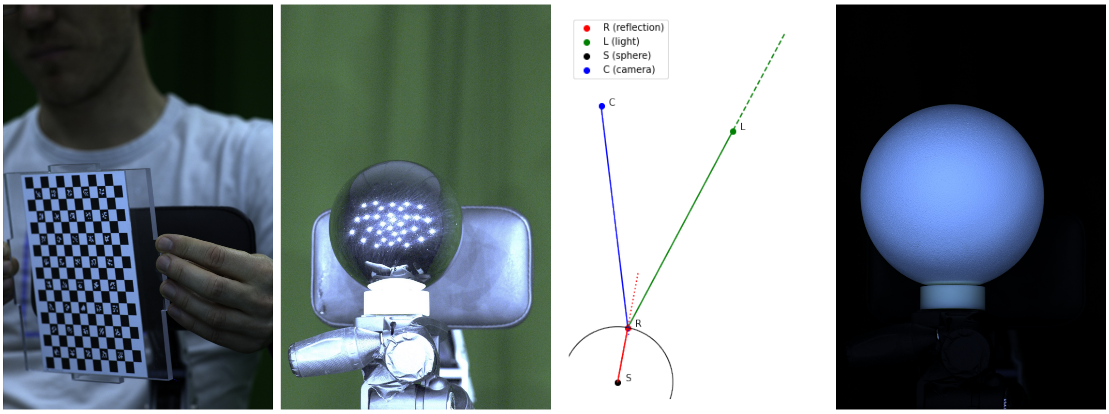
In order to not alter the extrinsic calibration of the cameras when touching the recording button, we use a remote that lets all cameras to be triggered from one device. The subjects take a seat on a chair with a head rest to facilitate sitting still when their had is leaned against the head rest and to allow using a fixed focus on the cameras.

6.2 Calibrating the light stage

In order to make the recorded data from this capturing setup useful, we now describe the multi-step procedure taken to calibrate the whole light stage.

For intrinsic and extrinsic calibration of the cameras (see Fig. 6.2a), we use a ChArUco board [An et al., 2018] printed on an acrylic glass plate. ChArUco combines a regular checkerboard pattern with an ArUco pattern [Garrido-Jurado et al., 2014]. ArUco markers provide fast detection and versatility, because the board does not need to be completely visible and occlusions are permitted. On the other hand, corners of a checkerboard pattern can be refined more accurately since each corner is surrounded by two black squares, so ChArUco combines the advantages of both those patterns.

For light calibration [Goldman et al., 2010], we use a chrome sphere (see Fig. 6.2b), positioned where later also our subjects will have their head positioned, to recover directions and intensities in 3D: We approximate the projection of the sphere into the image by a circle in each of the 6 camera images. Using the camera calibration, we project the center of these circles in each camera back into the world in order to recover a 3D ray each, along which the sphere center has to lie. Finding the point of closest intersection of those 6 rays then gives us an accurate reconstruction of the sphere center as a 3D point. With the measured diameter of the sphere, we continue to find the points of reflection of the individual lights by analytically intersecting the projected ray with the recovered sphere. Consecutively, we reflect the ray on the chrome sphere [Eberly, 2008] in order to



(a) In- and extrinsic calibration (b) The chrome sphere (c) Finding the light location (d) Lambertian sphere

Figure 6.2: **Calibrating the light stage.** We show the steps taken in order to achieve a calibrated light stage. (**a**) Intrinsic and extrinsic calibration of all the 6 cameras using a ChArUco board. (**b**) Identifying the points of reflection of the 32 lights in a chrome sphere. (**c**) Solving for the light direction by finding the ray of reflection, here visualized for one camera and light in 2D. (**d**) Lambertian sphere used for light intensity estimation.

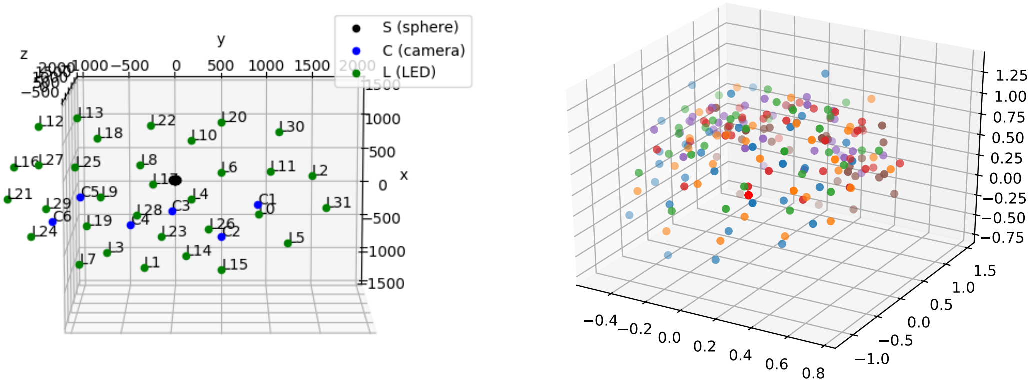
reconstruct the direction of the LEDs with respect to the sphere (see Fig. 6.2c). We then record a white Lambertian sphere (see Fig. 6.2d) to calibrate for the light intensities of the LEDs.

In Fig. 6.3 we show the result of this calibration process. The recoverd 3D locations of the sphere, the cameras and light are given in Fig. 6.3a. Subsequently, we express the light configuration with respect to each of the 6 camera planes, such that we obtain a total of $6 \cdot 32 = 192$ different light directions/intensities for each image (see Fig. 6.3b).

6.3 Recording data

For recording the data, we flash one LED per frame (see Fig. 6.4) and instruct our subjects to hold a static expression for the full duration of an LED cycle (32 frames, amounting to $\sim 0.53\,\text{s}$). We let the subjects trigger the recordings themselves through a remote, connected to the Arduino to trigger the light sequence. Meanwhile leaning their heads against a head rest makes it easier for the subject to hold a pose during the given duration. The order in which the LEDs are flashed is defined in a random but fixed pattern in order to not let the subjects follow the path of the lights with their eyes, or even worse, their pose, to further limit motion during the capture. In order to remove captures with motion, we filter our data based on the difference of two fully lit shots before and after each cycle.

We record a total of 482 sequences from 21 subjects (see Fig. 6.5), on average around 23 sequences per subject, resulting in $482 \cdot 6 \cdot 32 \cdot 32 = 2{,}961{,}408$ relighting pairs. Each pair is formed using any one of the 32 lights as input, and any one taken from the same



(a) 3D reconstruction of light stage

(b) Reconstructed lights in image space

Figure 6.3: **Result of light stage calibration.** We show the calibration result of the light stage. (**a**) Full 3D reconstruction of cameras in blue (from extrinsic calibration), the chrome sphere in black (from projecting circles in images), as to where subjects will have their heads and the location of lights in green (from reflecting the lights on the chrome sphere) when assuming a fixed distance from the sphere. This nicely aligns with Fig. 6.1 and Fig. 6.2b. (**b**) Distribution of lights after interpreting them relative to the camera frame, represented with one color per camera. The distance from the origin represents the recovered intensity.

Table 6.1: **Dataset details.** Distribution of salient characteristics in training, validation and test set.

	female	dark skin	glasses
training	2	3	4
validation	1	1	1
test	1	1	1
total	4	5	6
ratio	19.0%	23.8%	28.6%

sequence and same camera as output. We manually split them into 81% (17 subjects) for training, 9.5% (2 subjects) for validation and 9.5% (2 subjects) for testing according to Table 6.1, where we tried to achieve a meaningful distribution of salient characteristics in all of the subsets.

We did not ask the subjects to follow any specific protocol of facial expressions, besides being diverse, such that our evaluation on validation and test data is on *both* unseen expressions and unseen subjects. For some of the captures, we specifically asked the participants to cover their face partially with their hand in order to achieve challenging data which cannot be relit by any template-based model. Subjects that wear glasses were asked to record some sequences with and some without their glasses.

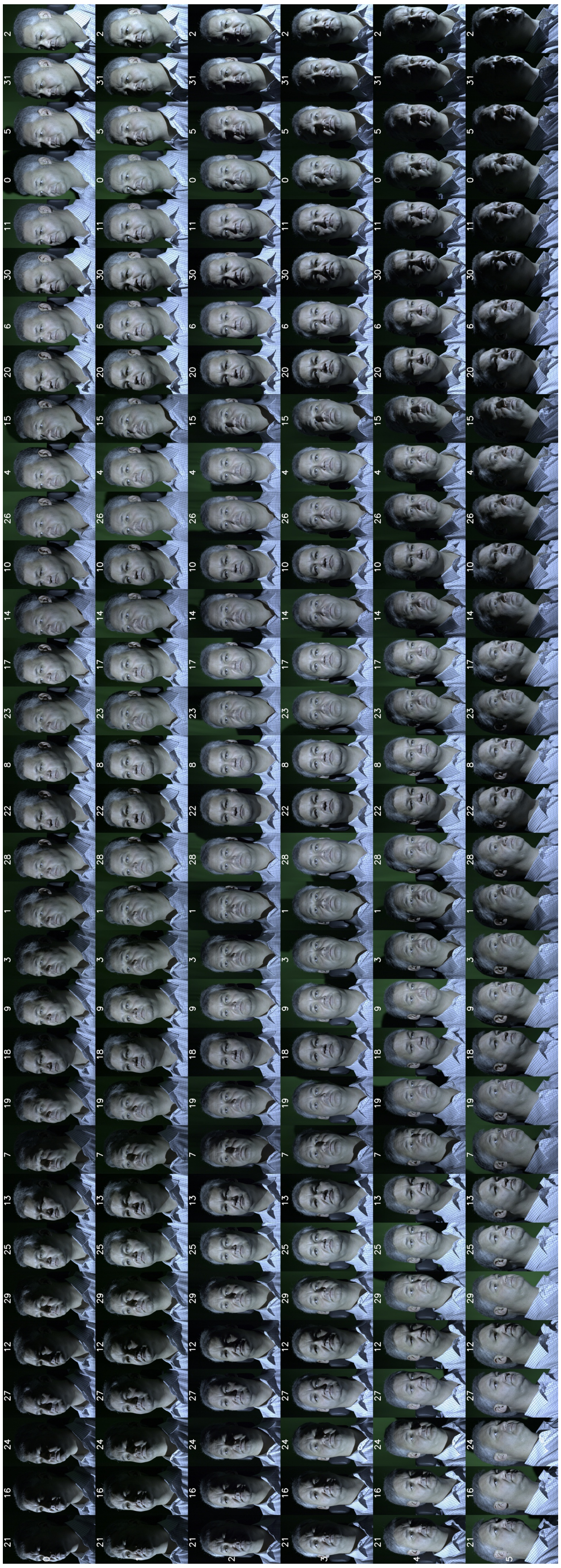


Figure 6.4: **Full light sequence capture from all cameras.** We show all directionally lit captures of one sequence as seen through each of the 6 cameras in the rows while lit under one of the 32 lights in the columns. Cameras and lights are both sorted by their azimuth (independent of their elevation) for improved visualization, even though they were captured in a random (but fixed) order to discourage subjects to follow the light sequence. The number on top of the head represents the index of the light in its flashing order.

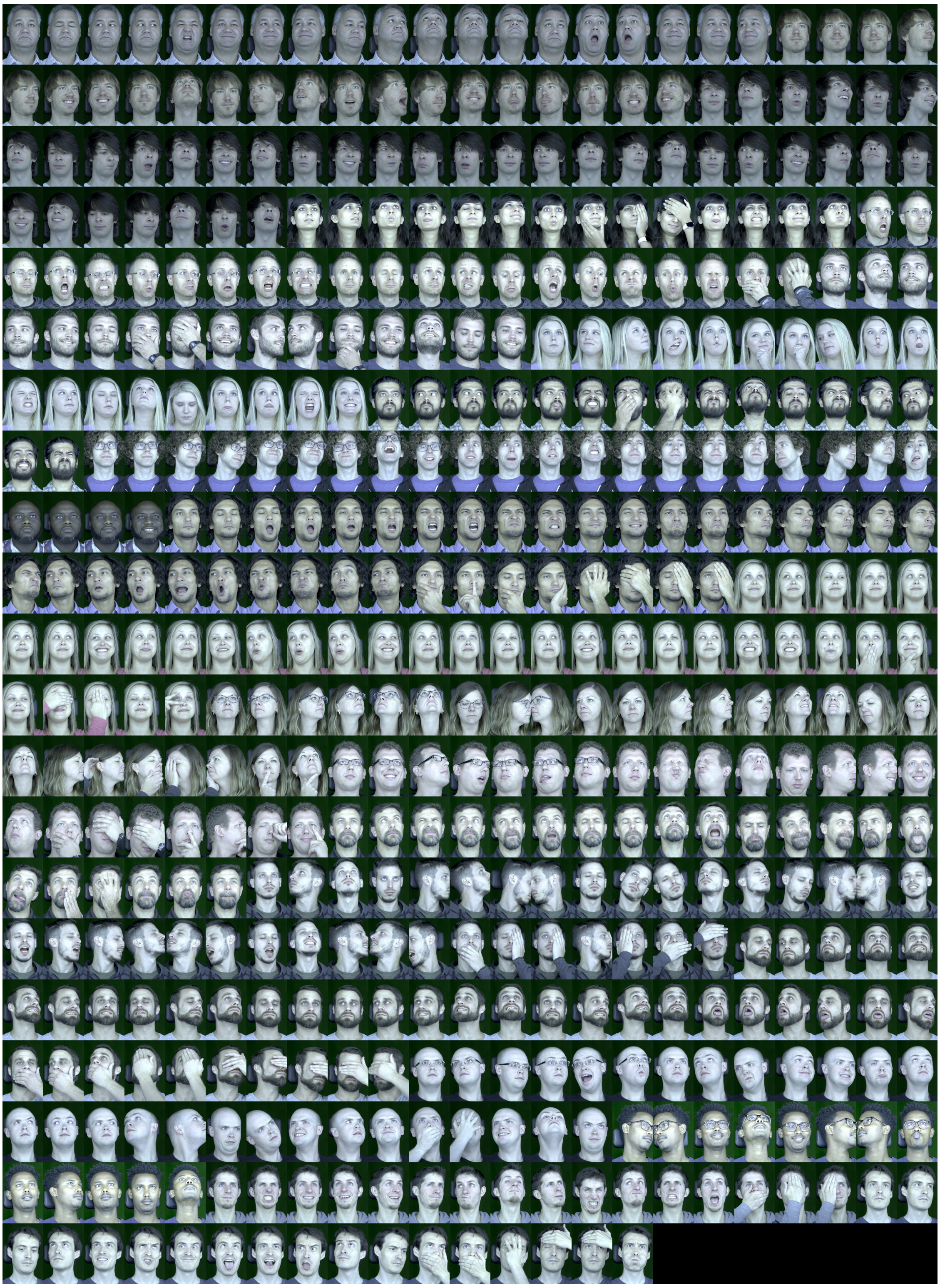


Figure 6.5: **All captured subjects in poses.** Diversity in the data was achieved by capturing a breadth of 21 subjects and letting them independently choose arbitrary expressions with only little guidance to, e.g., cover the face with a hand, to capture interesting scenes. Subjects with glasses were captured for part of the expressions with and partially without wearing their glasses.

6.4 Processing the intrinsic layers

After extraction of the raw data from the video streams, we use photometric stereo (PMS) reconstruction [Xiong et al., 2014] to separate the captures into one common albedo A and corresponding normals for the whole light sequence, while for each individual input image I, we recover shading S, and a non-diffuse residual image $R = I - A \odot S$. While at the same time refining the light calibration numerically, in total this leads to the inferred intrinsic layers albedo, normals, depth, shading, a non-diffuse residual, and a light visibility mask. We visualize those intrinsic layers in Fig. 6.6 for three exemplary lights.

6.5 Conclusion

Getting ground truth intrinsic images is no easy endeavor. Nevertheless, to obtain the required intrinsic layers for our neural network solving relighting in the next chapter, we were in need for a decomposition into more than simple reflectance and shading from the previous chapters. In this chapter, we saw that our light stage setup provides the expected controlled environment, that combined with a relentless eye for the detail in the calibration, provides high accuracy captures with according reconstructions. Successively running photometric stereo proved satisfactory in achieving the expected layers of albedo (the diffuse portion of reflectance), high frequency normals, cast shadows as visibility mask of the light source on the subject, and a non-diffuse residual term. These intrinsic layers will be of paramount importance in the next chapter, where we will focus on how the task of relighting can utilize this separation into intrinsic layers in its structured neural rendering model.

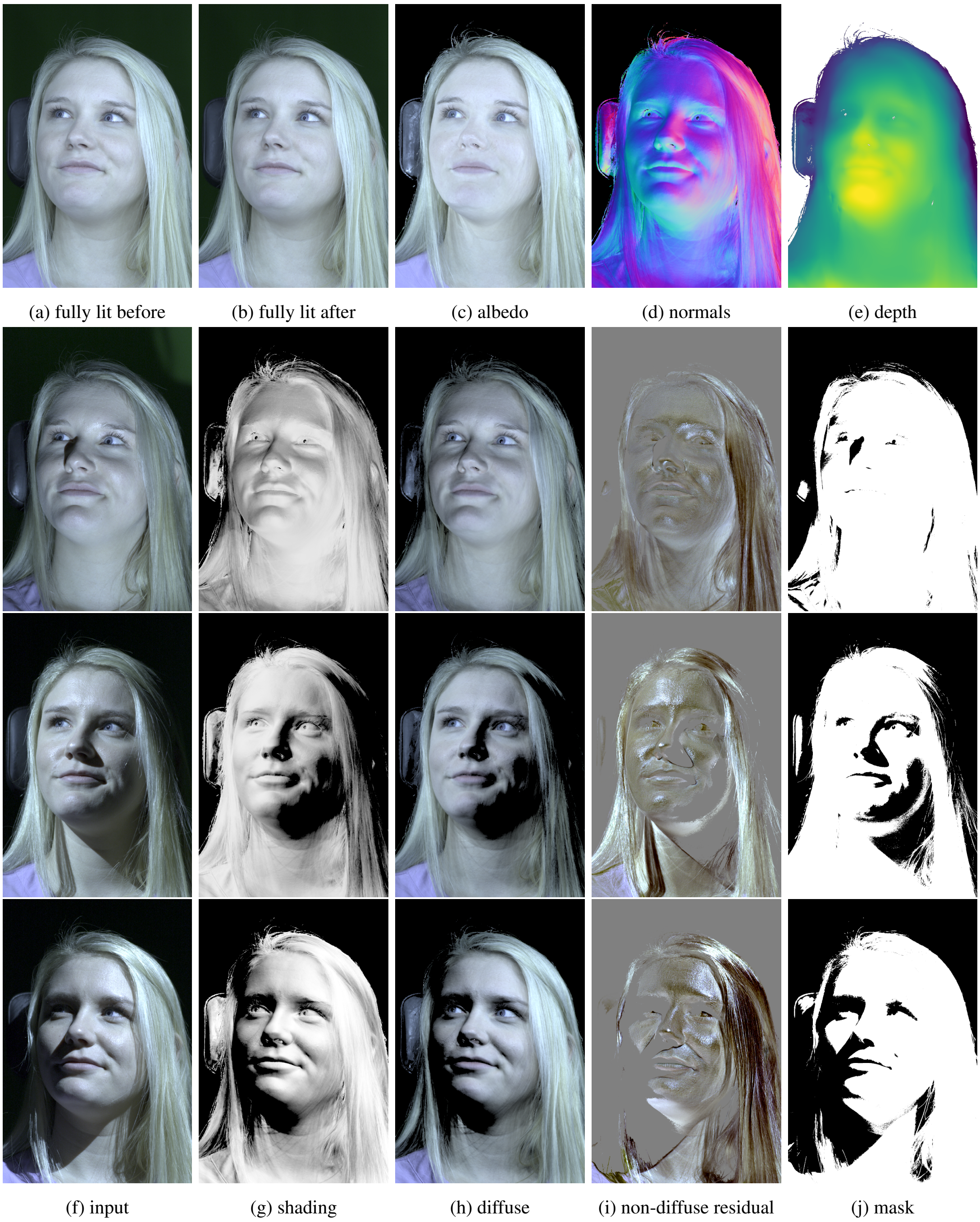


Figure 6.6: **Samples from the intrinsic layers.** Each recording sequence (**a**) starts and (**b**) ends with a fully lit capture (with all 32 lights on) to determine how much a subject moved during the capture. In between, each LED is flashed individually. With photometric stereo [Xiong et al., 2014] we then reconstruct common (**c**) albedo A, (**d**) normals, and (**e**) depth, each independent of the 32 individual lights. Then, exemplarily for three of the individual lights, the (**f**) input capture I, and the reconstructions of (**g**) shading S, (**h**) diffuse reconstruction $A \odot S$, (**i**) non-diffuse residual $R = I - A \odot S$ (scaled in intensity for better visibility), and (**j**) inferred light mask used for reconstruction.

Chapter 7

Learning Physics-guided Face Relighting under Directional Light

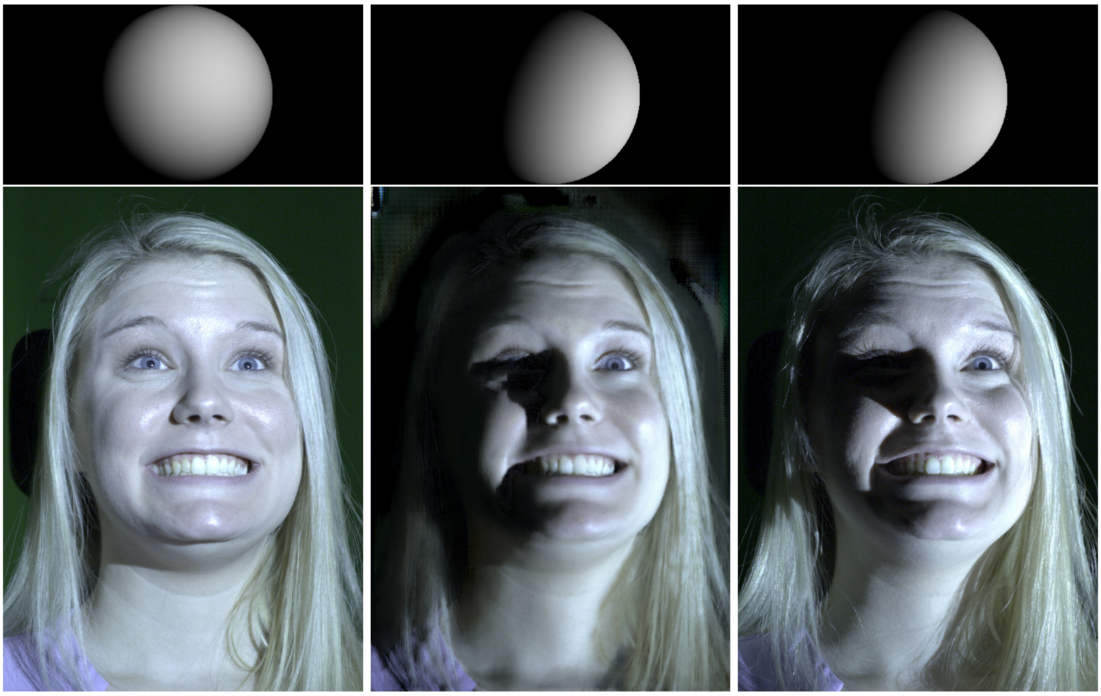
Relighting is an essential step in realistically transferring objects from a captured image into another environment. For example, authentic telepresence in Augmented Reality requires faces to be displayed and relit consistent with the observer's scene lighting. We investigate end-to-end deep learning architectures that both *de-light* and *relight* an image of a human face. Our model decomposes the input image into intrinsic components according to a diffuse physics-based image formation model. We enable non-diffuse effects including cast shadows and specular highlights by predicting a residual correction to the diffuse render. To train and evaluate our model, we take our portrait database of 21 subjects with various expressions and poses from Chapter 6, where each sample is captured in a controlled light stage setup with 32 individual light sources. Our method creates precise and believable relighting results (see Fig. 7.1 for a teaser) and generalizes to complex illumination conditions and challenging poses, including when the subject is not looking straight at the camera.

Contributions This chapter was presented as an oral in CVPR 2020 [Nestmeyer et al., 2020].

7.1 Introduction

In recent years Augmented Reality (AR) has seen widespread interest across a variety of fields, including gaming, communication, and remote work. For an AR experience to be immersive, the virtual objects inserted in the environment should match the lighting conditions of their observed surroundings, even though they were originally captured under different lighting. This task, known as *relighting*, has a long history in computer vision with many seminal works paving the way for modern AR technologies [Land and McCann, 1971, Barrow and Tenenbaum, 1978, Peers et al., 2007, Barron and Malik, 2015, Sengupta et al., 2018].

Relighting is often represented as a physics-based, two-stage process. First, *de-light* the object in order to recover its intrinsic properties of reflectance, geometry, and light-



(a) ℓ_{src}, input (b) ℓ_{dst}, prediction (c) ℓ_{dst}, ground truth

Figure 7.1: **Overview.** Given an unseen input image (**a**) from the test set that was lit by the according directional light ℓ_{src} above it, we relight it towards the directional light ℓ_{dst} in (**b**). To judge the performance, we provide the corresponding ground truth image in (**c**).

ing. Second, *relight* the object according to a desired target lighting. This implies an exact instantiation of the rendering equation [Kajiya, 1986] operating on lighting and surface reflectance representations capable of capturing the true nature of the light-material-geometry interactions. In practice, errors occur due to imperfect parametric models or assumptions. One common approximation is to assume diffuse materials [Barron and Malik, 2015, Sengupta et al., 2018]. Another approximation is smooth lighting, e.g. modeled as low-order spherical harmonics, which cannot produce hard shadows cast from point light sources like the sun. We consider the hard problem of relighting human faces, which are known for both their complex reflectance properties including subsurface scattering, view-dependent and spatially-varying reflectance, but also for our perceptual sensitivity to inaccurate rendering. Recent image-to-image translation approaches rely on deep learning architectures (e.g. Isola et al. [2017]) that make no underlying structural assumption about the (re)lighting problem. Given enough representational capacity, an end-to-end system can describe any underlying process, but is prone to large variance due to over-parameterization, and poor generalization due to physically implausible encodings. Test-time manipulation is also difficult with a semantically meaningless internal state. While this could potentially be alleviated with more training data, acquiring sufficient amounts is very time consuming.

Recent approaches have demonstrated that explicitly integrating physical processes in neural architectures is beneficial in terms of both robust estimates from limited data and increased generalization [Shu et al., 2017, Sengupta et al., 2018, Li et al., 2018]. However, these approaches have focused on the *de-lighting* process, and used simplified physical models for relighting that do not model non-diffuse effects such as cast shadows and specularities.

In this chapter, we bridge the gap between the expressiveness of a physically unconstrained end-to-end approach and the robustness of a physics-based approach. In particular, we consider relighting as an image-to-image translation problem and divide the relighting task into two distinct stages: a physics-based parametric rendering of estimated intrinsic components, and a physics-guided residual refinement. Our image formation model makes the assumption of directional light and diffuse materials. The subsequent refinement process is *conditioned* on the albedo, normals, and diffuse rendering, and dynamically accounts for shadows and any remaining non-diffuse phenomena.

We describe a neural architecture that combines the strengths of a physics-guided relighting process with the expressive representation of a deep neural network. Notably, our approach is end-to-end trained to simultaneously learn to both de-light *and* relight. With the novel dataset of human faces under varying lighting conditions and poses, introduced in the last Chapter 6, we demonstrate that our approach can realistically relight complex non-diffuse materials like human faces. Our directional lighting representation does not require assumptions of smooth lighting environments and allows us to generalize to arbitrarily complex output lighting as a simple sum of point lights. To our knowledge, this is the first work showing realistic relighting effects caused by strong directional lighting, such as sharp cast shadows, from a single input image.

7.2 Related work

Intrinsic images. Intrinsic image decomposition [Barrow and Tenenbaum, 1978] and the related problem of shape from shading [Zhang et al., 1999] have inspired countless derived works. Of interest, Barron and Malik [2015] propose to simultaneously recover shape, illumination, reflectance and shading from a single image and rely on extensive priors to guide an inverse rendering optimization procedure. Other methods recover richer lighting representations in the form of environment maps given the known geometry [Lombardi and Nishino, 2016]. More recent approaches rely on deep learning for the same task, for example using a combination of CNN and guided/bilateral filtering, as we have seen in Chapters 4 and 5 or a pure end-to-end CNN approach [Fan et al., 2018] with the common problem of hard to come by training data. Available datasets may include only sparse relative reflectance judgements [Bell et al., 2014], or sparse shading annotations [Kovacs et al., 2017], which limits learning and quantitative evaluation.

While many previous works focus on lighting estimation from objects [Barron and Malik, 2015, Lombardi and Nishino, 2016, Georgoulis et al., 2018, Meka et al., 2018] or even entire images [Karsch et al., 2014, Zhang et al., 2016, Gardner et al., 2017, Hold-Geoffroy et al., 2017, Zhang et al., 2018a], few papers explicitly focus on the *relighting* problem. Notably, Ren et al. [2015] use a small number of images as input, and, more recently, Xu et al. [2018] learn to determine which set of five light directions is optimal for relighting. Image-to-image translation [Isola et al., 2017] combined with novel multi-illumination datasets [Murmann et al., 2019] has lately demonstrated promising results in full scene relighting.

The explicit handling of moving hard shadows of Duchêne et al. [2015] and Philip et al. [2019] is relevant. While both works use multi-view inputs to relight outdoor scenes, our method works on a single input image to relight faces (our multi-view setup is only used to capture training data). Similar to our work, Yu and Smith [2019] regress to intrinsic components like albedo and normals, but their illumination model is spherical harmonics and therefore does not handle shadows. Sengupta et al. [2019] recently proposed a residual appearance renderer which bears similarities to our learned residual in that it models non-Lambertian effects. Both of the latter works optimize for intrinsic decomposition, whereas we learn end-to-end relighting. Our intrinsic components are only used as a meaningful intermediate representation.

Face relighting. Lighting estimation from face images often focuses on normalization for improving face recognition. For example, Wen et al. [2003] use spherical harmonics (SH) to relight a face image, and Wang et al. [2007] use a Markov random field to handle sharp shadows not modeled by low-frequency SH models. Other face modeling methods have exploited approximate lighting estimates to reconstruct the geometry [Lee et al., 2005, Suwajanakorn et al., 2014] or texture [Li et al., 2014]. In computer graphics similar ideas have been proposed for face replacement [Bitouk et al., 2008, Dale et al., 2011]. Low-frequency lighting estimation from a face has been explored in Shim [2012], Knorr and Kurz [2014], Shahlaei and Blanz [2015]. In contrast, Nishino and Nayar [2004] note that eyes reflect our surroundings and can be used to recover high frequency lighting. More closely related to our work, Calian et al. [2018] learn the space of outdoor lighting using a deep autoencoder and combine this latent space with an inverse optimization framework to estimate lighting from a face. However, their work is restricted to outdoor lighting and cannot be used explicitly for relighting.

Of particular relevance to our work, neural face editing [Shu et al., 2017] and the related SfSNet [Sengupta et al., 2018] train CNNs to decompose a face image into surface normals, albedo, and SH lighting. These approaches also impose a loss on the intrinsic components, as well as a rendering loss which ensures that the *combination* of these components is similar to the input image. FRADA [Le and Kakadiaris, 2019] revisited the idea of relighting for improving face recognition with face-specific 3D morphable models (similar to Shu et al. [2017]), while we do not impose any face-specific templates. Single image portrait relighting [Zhou et al., 2019a] bypasses the need for decomposition, while still estimating the illumination to allow editing. In a similar line of work, Sun et al. [2019] capture faces in a light stage using one light at a time, but then train using smoother illuminations from image based rendering which leads to artifacts when exposed to hard cast shadows or strong specularities. Recently, Meka et al. [2019] also used light stage data and train to relight to directional lighting as we do. However, their network expects a pair of images captured under spherical gradient illumination at test time, which can only be captured in a light stage. The portrait lighting transfer approach of Shu et al. [2018] directly transfers illumination from a reference portrait to an input photograph to create high-quality relit images, but fails when adding or removing non-diffuse effects.

7.3 Architecture of the Relighting Network

The following two sections first introduce an image formation process (Sec. 7.3.1) and then describe its integration into a physics-based neural relighting architecture (Sec. 7.3.2).

7.3.1 Image formation process

The image formation process describes the physics-inspired operations transforming the intrinsic properties of a 3D surface to a rendered output. The majority of physics-based works are based on specific instantiations of the rendering equation [Kajiya, 1986] (introduced in Sec. 3.2.5):

$$L_o(\omega_o) = \int_{\omega_i \in \Omega} f(\omega_i, \omega_o) L_i(\omega_i) \langle \mathbf{n}, \omega_i \rangle \, \mathrm{d}\omega_i, \tag{7.1}$$

where ω_i, ω_o are the incoming and outgoing light directions relative to the surface normal $\mathbf{n}$ at the surface point $\mathbf{X}_j$. $L_i(\omega_i)$ and $L_o(\omega_o)$ are the corresponding (ir)radiances, $f(\cdot,\cdot)$ is the BRDF describing the material's reflectance properties, and $\langle \mathbf{n}, \omega_i \rangle$ is the attenuating factor due to Lambert's cosine law.

This model is often simplified further by assuming a *diffuse* decomposition into albedo $a \in \mathbb{R}$ and shading $s \in \mathbb{R}$,

$$a = f(\omega_i, \omega_o), \qquad \text{[const.]} \tag{7.2}$$

$$s = \int_{\omega_i \in \Omega} L_i(\omega_i) \langle \mathbf{n}, \omega_i \rangle \, \mathrm{d}\omega_i. \tag{7.3}$$

Non-diffuse effects. A realistic relighting approach must relax modeling assumptions to allow complex reflectance properties such as subsurface scattering, transmission, polarization, etc., and, if using Eq. (7.2), specularities. Unfortunately, learning a spatially varying BRDF model $f(\omega_i, \omega_o)$ based on a non-parametric representation is infeasible: assuming an image size of 512×768 and a pixelwise discretization of the local half-angle space [Matusik, 2003] would result in 1.7×10^{12} parameters. Learning a low-dimensional representation in terms of semantic parameters [Burley, 2012] seems like a viable alternative but is still prone to overfitting and cannot account for light-material-interactions outside of its parametric space.

We propose a hybrid approach and decompose f into two principled components, a diffuse albedo a and a light-varying residual r:

$$f(\omega_i, \omega_o) = a + r(\omega_i, \omega_o). \tag{7.4}$$

This turns Eq. (7.1) into

$$L_o(\omega_o) = as + \int_{\omega_i \in \Omega} r(\omega_i, \omega_o) L_i(\omega_i) \langle \mathbf{n}, \omega_i \rangle \, \mathrm{d}\omega_i. \tag{7.5}$$

For a light source with intensity $I(\omega_i)$, we can identify $L_i(\omega_i) = I(\omega_i)v(\omega_i)$, where $v \in \{0,1\}$ is the binary visibility of the light source. Under the assumption of a single directional light source from $\widetilde{\omega}_i$, we integrate over one point only, so if we further write $\widetilde{r}(\widetilde{\omega}_i, \omega_o) = r(\widetilde{\omega}_i, \omega_o) I(\widetilde{\omega}_i) \langle \mathbf{n}, \widetilde{\omega}_i \rangle$, we can re-formulate our rendering equation Eq. (7.1) to

$$L_o(\omega_o) = (as + \widetilde{r}(\widetilde{\omega}_i, \omega_o)) \cdot v(\widetilde{\omega}_i). \tag{7.6}$$

This will be the underlying image formation process in all subsequent sections. While as captures much of the diffuse energy across the image according to an explicit generative model, the residual $\widetilde{r}(\widetilde{\omega}_i, \omega_o)$ accounts for physical effects outside of the space representable by Eq. (7.2) and is modeled as a neural network (akin to Sengupta et al. [2019]). We do not impose any assumptions on $r(\widetilde{\omega}_i, \omega_o)$, even allowing light subtraction, but do enforce a to be close to the ground truth albedo of a diffuse model which we obtain from photometric stereo [Xiong et al., 2014].

Discussion of our physics guided relighting approach. While directional lights are conceptually simple, they lead to challenging relighting problems. Our combination of an explicit diffuse rendering process and a non-diffuse residual (with implicit shading) serves several purposes:

1. Describing most of the image intensities with a physics-based model means the output image will be more consistent with the laws of physics;

2. Describing specular highlights as residuals alleviates learning with a CNN;

3. Leaving the residual unconstrained (up to ground truth guidance) allows us to model effects that are not explainable by the BRDF, such as subsurface scattering and indirect light;

4. Modeling visibility explicitly helps, because the simple diffuse model does not

handle cast shadows. At the same time, expecting the residual to take care of shadow removal by resynthesis is much harder than just masking it.

7.3.2 Physics-guided relighting

Presented with an input image I_{src} that was lit by an input illumination ℓ_{src}, our goal is to learn a generator G, relighting I_{src} according to a desired output illumination ℓ_{dst},

$$G(I_{\text{src}}, \ell_{\text{src}}, \ell_{\text{dst}}) = I_{\text{dst}}. \tag{7.7}$$

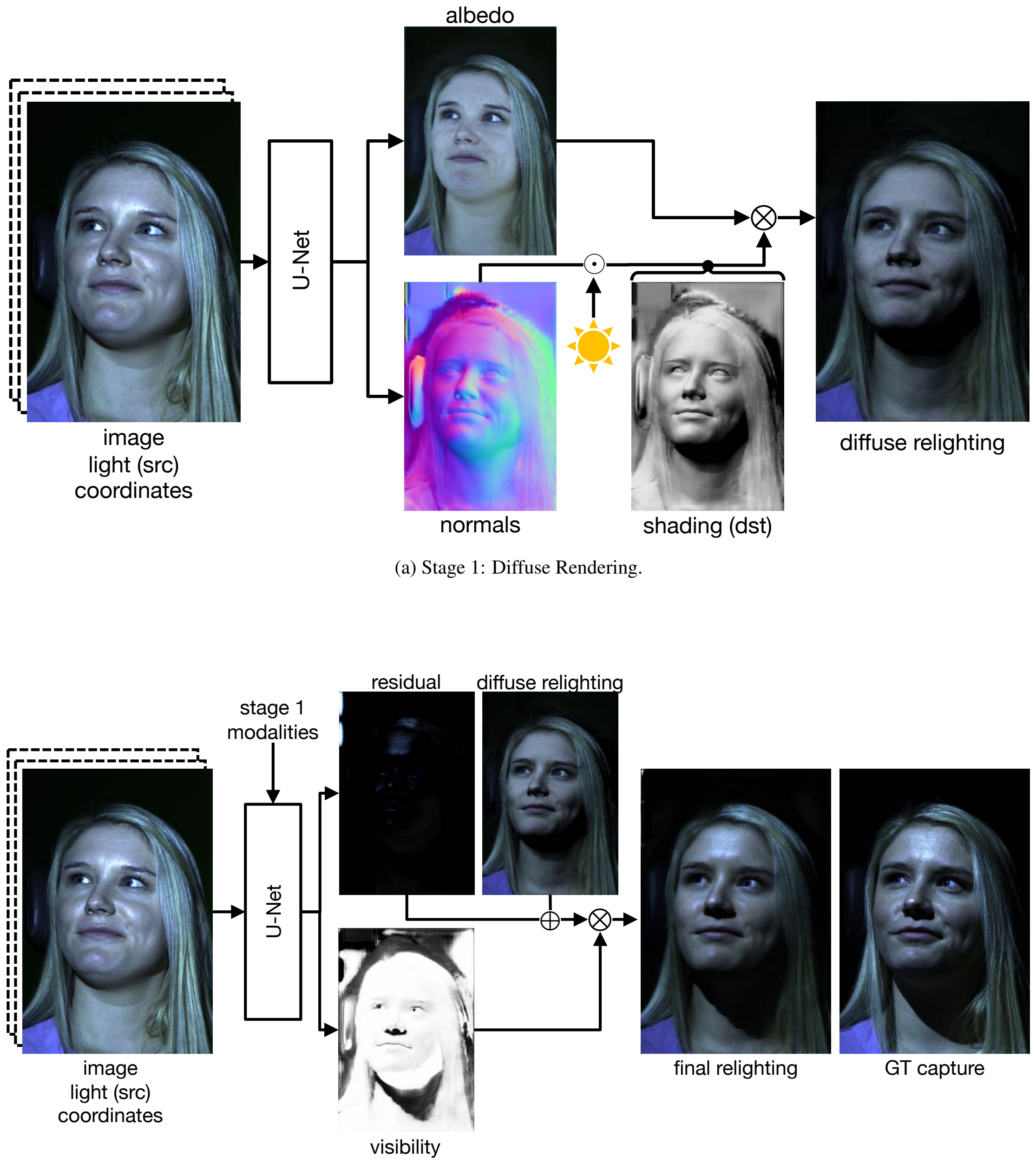
At training time, we assume ℓ_{src} and ℓ_{dst} to be directional lights, which is known to be a particularly challenging instance of relighting and accurately matches our captured data (see Chapter 6). At test time, this is not a limitation, since we can easily fit a set of directional lights to an environment map to perform more complex relighting (see Sec. 7.6).

Our physics-guided approach to solving the relighting task consists of a recognition model inferring intrinsic components from observed images (de-lighting) and a generative model producing relit images from intrinsic components (relighting). While the recognition model takes the form of a traditional CNN, the generative model follows our image formation process (Sec. 7.3.1) and is represented by structured layers with clear physical meaning. In line with Eq. (7.6), we implement the latter as a two-stage process:
(Stage 1) Using the desired target lighting, we compute shading from predicted normals and multiply the result with our albedo estimate to obtain a diffuse render;
(Stage 2) Conditioned on all intrinsic states predicted in stage 1, we infer a residual image and a visibility map, which we combine with the diffuse render according to Eq. (7.6). An illustration of this pipeline is shown in Fig. 7.2. Since all its operations are differentiable and directly stacked, this allows us to learn the proposed model in an end-to-end fashion from input to relit result.

We introduce losses for all internal predictions, i.e., albedo, normals, shading, diffuse rendering, visibility, and residual. We emphasize the importance of using the right loss function and refer to Sec. 7.5.1 for a comprehensive study. In order to obtain the corresponding guidance during training, we use standard photometric stereo reconstruction [Xiong et al., 2014].

7.4 Data

Our data from Chapter 6 comprises a diverse set of facial expressions captured under various lighting conditions. Capturing with 6 cameras and 32 white LEDs lit in sequence gives us the data necessary to record each of the 482 poses/expressions from 21 subjects under 192 light configurations each, which provides a total of 2,961,408 relighting pairs for training that provide the necessary intrinsic layers as described in Sec. 6.4.



(a) Stage 1: Diffuse Rendering.

(b) Stage 2: Non-Diffuse Residual.

Figure 7.2: **Physics-guided relighting with structured generators.** Our generator consists of two stages modeling diffuse and non-diffuse effects. All intrinsic predictions are guided by losses w.r.t. photometric stereo reconstructions. **(a)** We use a U-Net with grouped convolutions to make independent predictions of the intrinsic components. Predicted normals are always re-normalized to unit vectors. Given a desired output lighting, we compute shading from normals and render a diffuse output. **(b)** Conditioned on all modalities inferred in (a), we predict a non-diffuse residual and binary light visibility map to model specularities, cast shadows, and other effects not captured by our instance of the rendering equation.

Augmentation. Modern neural architectures are much better at interpolation than extrapolation. It is therefore critical to cover the space of valid light transports as well as possible. To this end, we perform a series of data augmentation steps in an attempt to establish strong correlations throughout the parametric relighting space:

1. We flip all training images along the horizontal and vertical axis, increasing the effective dataset size by a factor of 4. Note that this also requires adaptation of the corresponding light directions and normals;
2. We perform a linear scaling $\mathbf{x}' = s \cdot \mathbf{x}$, $s \sim \mathcal{U}_{[0.6,1.1]}$, of the images, shading, residuals and light intensities. In practice, we did not observe substantial benefits compared to training without scaling;
3. We randomly perturb the light calibration with Gaussian noise $n \sim \mathcal{N}(0, 0.01^2)$ to improve generalization and account for minimal calibration errors;
4. For quantitative results, we perform a spatial rescaling to $\frac{1}{8}$th of the original image resolution (135×256), train on *random crops* of size 128×128 and test on center crops with the same resolution to have comparability with SfSNet. Qualitative results are generated by rescaling to $\frac{1}{2}$ of the original resolution (540×1024), trained on *random crops* of size 512×768 and tested on center crops of that resolution.

7.5 Experiments

Our models were implemented using PyTorch [Paszke et al., 2017] with a U-Net [Ronneberger et al., 2015] generator and PatchGAN [Goodfellow et al., 2014] discriminator (for the final relit image) based on the implementations provided by pix2pix [Isola et al., 2017]. The images in our dataset are camera RAW, represented as 16-bit linear RGB values nominally in the range $[0, 1]$. There is under- and over-shoot headroom, but for training and evaluation we clamp them into this range and linearly transform to $[-1, 1]$ as input into the network.

7.5.1 Evaluation metric

Quantitatively comparing the relit prediction $\hat{I}_{\text{dst}}$ of the generator against the ground truth I_{dst} requires an appropriate error measure. We consider the L_1 and L_2 norms but recognize that they do not coincide with human perceptual response. We also consider the "Learned Perceptual Image Patch Similarity" (LPIPS) loss suggested by Zhang et al. [2018b] using the distance of CNN-features pretrained on ImageNet. Another prevailing metric of image quality assessment is structural similarity (SSIM) [Wang et al., 2004] and its multi-scale variant (MS-SSIM) [Wang et al., 2003]. In our evaluation, we use the corresponding dissimilarity measure $\text{DSSIM} = \frac{1-\text{SSIM}}{2}$, and likewise for MS-SSIM, to consistently report errors.

Table 7.1: **Loss selection.** We explore the influence of different training losses and evaluation metrics on direct image-to-image translation ("pix2pix") and our structured guidance approach ("ours"). For each class, we show validation scores for all pairwise combinations of 5 training losses (rows) and the same 5 evaluation metrics (columns). The best model for each evaluation metric is shown in bold.

Model	Training Loss	Evaluation Metric				
		L_1	L_2	LPIPS	DSSIM	MS-DSSIM
pix2pix	L_1	.0452	.0067	.2564	.1707	.1144
	L_2	.0516	.0082	.2663	.1911	.1369
	LPIPS	.0424	.0062	**.1868**	.1440	.0992
	DSSIM	**.0406**	**.0055**	.2138	**.1378**	.0930
	MS-DSSIM	.0422	.0058	.2358	.1547	**.0913**
ours	L_1	.0406	.0055	.2237	.1484	.0913
	L_2	.0415	.0056	.2302	.1547	.0953
	LPIPS	.0365	.0048	**.1701**	.1308	.0803
	DSSIM	**.0362**	**.0045**	.2008	**.1270**	**.0793**
	MS-DSSIM	.0410	.0055	.2165	.1470	.0910

LPIPS: "Learned Perceptual Image Patch Similarity" [Zhang et al., 2018b];
DSSIM: structured dissimilarity; MS-DSSIM: multi-scale DSSIM

When defining the loss function during training, the same choices of distance metrics are available. To densely evaluate their performance, we report in Table 7.1 the results of training all intrinsic layers with the same loss function from the options above. Surprisingly, we conclude that, for our task, using DSSIM for the training loss consistently leads to models which generalize better on the validation set using most of the error metrics. The only exception is evaluation using the LPIPS metric, which is better when also trained using this metric. Therefore, we chose the models trained on DSSIM for computing the final test results.

7.5.2 Baseline comparisons

We now provide quantitative and qualitative comparisons of our proposed architecture, to related work.

Baselines

Our baselines comprise the following set of related methods:

PMS. To understand the lower error bound of a diffuse model, we take albedo A and shading S from photometric stereo (PMS; [Xiong et al., 2014]) and diffuse render via $A \odot S$. We note that this model has access to all light configurations at the same time, with the desired target illumination amongst them. Since this gives an unfair advantage, we do not highlight results for this model in Table 7.2.

SfSNet (pretrained). We take the pretrained network of SfSNet [Sengupta et al., 2018] and apply it to our data by using their decomposition into albedo and normals, but ignoring the output spherical harmonics estimate. Instead, we compute target shading as the dot product of ℓ_{dst} and normals to have a direct comparison with our assumption of directional light and present the result after diffuse rendering.

SfSNet (retrained). We retrain SfSNet [Sengupta et al., 2018] on the calibrated PMS data and also provide the source illumination as input, to which our model has access as well. Compared to the pretrained model above, this baseline can be seen as a fairer comparison to SfSNet.

Pix2pix/no guidance. The arguably simplest way to learn Eq. (7.7) from data is to instantiate G as a traditional neural network consisting of a series of generic convolutional layers with no semantic meaning and no knowledge of the image formation process.We adapt the pix2pix translation GAN [Isola et al., 2017] to our use case by conditioning the generator on the input image as well as the source and target illumination. This ensures an objective comparison with our more structured model, which also has access to lighting information.

Evaluation

Qualitative evaluation. We compiled a collection of qualitative results in Figs. 7.3 and 7.4. While the shading of SfSNet is smooth, it has a bias towards an albedo which probably resembles skin color in their training data and does not distinguish well between different skin tones and hair. As expected, retraining their model on our data leads to more accurate results. Still, due to the diffuse assumption, it looks flat compared to our more expressive model. It misses specularities and surface normals orthogonal to the light direction are missing ambient light from inter-reflections. The pix2pix model generates promising results, but its domain-agnostic architecture often leads to physically implausible artifacts, such as missing shadows. In comparison, the predictions of our proposed architecture are typically the most realistic, mainly due to its need to estimate a consistent albedo, as can be seen for example at the hair in the first row of Fig. 7.3. The last row shows a hand occluding the face, leading to strong cast shadows that have to be introduced and removed. Our model using intrinsic guidance gracefully handles that case.

While our data allows foreground masking computed from PMS, we show the full image predictions for better judgment. At test time, an off-the-shelf face matting approach, e.g. Wadhwa et al. [2018], could be used for cleaning the predictions.

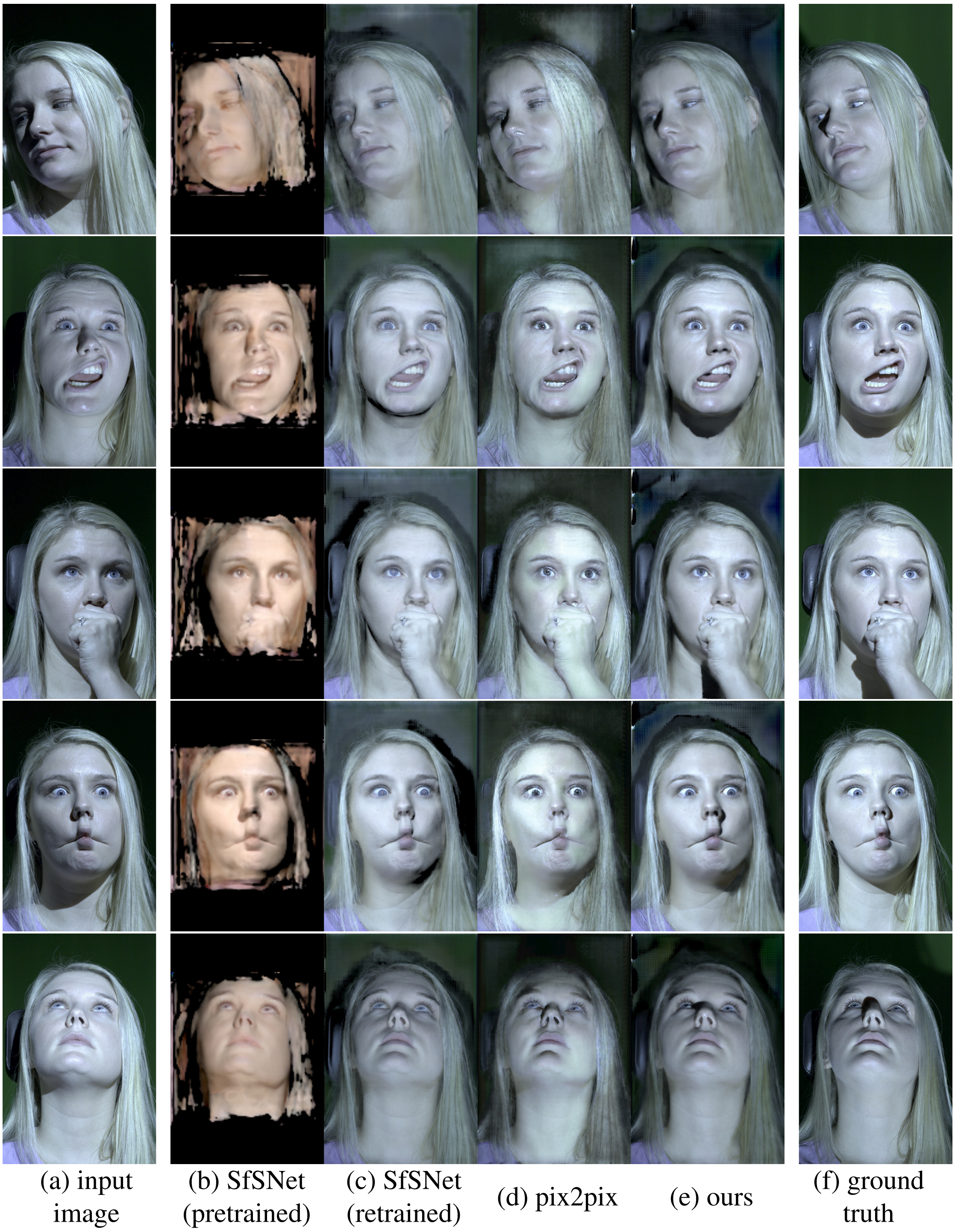
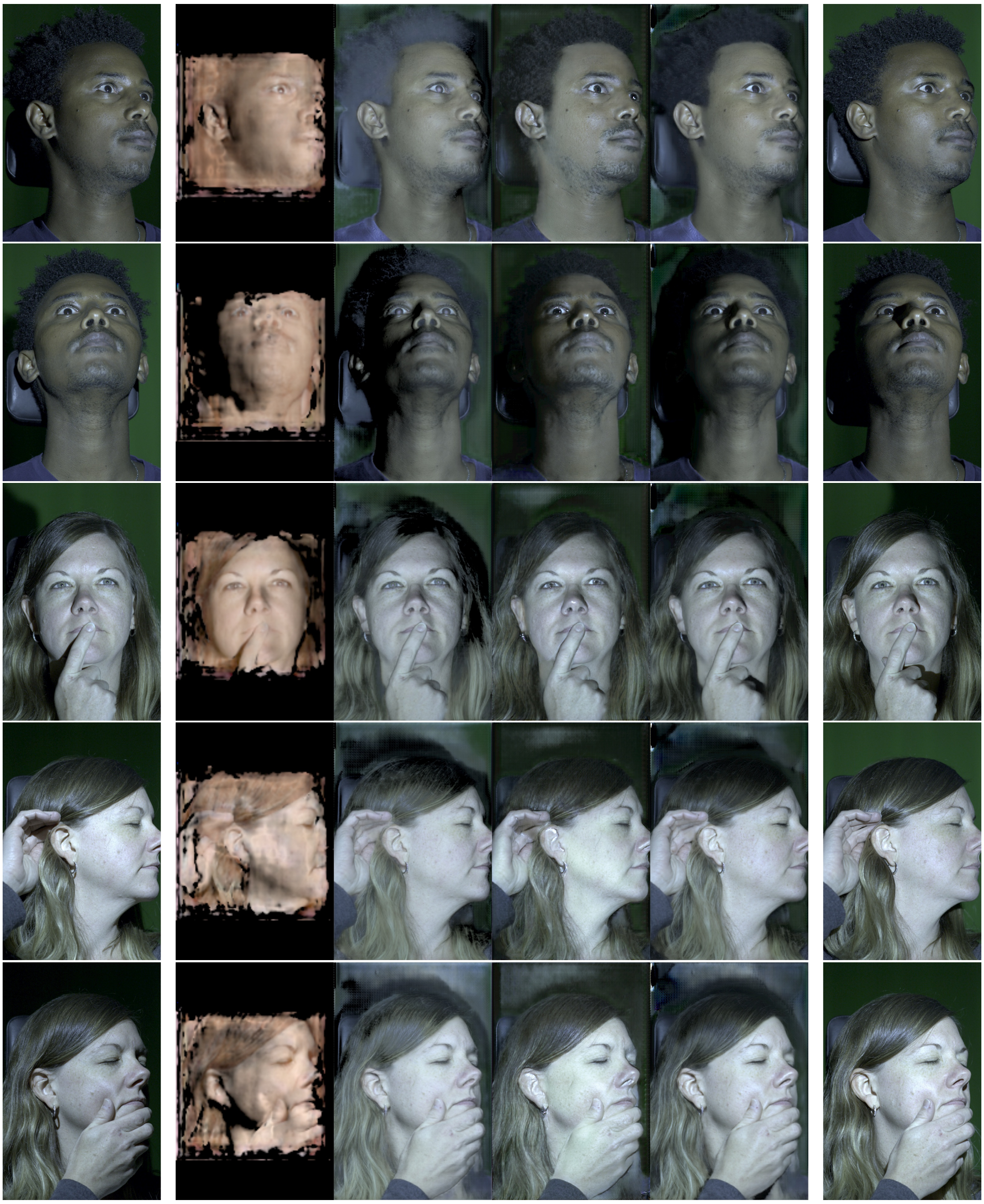


Figure 7.3: **Qualitative evaluation on unseen subjects and expressions.** We compare relighting (**a**) the input image with (**b/c**) pretrained and retrained variants of SfSNet, (**d**) pix2pix, and (**e**) our model. In (**f**), we show the ground truth capture of the given target illumination. Notice our model's ability to generate realistic shadows and specular highlights. All results have been converted from linear to sRGB. See Fig. 7.4 for more results.



(a) input image

(b) SfSNet (pretrained)

(c) SfSNet (retrained)

(d) pix2pix

(e) ours

(f) ground truth

Figure 7.4: **Qualitative evaluation on unseen subjects and expressions (continued).** Additionally to Fig. 7.3, we compare relighting (**a**) the input image with (**b/c**) pretrained and retrained variants of SfSNet, (**d**) pix2pix, and (**e**) our model. In (**f**), we show the ground truth capture of the given target illumination. Notice our model's ability to generate realistic shadows and specular highlights. All results have been converted from linear to sRGB.

Table 7.2: **Quantitative evaluation.** We show a quantitative comparison of our approach to baseline methods. Performance on the test set is reported under the assumption of both known ('with') and unknown ('w/o') source illumination. All models have been trained with the DSSIM loss.

L	Model	Evaluation Metric				
		L_1	L_2	LPIPS	DSSIM	MS-DSSIM
with	SfSNet (R)	.0636	**.0121**	.2508	.1840	.1277
	pix2pix	.0668	.0144	.2430	.1832	.1328
	ours	**.0609**	.0123	**.2144**	**.1618**	**.1138**
w/o	SfSNet (P)	.1359	.0424	.4703	.3221	.3121
	pix2pix	.0815	.0189	.2783	.2076	.1623
	ours	**.0684**	**.0142**	**.2273**	**.1763**	**.1316**
	PMS	.0391	.0047	.1630	.1125	.0561

L: access to source illumination; LPIPS: Zhang et al. [2018b];
DSSIM: structured dissimilarity, MS-DSSIM: multi-scale DSSIM;
PMS: photometric stereo; P/R: (p)retrained

We encourage the reader to look at more qualitative results of this type on our project page[1], where we also show relighting under a moving target illumination.

Quantitative evaluation. In Table 7.2 (first block), we analyze the quantitative performance of our model in the described scenario with known source illumination ℓ_{src}. The test set comparison with the diffuse SfSNet and the unstructured pix2pix baseline confirms the importance of our physics-based guidance and non-Lambertian residuals. An extension of our model without the assumption of known source illumination (second block) will be discussed in Sec. 7.6.1. For reference, the PMS reconstruction, restricted to a diffuse model but computed from multiple images, is also shown.

7.5.3 Additional qualitative comparisons

We provide more qualitative comparisons to the following recent portrait relighting approaches in Fig. 7.5.

The *mass transport relighting* [Shu et al., 2018] approach is different in that it defines the target lighting as that of *another* portrait. To match those conditions, we set the desired output to directly be the ground truth reference image. Despite these optimal conditions, Shu et al. [2018] fails to generate specular highlights and cast shadows, which are well-captured by our technique.

[1] https://lvsn.github.io/face-relighting

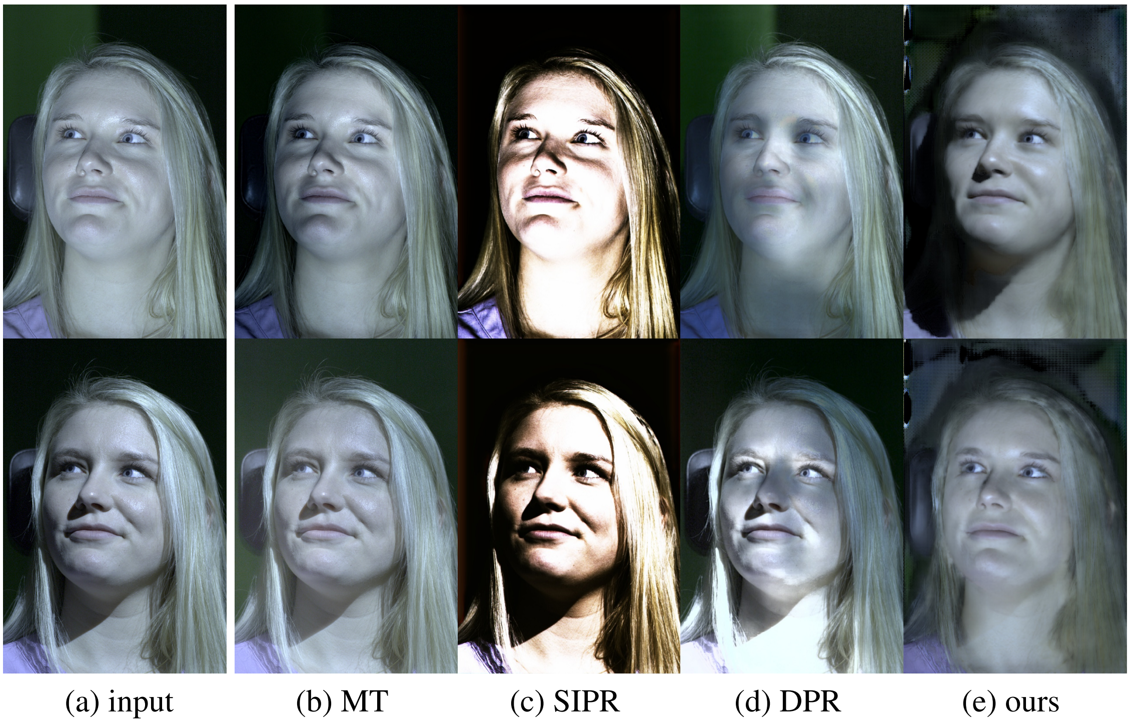


Figure 7.5: **Additional qualitative comparisons.** We relight the input in the first row to the input in the second row, and vice versa. Results in the Mass Transport (MT) [Shu et al., 2018] approach and Single Image Portrait Relighting (SIPR) [Sun et al., 2019] were provided by the authors. For Deep Portrait Relighting (DPR) [Zhou et al., 2019a], we use their provided code and approximate the light directions manually.

Single image portrait relighting using an environment map is learned by Sun et al. [2019]. Training images are produced by compositing multiple 'one light at a time' captures. As already discussed in Sun et al. [2019], the method fails on strong light.

Finally, Zhou et al. [2019a] learn to do *deep portrait relighting* using a spherical harmonics representation, which also handles smooth lighting exclusively, see Fig. 7.5.

7.5.4 Dynamic input lighting

We illustrate the consistency and robustness of our approach by relighting multiple source light configurations to the same target lighting (see Figs. 7.6 and 7.7). Our examples cover a wide spectrum of source illuminations, including strong and challenging directional lights originating on the side of the face. The extreme cases on the far left and right, in particular, require the removal of strong shadows. The noise in these low light areas is high, and visual cues are weak, which makes consistent relighting challenging. Please also refer to the video on the project page, where we show results in a dynamic environment with moving lights.

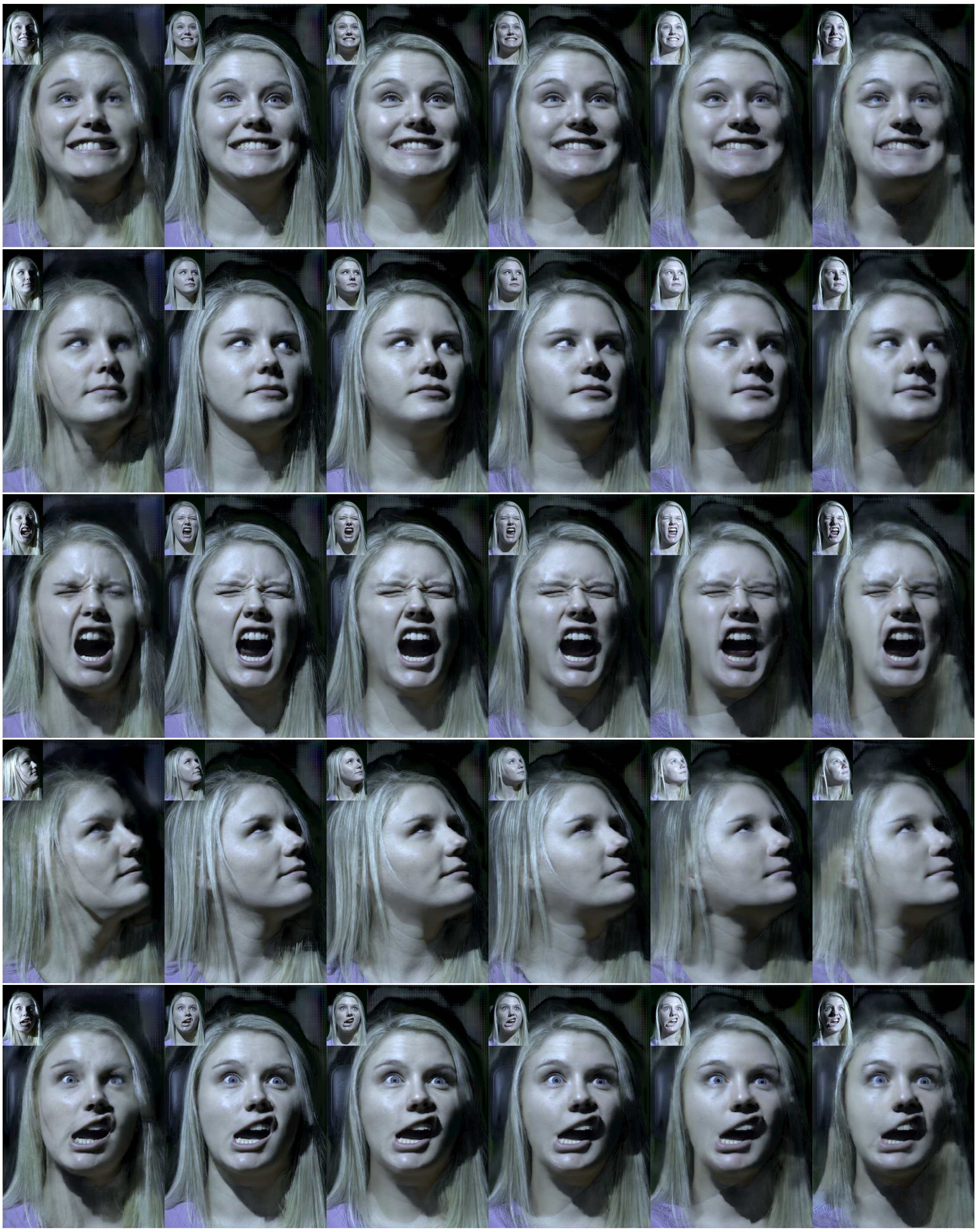


Figure 7.6: **Dynamic input lighting.** In each row, we show a different facial expression that we relight from different input light configurations (columns; see small inset) to the same target light configuration. All results have been converted from linear to sRGB. See Fig. 7.7 for more results.

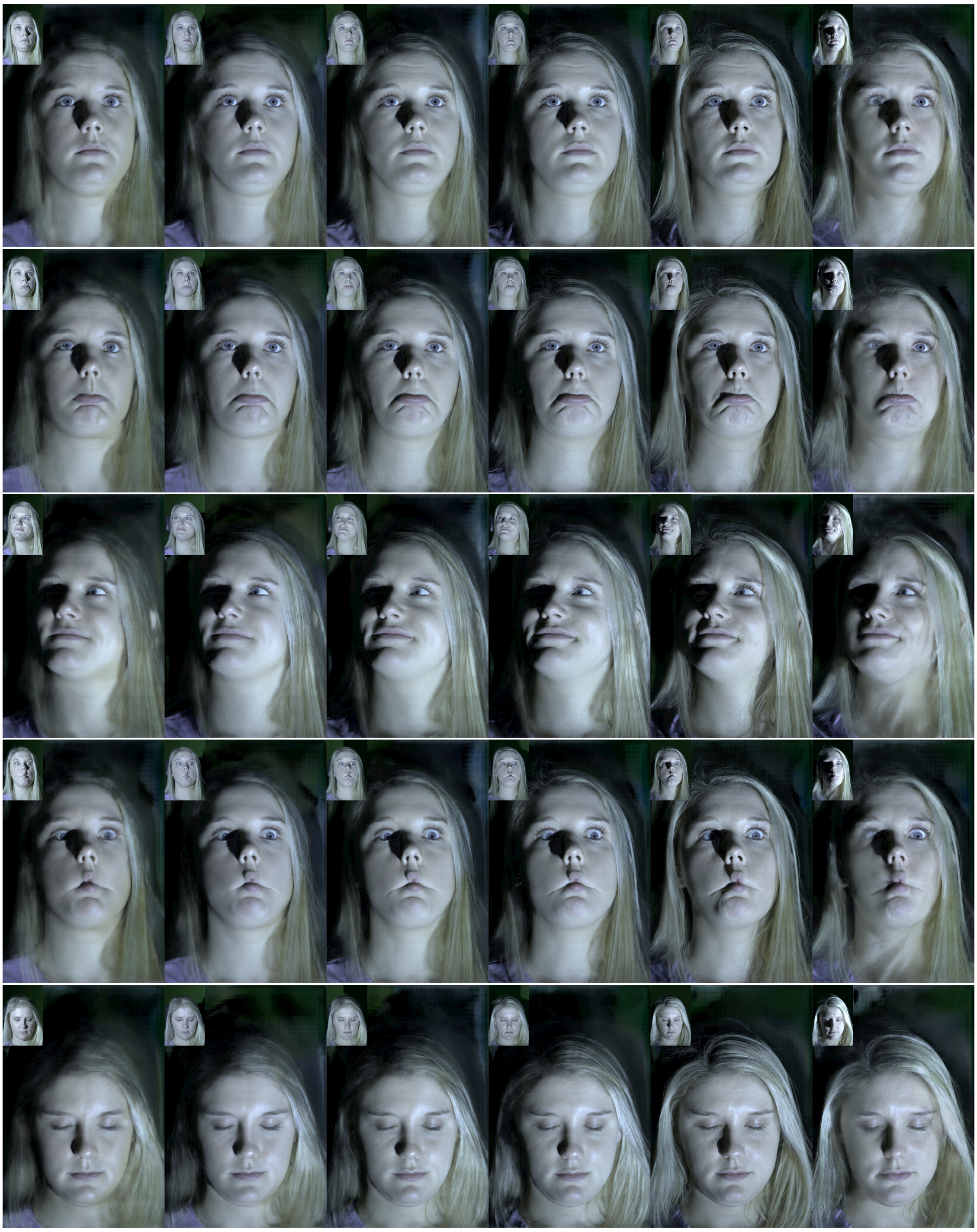
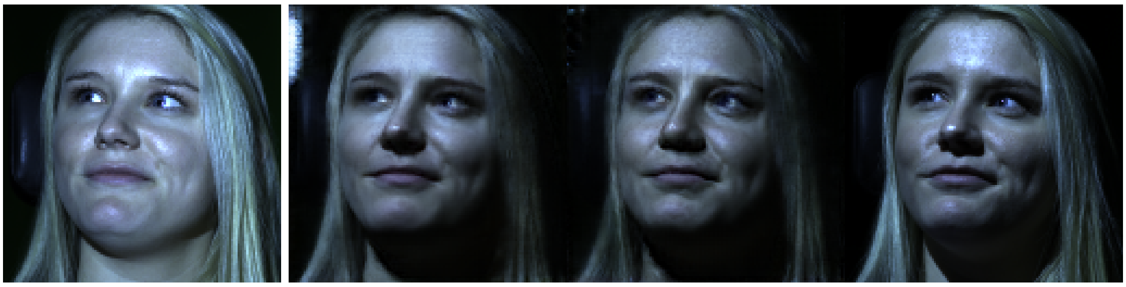


Figure 7.7: **Dynamic input lighting (continued).** In each row, we show a different facial expression that we relight from different input light configurations (columns; see small inset) to the same target light configuration. All results have been converted from linear to sRGB.



(a) input (b) with (c) w/o (d) GT

Figure 7.8: **Relighting with unknown source illumination.** (**a**) Input image. (**b/c**) Our relighting model *with* and *without* access to source illumination. (**d**) Ground truth output.

7.6 Extensions

We now demonstrate that our model successfully generalizes to different scenarios, including unknown input illumination, relighting with environment maps, relighting images captured in the wild and how our model performs on generic objects outside the domain of human faces.

7.6.1 Relighting with unknown source illumination

While we cannot remove the need for the target illumination, information about the source illumination is already contained in the input image, allowing for implicit learning of ℓ_{src}. To illustrate our model's ability to extract these signals, we trained a version of our architecture without explicit access to the input lighting; these results, as well as a comparison to the corresponding baseline variants, are shown in Table 7.2 (second block) and Fig. 7.8. As expected, all models incur a small drop in performance compared to their counterparts with explicit knowledge. Nonetheless, our model *without* access to the source illumination achieves similar (and in some cases better) performance than the pix2pix model *with* access to the source illumination.

7.6.2 Relighting with environment maps

Directional light sources are a very general representation, and our approach easily allows for relighting with environment maps as visualized in our examples of relit scenes with five environment maps in Figs. 7.9 and 7.10. While more principled approaches like importance sampling are available [Agarwal et al., 2003], for illustrative purposes here we simply sample environment maps by downscaling them to 64×32 pixels and instantiating our relighting prediction with one light direction per pixel. Since light is additive, we then mix the resulting predictions according to their color and intensity. More results showing temporal stability under dynamically changing environment maps can be found in the video on our project page.

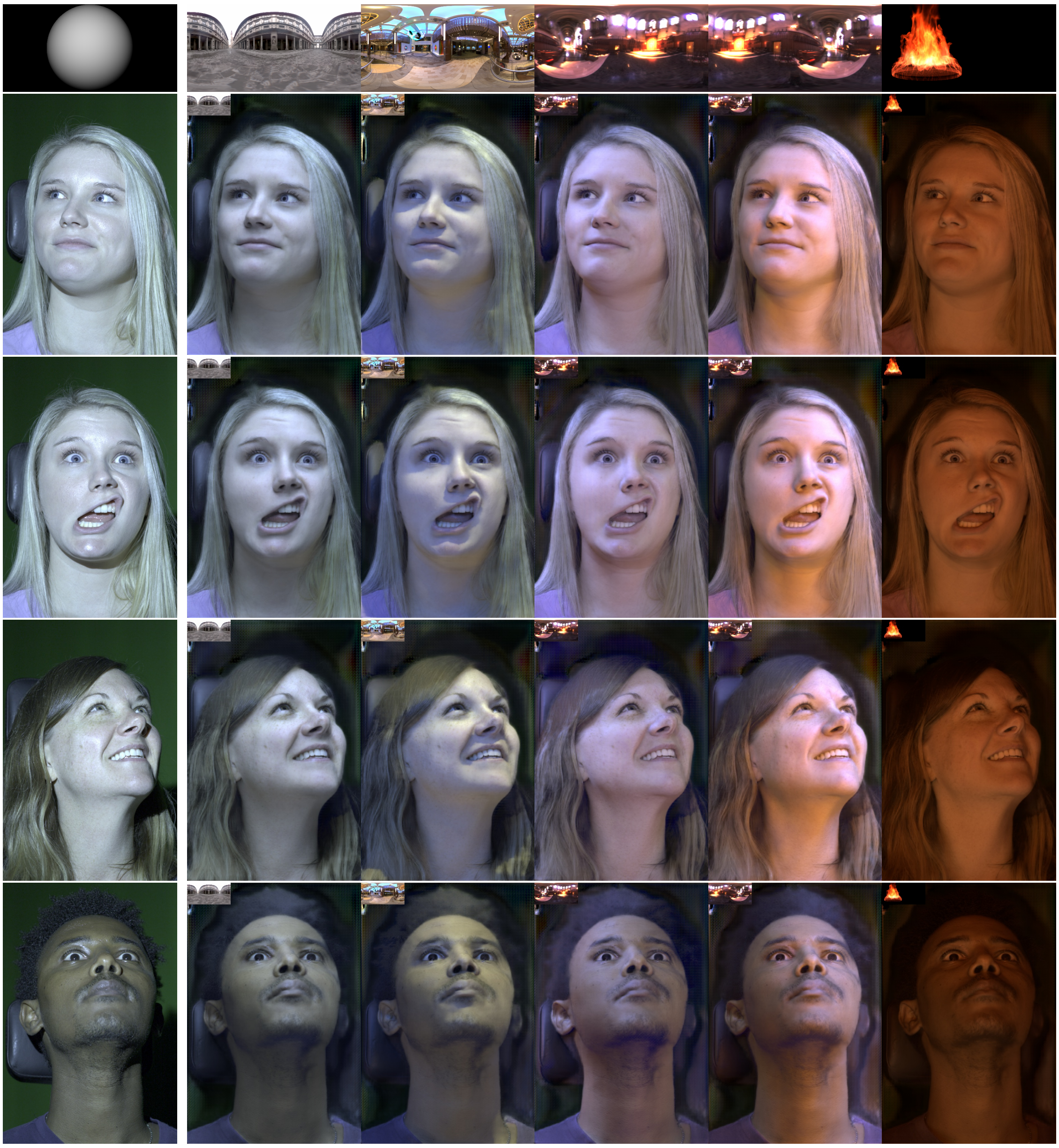


Figure 7.9: **Relighting with Environment Maps.** We relight the input (first column) w.r.t. 5 different environment maps (first row), ordered from cold (second column) to warm (sixth column) dominating color temperatures. All results have been converted from linear to sRGB. See Fig. 7.10 for more results.

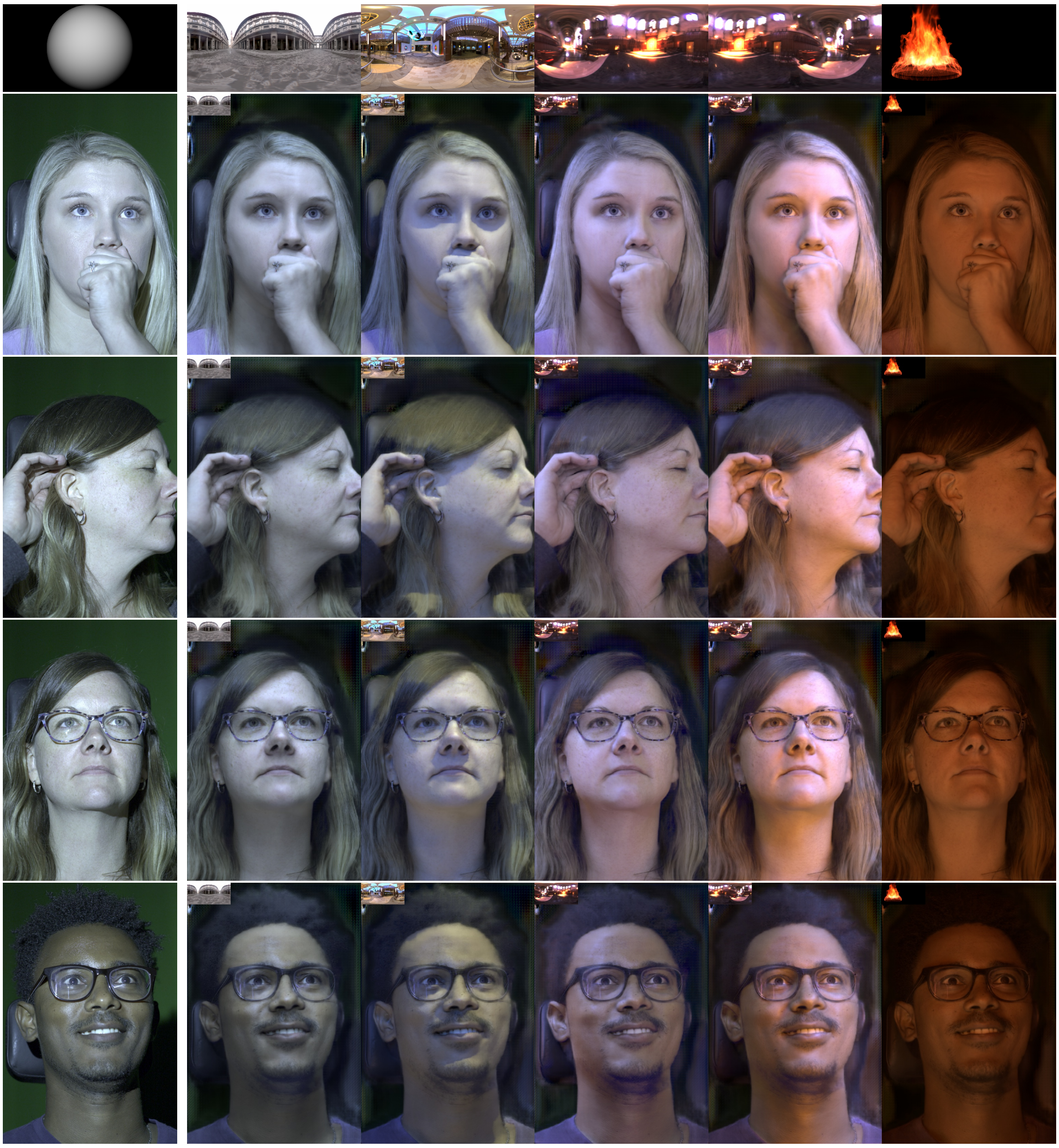


Figure 7.10: **Relighting with Environment Maps (continued).** We show the same type of visualization as in Fig. 7.9 but focus on more challenging scenarios, such as expressions affecting the face topology and glasses. All results have been converted from linear to sRGB.

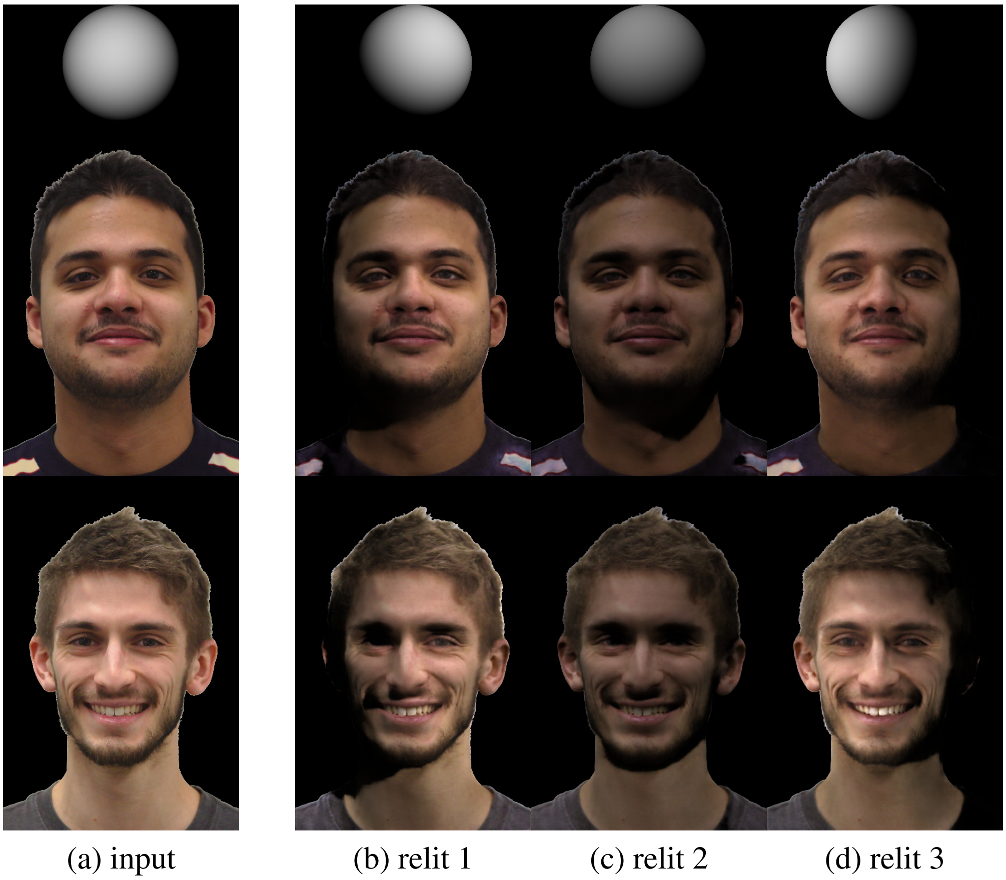
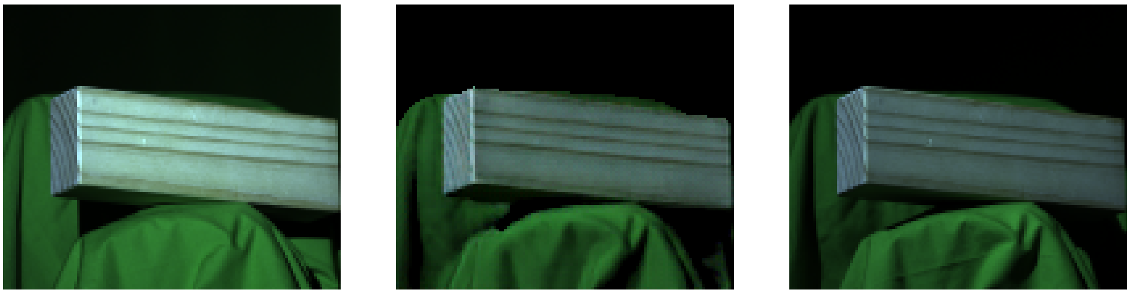


Figure 7.11: **Relighting in the wild.** We consider portraits not taken in our capture environment (**a**) and relight them with respect to 3 different target point lights (**b-d**).

7.6.3 Relighting in the wild

To demonstrate generalization outside the domain of our lab-captured dataset, we conducted experiments using pictures taken in an office environment with a Canon EOS 5D Mark III and visualize results of relighting towards three target lights in Fig. 7.11.

We emphasize the practical difficulties of relighting those portraits, including unknown discrepancies in the imaging pipeline (camera sensor, illuminant color, image processing etc.) and approximation of the unknown source lighting. Since the portraits are taken under uncontrolled office lighting, this results in images which are diffusely lit by multiple input light sources. Although this violates our model assumption of a single directional light source, we run our model using an input light direction ℓ_{src} in the image that would simply light the portrait centrally. To compensate for different illumination colors in the input, we compute the mean of a 51×76 center patch and apply a linear color transform towards our data distribution. The background is masked by hand, which could be automated [Wadhwa et al., 2018]. Note that we can show neither quantitative nor qualitative comparisons to ground truth relit images since they do not exist.



(a) input

(b) relit

(c) GT

Figure 7.12: **Relighting of Generic Objects.** Example of a wooden block relit by our model after training on other poses of that object. (**a**) Input image. (**b**) Relit output with masked background. (**c**) Ground truth output.

The dynamic component of relighting in the wild can be seen in the video on our project page[2] and more static results are found in Fig. C.1 of the Appendix.

7.6.4 Relighting of Generic Objects

Considering applications like teleconferencing in AR, our main focus was on relighting portraits. Due to their complex interactions with light, human faces (and skin in general) provide a great testbed for relighting. Nonetheless, our approach does not rely on any explicit face template and is thus applicable to arbitrary objects. To illustrate this fact, we train our model on several poses of a wood block and show a relighting result of an unseen pose in Fig. 7.12. Note that full abstraction to arbitrary object classes would require substantially more training data and is thus left for future work.

7.7 Conclusion

We propose a method which learns to relight a face with strong directional lighting, accurately reproducing non-diffuse effects such as specularities and hard-cast shadows. We introduce a structured relighting architecture with semantic decomposition into, and subsequent re-rendering from intrinsic components. On our challenging light-stage dataset with directional light, the integration of an explicit generative rendering process and a non-diffuse neural refinement layer within an end-to-end architecture proved to be superior. We found that our model tends to produce shadows and albedo maps that are qualitatively closer to reality than all baselines. A more structured approach also has advantages beyond raw performance, including better interpretability, as well as explainability and the possibility for direct manipulation and extraction of its semantically meaningful intermediate layers for downstream tasks. A comprehensive study of different losses and

[2] https://lvsn.github.io/face-relighting

evaluation metrics highlighted the benefits of training on a perceptual loss over more traditional choices. Thus, our model can be useful in a wide range of face-centric and more general applications in, e.g., augmented reality.

Limitations and future work. While our model generally deals well with cast shadows in the input image (see Fig. 7.5), results get worse when there is so little light that the camera mostly returns noise. Although a crude infilling for those pixels based on context can be learned, an interesting future direction would be to identify these pixels explicitly. A dedicated infilling method, conditioned on the properly relit parts of the image and other intrinsic layers, could be applied to them. To cancel ambient input illumination (as in Fig. 7.11), future work could experiment with taking a flash photo and subtracting a second non-flash photo (in linear color space).

Chapter 8

Conclusions

In this thesis, we looked into multiple aspects of intelligent systems and how they are interconnected through intrinsic images, the separation of images into their constituent layers which is giving important insights in a powerful vision system.

We first covered the field of *Robotics* and understood the need for planning coherent future moves in a multi robot system. It was achieved by assigning dynamic priority roles among the individual UAVs, which they use to accordingly scale their traveling forces. This leadership principle allows to effectively and efficiently visit set locations in parallel while never losing connectivity in the overall group. An extension of this setup where a human is operating one of the robots makes it obvious that more artificial intelligence is necessary to be able to perceive the world in the first place. In order to do so, an illumination independent representation of a scene is deemed beneficial.

Because also humans rely on vision as their most important sense of perceiving the world, we therefore shifted the focus to *Computer Vision*, to perceive and understand the surroundings through cameras. Hence, we summarized the preliminaries of machine learning and computer vision to lay the foundation for the rest of the thesis. This allowed us to then be able to study a Convolutional Neural Network approach that predicts intrinsic images, a decomposition of scenes into their reflectance and shading components, of which the first provides the desired illumination-invariant representation. Since this separation is an ill-posed problem and data are hard to come by, we tackled this by learning from existing sparse human reflectance judgments and presented a direct CNN prediction model that infers a dense result without any post-processing step which therefore results in fast inference. We realized that a context-free per-pixel judgment is sufficient for competitive results, but continued by presenting a novel way of introducing the prior knowledge of sparse reflectance as edge-aware filtering to improve results thereon. This generally applicable framework proved advantageous also for a wide variety of other reflectance predicition algorithms.

Afterwards we looked into another big goal of intelligent systems, namely Augmented Reality (AR). We studied how to relight faces in order to enable telepresence when conferencing with another participant in AR. Having the sender's face being shown in the receiver's augmented reality glasses requires to relight it towards the receiver's scene lighting as spotted through the glasses. The previously used separation into reflectance and shading under the Lambertian assumption alone leads to flat looking faces under strong directional light because all the specularities and cast shadows are missing. Realizing this, we saw the need to create a novel relighting dataset which provides additional intrinsic layers that we subsequently learn to predict. Since synthetic data leads to reduced realism, we described how a capturing setup in the form of a light stage can be built and calibrated so that it later on is able to generate the necessary ground truth intrinsic layers for a diverse set of facial expressions under various lighting conditions. We then provided details of a structured approach that learns to generate relit portraits through a mix of rendering, which utilizes the predicted intrinsic layers of albedo and normals, and additional non-Lambertian ingredients. This pysics-based approach leads to results more consistent with the laws of physics, while still allowing for the power of unstructured residuals to model effects not explainable by a BRDF. As expected, this approach proved to be beneficial over a completely unstructured model or a merely rigid Lambertian rendering.

All together, this work will serve as one more piece in the puzzle towards the big goal of general artificial intelligence [Das et al., 2018].

Appendix A
Decentralized connectivity maintenance

For the sake of completeness, we recap the main features of the connectivity maintenance algorithm presented in Robuffo Giordano et al. [2013] with some changes in the variable names for readability. We start by defining $d_{ij} = \|q_i - q_j\|$ as the distance between two robot positions q_i and q_j, and $d_{ijo} = \min_{\varsigma \in [0,1], o \in \mathcal{O}} \|q_i + \varsigma(q_j - q_i) - o\|$ as the closest distance from the line of sight between robot i and j to any obstacle.

Next, we define a control force f_i^λ, that continuously ensures generalized connectivity of the network of robots. It is called generalized connectivity, because in addition to physical limits of the networking device, also collision avoidance and limits of sensing are modeled within the connectivity framework.

The main conceptual steps behind the computation of f_i^λ can be summarized as follows:

1. Define an auxiliary weighted graph $\mathcal{G}^\lambda(t) = (\mathcal{V}, \mathcal{E}^\lambda, W)$, where W is a symmetric nonnegative $n \times n$ matrix whose entries W_{ij} represent the weight of the edge (i, j) and $(i, j) \in \mathcal{E}^\lambda \Leftrightarrow W_{ij} > 0$.

2. Design every weight W_{ij} as a *smooth* function of the robot positions q_i, q_j and of the obstacle points surrounding q_i and q_j, with the property that $W_{ij} = 0$, meaning the link between robot i and j is broken, if and only if at least one of the following conditions is verified:

   a) the maximum sensing range R_s is reached: $d_{ij} \geq R_s$,

   b) the minimum desired distance to obstacles R_o is reached (where $R_o < R_m$): $d_{ijo} \leq R_o$;

   c) the minimum desired inter-robot distance R_c is reached: $d_{ik} \leq R_c$ for at least one $k \neq i$.

3. Compute f_i^λ as the negative gradient of a potential function $V^\lambda(\lambda_2)$ that grows unbounded when $\lambda_2 \to \lambda_2^{\min}$ from above, where λ_2 is the second smallest eigenvalue of the (symmetric and positive semi-definite) Laplacian matrix

$$L = \mathrm{diag}_{i=1}^{n}\left(\sum_{j=1}^{n} W_{ij}\right) - W, \tag{A.1}$$

and $\lambda_2^{\min}$ is a non-negative parameter. This eigenvalue λ_2 is often also called Fiedler eigenvalue.

It is known from graph theory that a graph is connected if and only if the Fiedler eigenvalue of its Laplacian is positive [Fiedler, 1973]. If $\mathcal{G}^\lambda(0)$ is connected, and in particular $\lambda_2(0) > \lambda_2^{\min}$, then under the action of f_i^λ, the value of $\lambda_2(t)$ can never decrease below $\lambda_2^{\min}$ and therefore $\mathcal{G}^\lambda(t)$ always stays connected.

From a formal point of view the anti-gradient of V^λ for the i-th robot takes the form

$$f_i^\lambda = -\frac{\partial V^\lambda(\lambda_2)}{\partial q_i} = -\frac{dV^\lambda}{d\lambda_2}\frac{\partial \lambda_2}{\partial q_i}. \tag{A.2}$$

Moreover, if the formal expression of V^λ and W are known then Eq. (A.2) can be analytically computed via the expression

$$\frac{\partial \lambda_2}{\partial q_i} = \sum_{j \in \mathcal{N}_i} \frac{\partial W_{ij}}{\partial q_i}\left(v_2^{(i)} - v_2^{(j)}\right)^2 \tag{A.3}$$

(see Yang et al. [2010]), where $v_2^{(i)}$ is the i-th component of the normalized eigenvector of L associated to λ_2.

In order to have a fully decentralized computation of f_i^λ, the robots perform a distributed estimation of both $\lambda_2(t)$ and $v_2^{(i)}(t)$, for all $i = 1,\dots,N$, as shown in Yang et al. [2010]. In Robuffo Giordano et al. [2013] the authors finally prove the passivity (and then the stability) of the system w.r.t. the pair (f_i, v_i) for all $i = 1,\dots,N$, as well as the possibility to compute the connectivity force f_i^λ in Eq. (A.2) in a completely decentralized way.

Appendix B

Supplementary material for Chapters 4 and 5

B.1 Gradients for WHDR-Hinge

For backpropagation through the neural network, we need the gradients of our new WHDR-Hinge loss layer w.r.t. the reflectance layer. By linearity of the (partial) derivative for a given pixel j, we have:

$$\frac{\partial}{\partial R_j^c}\mathrm{WHDR}_\delta(J,R)_{\mathrm{Hinge}} = \frac{\partial}{\partial R_j^c}\frac{\sum_i w_i \ell\left(\frac{R_{i_1}}{R_{i_2}}\right)}{\sum_i w_i} = \frac{\sum_i w_i \cdot \frac{\partial}{\partial R_j^c}\ell\left(\frac{R_{i_1}}{R_{i_2}}\right)}{\sum_i w_i} \tag{B.1}$$

To entangle the derivative for a fixed judgement i, we will write $y(L) := \frac{L_{i_1}}{L_{i_2}}$ and $L(R_j) := \frac{1}{3}\sum_{c=1}^{3} R_j^c$. Hence we have $\ell\left(\frac{R_{i_1}}{R_{i_2}}\right) = \ell\left(y\left(L(R)\right)\right)$ and therefore with the chain rule

$$\frac{\partial}{\partial R_j^c}\ell\left(\frac{R_{i_1}}{R_{i_2}}\right) = \frac{\partial}{\partial R_j}\ell\left(y\left(L(R)\right)\right) = \frac{\partial}{\partial y}\ell\left(y\right)\frac{\partial}{\partial L}y(L)\frac{\partial}{\partial R_j}L(R) \tag{B.2}$$

It is easy to see that

$$\frac{\partial}{\partial L_j}y(L) = \frac{\partial}{\partial L_j}\frac{L_{i_1}}{L_{i_2}} = \begin{cases} \frac{1}{L_{i_2}} & \text{if } j = i_1 \\ -\frac{L_{i_1}}{L_{i_2}^2} & \text{if } j = i_2 \\ 0 & \text{otherwise} \end{cases} \tag{B.3}$$

and

$$\frac{\partial}{\partial R_j}L(R_j) = \frac{\partial}{\partial R_j}\frac{1}{3}\sum_{c=1}^{3} R_j^c = \frac{1}{3}\begin{pmatrix}1\\1\\1\end{pmatrix} \tag{B.4}$$

therefore we have

$$\frac{\partial}{\partial L} y(L) \frac{\partial}{\partial R_j} L(R) \tag{B.5}$$

If we make ℓ dependent on y, we can rewrite:

$$\ell(y) = \begin{cases} \max\left(0,\ y - \frac{1}{1+\delta}\right) & \text{if } J_i = 1 \\ \max\left(0,\ \frac{1}{1+\delta} - y,\ y - (1+\delta)\right) & \text{if } J_i = E \\ \max(0,\ 1+\delta - y) & \text{if } J_i = 2 \end{cases} \tag{B.6}$$

so the (partial) derivatives become:

$$\frac{\partial}{\partial y} \ell(y) = \begin{cases} \begin{cases} 1 & \text{if } y > \frac{1}{1+\delta} \\ 0 & \text{otherwise} \end{cases} & \text{if } J_i = 1 \\ \begin{cases} -1 & \text{if } y < \frac{1}{1+\delta} \\ 1 & \text{if } y > 1+\delta \\ 0 & \text{otherwise} \end{cases} & \text{if } J_i = E \\ \begin{cases} -1 & \text{if } y < 1+\delta \\ 0 & \text{otherwise} \end{cases} & \text{if } J_i = 2 \end{cases} \tag{B.7}$$

Hence, we get the full gradient in (B.1) that uses (B.2) by combining (B.7) and (B.5):

- In the case of $J_i = 1$ this is:

$$\frac{\partial}{\partial R_{i_1}} \max\left(0, \frac{R_{i_1}}{R_{i_2}} - \frac{1}{1+\delta}\right) = \begin{cases} \frac{1}{R_{i_2}} & \text{if } \frac{R_{i_1}}{R_{i_2}} > \frac{1}{1+\delta} \\ 0 & \text{otherwise} \end{cases} \tag{B.8}$$

$$\frac{\partial}{\partial R_{i_2}} \max\left(0, \frac{R_{i_1}}{R_{i_2}} - \frac{1}{1+\delta}\right) = \begin{cases} -\frac{R_{i_1}}{R_{i_2}^2} & \text{if } \frac{R_{i_1}}{R_{i_2}} > \frac{1}{1+\delta} \\ 0 & \text{otherwise} \end{cases} \tag{B.9}$$

$$\forall j \notin \{i_1, i_2\} : \frac{\partial}{\partial R_j} \max\left(0, \frac{R_{i_1}}{R_{i_2}} - \frac{1}{1+\delta}\right) = 0 \tag{B.10}$$

- In the case of $J_i = E$ this is:

$$\frac{\partial}{\partial R_{i_1}} \max\left(0, 1+\delta - \frac{R_{i_1}}{R_{i_2}}\right) = \begin{cases} -\frac{1}{R_{i_2}} & \text{if } \frac{R_{i_1}}{R_{i_2}} < \frac{1}{1+\delta} \\ \frac{1}{R_{i_2}} & \text{if } \frac{R_{i_1}}{R_{i_2}} > 1+\delta \\ 0 & \text{otherwise} \end{cases} \tag{B.11}$$

$$\frac{\partial}{\partial R_{i_2}} \max\left(0, 1+\delta - \frac{R_{i_1}}{R_{i_2}}\right) = \begin{cases} \frac{R_{i_1}}{R_{i_2}^2} & \text{if } \frac{R_{i_1}}{R_{i_2}} < \frac{1}{1+\delta} \\ -\frac{R_{i_1}}{R_{i_2}^2} & \text{if } \frac{R_{i_1}}{R_{i_2}} > 1+\delta \\ 0 & \text{otherwise} \end{cases} \tag{B.12}$$

$$\forall j \notin \{i_1, i_2\} : \frac{\partial}{\partial R_j} \max\left(0, 1+\delta - \frac{R_{i_1}}{R_{i_2}}\right) = 0 \tag{B.13}$$

- In the case of $J_i = 2$ this is:

$$\frac{\partial}{\partial R_{i_1}} \max\left(0, 1+\delta - \frac{R_{i_1}}{R_{i_2}}\right) = \begin{cases} -\frac{1}{R_{i_2}} & \text{if } \frac{R_{i_1}}{R_{i_2}} < 1+\delta \\ 0 & \text{otherwise} \end{cases} \tag{B.14}$$

$$\frac{\partial}{\partial R_{i_2}} \max\left(0, 1+\delta - \frac{R_{i_1}}{R_{i_2}}\right) = \begin{cases} \frac{R_{i_1}}{R_{i_2}^2} & \text{if } \frac{R_{i_1}}{R_{i_2}} < 1+\delta \\ 0 & \text{otherwise} \end{cases} \tag{B.15}$$

$$\forall j \notin \{i_1, i_2\} : \frac{\partial}{\partial R_j} \max\left(0, 1+\delta - \frac{R_{i_1}}{R_{i_2}}\right) = 0 \tag{B.16}$$

In addition we need the derivatives for each RGB component: If we use $R = \frac{1}{3}(r+g+b)$, then we have $\frac{\partial R}{\partial r} = \begin{pmatrix} \frac{1}{3} \\ 0 \\ 0 \end{pmatrix}$, $\frac{\partial R}{\partial g} = \begin{pmatrix} 0 \\ \frac{1}{3} \\ 0 \end{pmatrix}$, $\frac{\partial R}{\partial b} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{3} \end{pmatrix}$.

B.2 Extended qualitative results

To better assess the qualitative performance of our approach in comparison to related work, a collection of results is compiled in Figs. B.1 to B.3. The images are randomly sampled from the intersection of the Narihira et al. [2015b] and Zoran et al. [2015] test split. The 'flat' image used for guidance in filtering (Sec. 5.2) is given in the first row each.

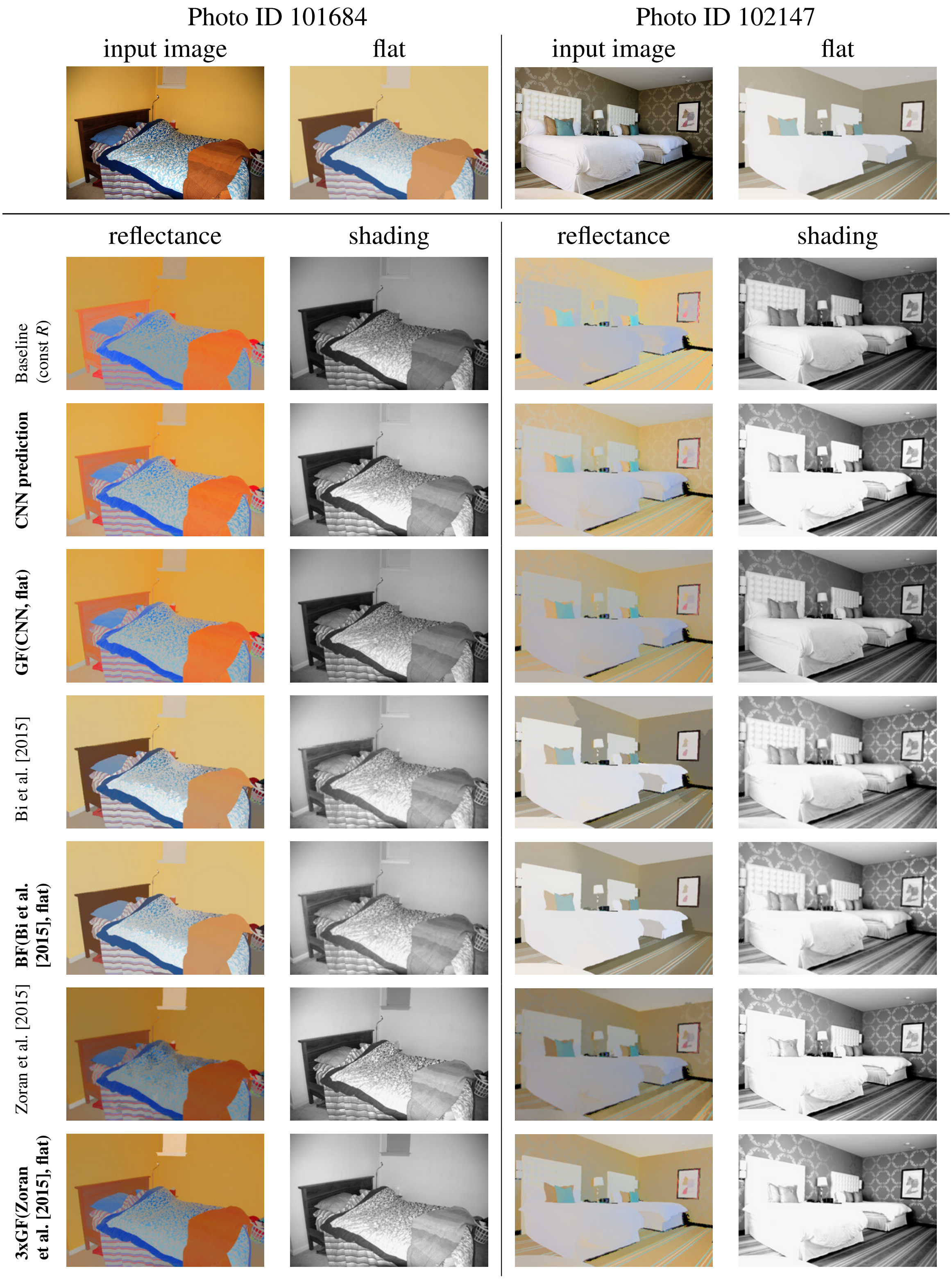


Figure B.1: **Qualitative comparison on sample images of IIW.** The first row gives the input image and the 'flat' image (see Sec. 5.2) used for filtering. In the following rows the decompositions into reflectance in the first column and shading in the second of several methods is shown. All outputs are mapped to sRGB for display.

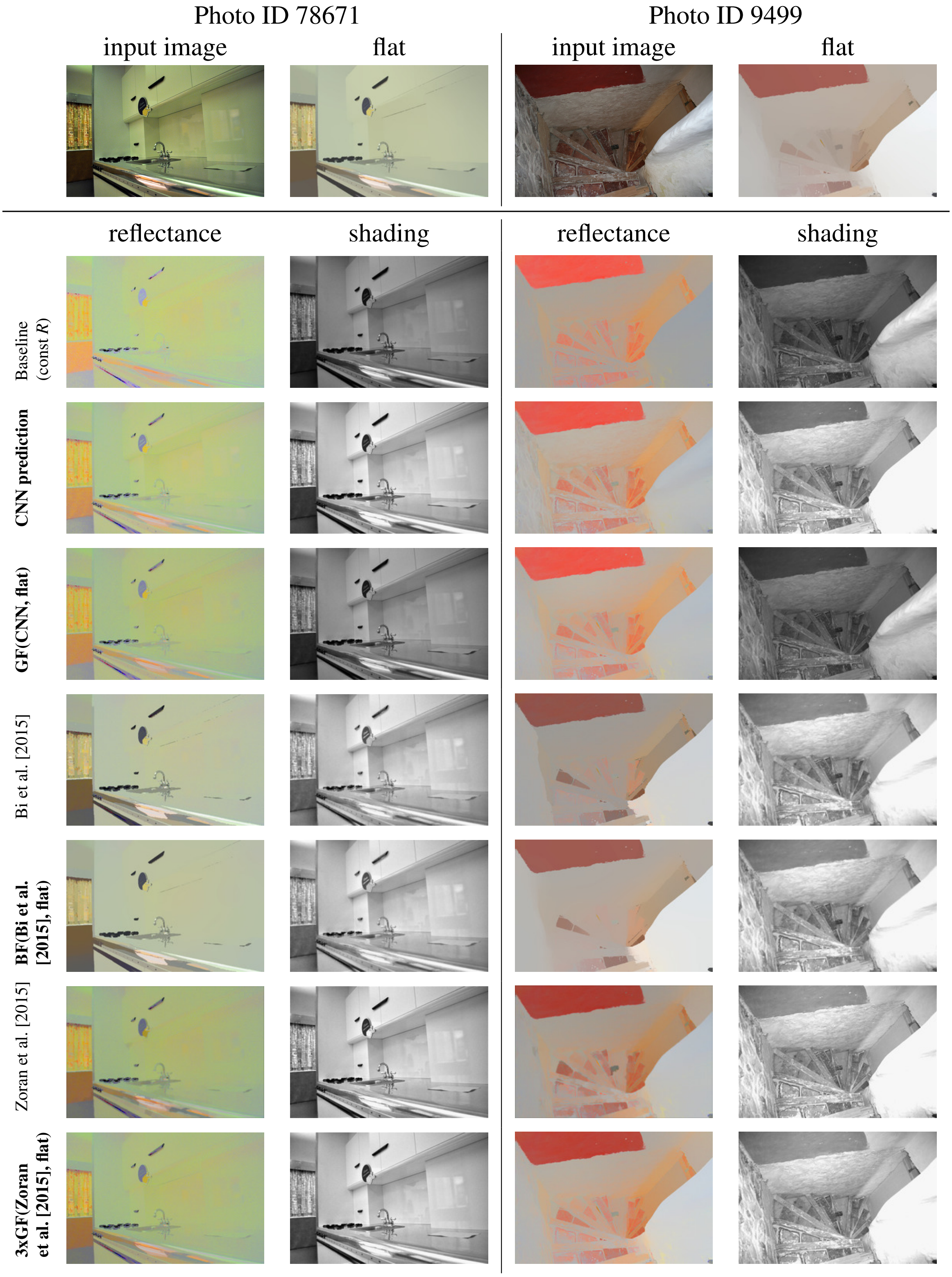


Figure B.2: Extends Fig. B.1 with Photo IDs 78671 and 9499.

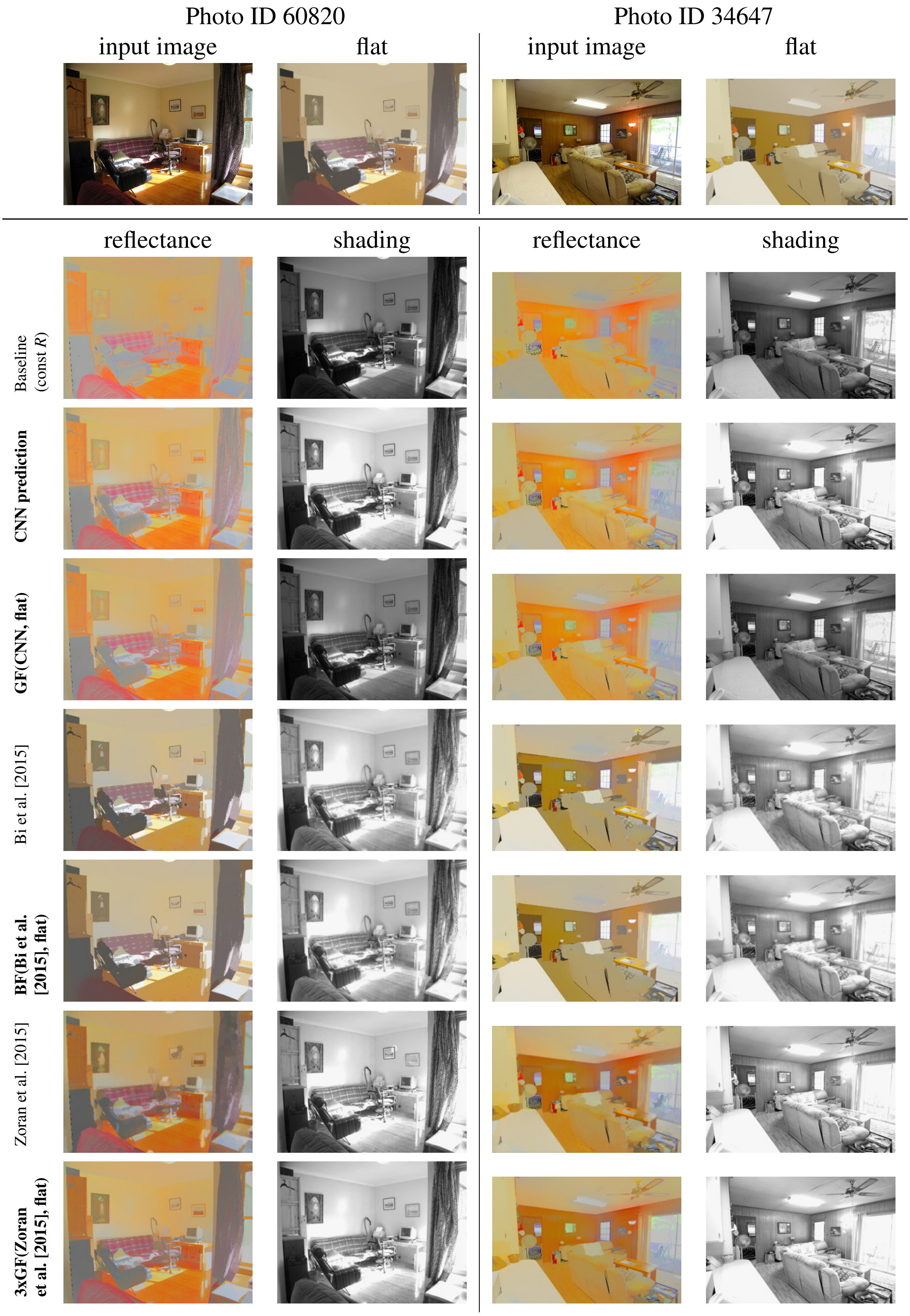


Figure B.3: Extends Fig. B.1 with Photo IDs 60820 and 34647.

Appendix C

Supplementary material for Chapter 7

C.1 Extended results for Relighting in the wild

For relighting in the wild (see Sec. 7.6.3), we show more face relightings in Fig. C.1, this time of images taken with a Canon EOS 6D, again outside of our capture environment. All images were taken in an office with relatively diffuse lighting. Compared to Fig. 7.11, no additional back-transformation to the original light color distribution was applied.

C.2 Multiple input light sources

Our experiments with environment maps show that the extension to multiple output lights is straightforward. Although the case of multiple input lights is less obvious, our model is able to generate meaningful results even when the input image was lit under complex lighting. Indeed, we have already shown results of this type when we discussed relighting in the wild, which does not put any constraints on the source lighting. In order to explore the performance in this regime more systematically, we additionally conducted controlled experiments by using our light stage setup to synthesize multi-illumination images. Since light is additive, we can simply combine captures under lights from different directions to artificially create inputs that were lit by multiple lights. Providing these as input to our model, which violates our model assumption, we still get meaningful results, as can be seen in Fig. C.2. Specifically, we sampled 3 input light directions at random to create the input image, then ran our model 3 times, each time providing the combined input image and one of the 3 source light directions. Finally, we compose the corresponding output images.

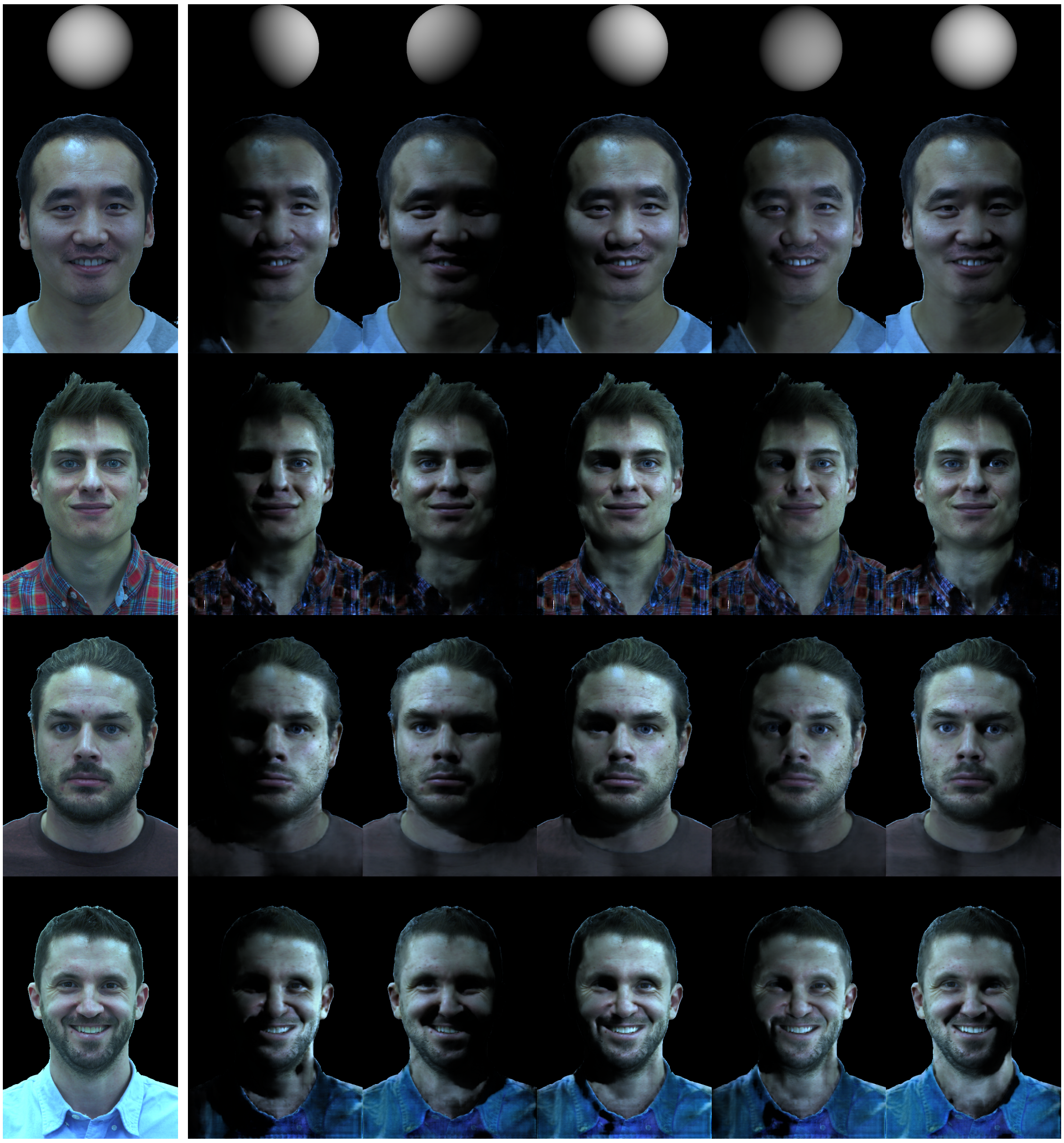


Figure C.1: **Relighting in the wild.** We consider portraits not taken in our capture environment (first column) and relight them with respect to 5 different target point lights. Point light directions are visualized by rendered spheres at the top.

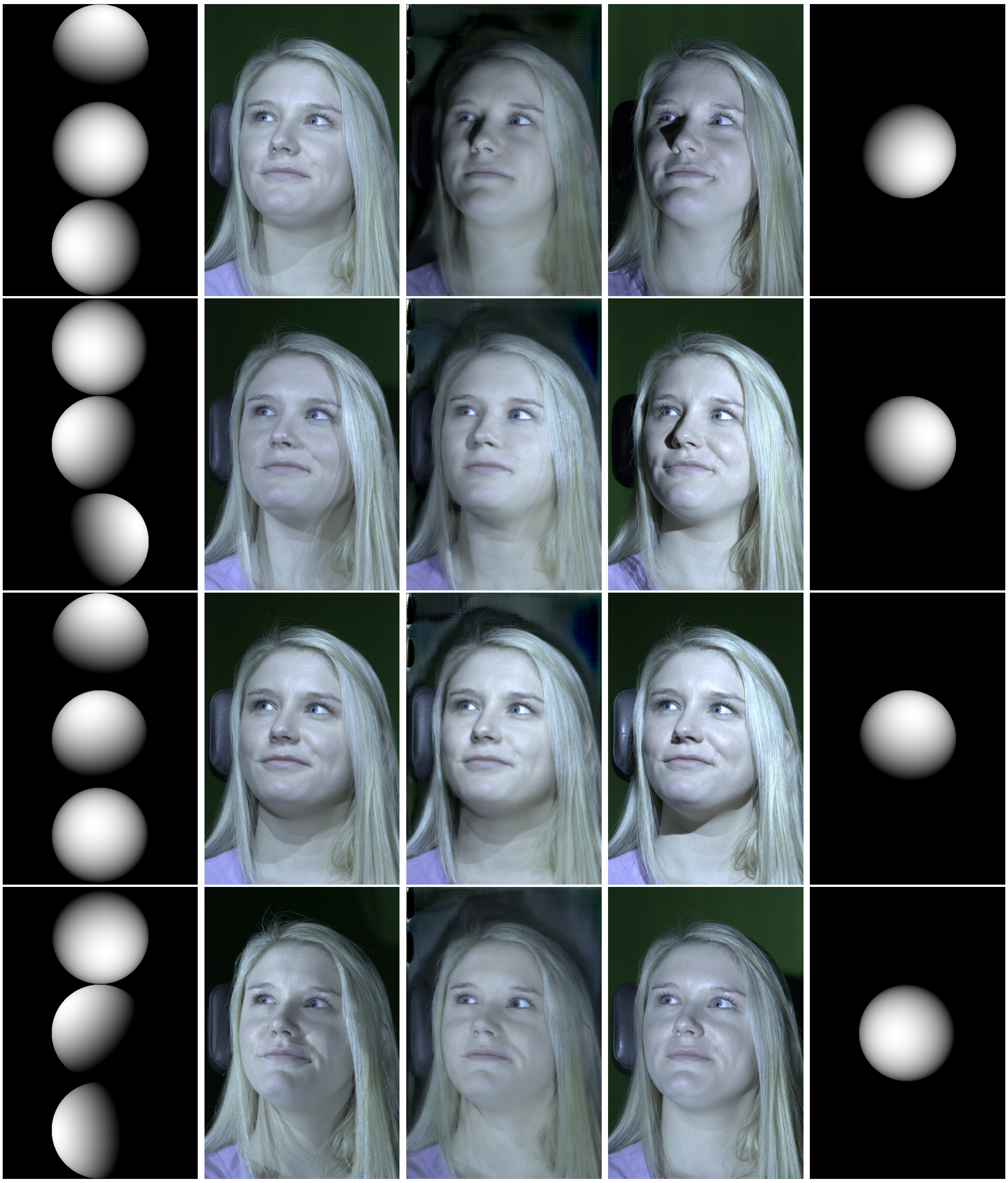

(a) input light (b) input image (c) ours (d) GT (e) output light

Figure C.2: **Multiple input light sources.** Using multiple light sources (**a**), we construct the input image (**b**) and relight it with our model (**c**) towards the desired ground truth (GT) (**d**) under the output light direction (**e**).

Bibliography

Edward H Adelson. Lightness perception and lightness illusions. *New Cogn. Neurosci*, 339, 2000. → Cited on pages 80, 89.

Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 605–612. ACM, 2003. → Cited on page 130.

Gwon Hwan An, Siyeong Lee, Min-Woo Seo, Kugjin Yun, Won-Sik Cheong, and Suk-Ju Kang. Charuco board-based omnidirectional camera calibration method. *Electronics*, 7(12):421, 2018. → Cited on page 106.

G. Antonelli, F. Arrichiello, S. Chiaverini, and R. Setola. A self-configuring MANET for coverage area adaptation through kinematic control of a platoon of mobile robots. In *2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1332–1337, Edmonton, Canada, Aug. 2005. → Cited on pages 28, 29, 33.

G. Antonelli, F. Arrichiello, S. Chiaverini, and R. Setola. Coordinated control of mobile antennas for ad-hoc networks in cluttered environments. In *9th Int. Conf. on Intelligent Autonomous Systems*, Tokyo, Japan, Mar. 2006. → Cited on page 29.

Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015. → Cited on pages 81, 83, 113, 114, 115.

Harry G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer vision systems, A Hanson & E. Riseman (Eds.)*, pages 3–26, 1978. → Cited on pages 69, 79, 80, 105, 113, 115.

Shida Beigpour and Joost Van de Weijer. Object recoloring based on intrinsic image estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 327–334. IEEE, 2011. → Cited on page 65.

Shida Beigpour, Marc Serra, Joost van de Weijer, Robert Benavente, María Vanrell, Olivier Penacchio, and Dimitris Samaras. Intrinsic image evaluation on synthetic complex scenes. In *ICIP*, pages 285–289, 2013. → Cited on page 80.

Shida Beigpour, Andreas Kolb, and Sven Kunz. A comprehensive multi-illuminant dataset for benchmarking of the intrinsic image algorithms. In *Proceedings of the*

IEEE International Conference on Computer Vision, pages 172–180, 2015. → Cited on page 80.

T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, 2006. → Cited on page 64.

Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):159, 2014. → Cited on pages 79, 80, 81, 82, 83, 84, 89, 90, 92, 93, 101, 102, 115.

Sai Bi, Xiaoguang Han, and Yizhou Yu. An L1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Trans. Graph.*, 34(4):78:1–78:12, July 2015. ISSN 0730-0301. doi: 10.1145/2766946. URL `http://doi.acm.org/10.1145/2766946`. → Cited on pages 81, 83, 90, 91, 92, 96, 97, 98, 101, 103, 144, 145, 146.

L. Biagiotti and C. Melchiorri. *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008. ISBN 978-3540856283. → Cited on pages 36, 38, 45.

Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter N Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. *ACM Transactions on Graphics (SIGGRAPH 2008)*, 27(3):39:1–39:8, 2008. → Cited on page 116.

Blender. Blender - a 3d modelling and rendering package. `http://www.blender.org/`, 2021. Accessed: 2021-06-07. → Cited on page 80.

Nicolas Bonneel, Kalyan Sunkavalli, James Tompkin, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. Interactive Intrinsic Video Editing. *ACM Transactions on Graphics (SIGGRAPH Asia 2014)*, 33(6), 2014. → Cited on pages 65, 82, 83.

Adrien Bousseau, Sylvain Paris, and Frédo Durand. User-assisted intrinsic images. In *ACM Transactions on Graphics (TOG)*, volume 28, page 130. ACM, 2009. → Cited on page 82.

W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Trans. on Robotics and Automation*, 21(3):376–386, 2005. → Cited on page 33.

Brent Burley. Physically-based shading at disney. In *SIGGRAPH 2012 Courses*, 2012. → Cited on page 118.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012*, pages 611–625. Springer, 2012. → Cited on pages 80, 83.

Dan A Calian, Jean-François Lalonde, Paulo Gotardo, Tomas Simon, Iain Matthews, and Kenny Mitchell. From faces to outdoor light probes. In *Computer Graphics Forum*, volume 37, pages 51–61. Wiley Online Library, 2018. → Cited on page 116.

Qifeng Chen and Vladlen Koltun. A simple model for intrinsic image decomposition with depth cues. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 241–248. IEEE, 2013. → Cited on page 82.

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Neural Information Processing Systems (NIPS)*, 2016. → Cited on page 82.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. → Cited on page 76.

David Corney, John-Dylan Haynes, Geraint Rees, and R Beau Lotto. The brightness of colour. *PloS one*, 4(3):e5091, 2009. → Cited on page 92.

Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. *ACM Transactions on Graphics (SIGGRAPH Asia 2011)*, 30(6), December 2011. → Cited on page 116.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2054–2063, 2018. → Cited on page 138.

Edsger W Dijkstra. The threats to computing science. In *ACM 1984 South Central Regional Conference, Austin, TX, Nov*, pages 16–18, 1984. URL `https://www.cs.utexas.edu/users/EWD/ewd08xx/EWD898.PDF`. → Cited on page 20.

Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. Multiview Intrinsic Images of Outdoors Scenes with an Application to Relighting. *ACM Transactions on Graphics (TOG)*, 34(5):1–16, 2015. → Cited on page 116.

J. W. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion without global localization via local frontiers. *Autonomous Robots*, 32(1):81–95, 2012. → Cited on page 33.

David Eberly. Computing a point of reflection on a sphere. Technical report, Geometric Tools, Redmond WA 98052, Feb 2008. → Cited on page 106.

J. Faigl and G. A. Hollinger. Unifying multi-goal path planning for autonomous data collection. In *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2937–2942, Chicago, IL, Sep. 2014. → Cited on page 28.

Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. Revisiting deep intrinsic image decompositions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8944–8952, 2018. → Cited on pages 102, 103, 115.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23 (98):298–305, 1973. → Cited on pages 32, 140.

A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Trans. on Mechatronics*, 14 (2):163–175, 2009. → Cited on pages 28, 29, 33.

R. A. Freeman, P. Yang, and K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *45th IEEE Conf. on Decision and Control*, pages 338–343, San Diego, CA, Jan. 2006. → Cited on page 49.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. → Cited on page 77.

Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. → Cited on page 77.

Elena Garces, Adolfo Munoz, Jorge Lopez-Moreno, and Diego Gutierrez. Intrinsic images by clustering. In *Computer Graphics Forum*, volume 31, pages 1415–1424. Wiley Online Library, 2012. → Cited on pages 83, 89.

Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090*, 2017. → Cited on page 116.

Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. → Cited on page 106.

Peter Vincent Gehler, Carsten Rother, Martin Kiefel, Lumin Zhang, and Bernhard Schölkopf. Recovering intrinsic images with a global sparsity prior on reflectance. In *Advances in Neural Information Processing Systems (NIPS)*, pages 765–773, 2011. → Cited on pages 81, 84, 95.

Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Efstratios Gavves, Mario Fritz, Luc Van Gool, and Tinne Tuytelaars. Reflectance and natural illumination from single-material specular objects using deep learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1932–1947, 2018. → Cited on page 115.

Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2010. → Cited on page 106.

Dennis H Goldstein, David B Chenault, and J Larry Pezzaniti. Polarimetric characterization of spectralon. In *Polarization: Measurement, Analysis, and Remote Sensing II*, volume 3754, pages 126–137. International Society for Optics and Photonics, 1999. → Cited on page 72.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. → Cited on page 121.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. → Cited on page 73.

V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi. The TeleKyb framework for a modular and extendible ROS-based quadrotor control. In *6th European Conference on Mobile Robots*, pages 19–25, Barcelona, Spain, Sep. 2013. → Cited on page 52.

Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2335–2342. IEEE, 2009. → Cited on pages 80, 81, 83.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018. → Cited on pages 76, 77.

Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In *The European Conference on Computer Vision (ECCV)*, September 2018. → Cited on page 77.

Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *Computer Vision–ECCV 2010*, pages 1–14. Springer, 2010. → Cited on page 96.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015. → Cited on pages 76, 89.

Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2017. → Cited on page 116.

G. Hollinger and S. Singh. Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE Trans. on Robotics*, 28(4):967–973, 2012. → Cited on page 29.

Berthold Klaus Paul Horn. *Robot vision.* MIT press, 1986. → Cited on pages 69, 70, 71, 72.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. → Cited on page 74.

A. Howard, L. E. Parker, and G. S. Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6):431–447, 2006. → Cited on page 28.

Robert William Gainer Hunt and Michael R Pointer. *Measuring colour.* John Wiley & Sons, 1987. → Cited on page 70.

Katsushi Ikeuchi. *Computer vision: A reference guide*. Springer Publishing Company, Incorporated, 2014. ISBN 978-0-387-31439-6. → Cited on page 72.

David S Immel, Michael F Cohen, and Donald P Greenberg. A radiosity method for non-diffuse environments. In *ACM Siggraph Computer Graphics*, volume 20, pages 133–142. ACM, 1986. → Cited on page 71.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. → Cited on pages 114, 116, 121, 123.

Varun Jampani, Martin Kiefel, and Peter V Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4452–4461, 2016. → Cited on page 77.

P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Trans. on Robotics and Automation*, 17(5):748–760, 2001. → Cited on page 33.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. → Cited on page 87.

James T. Kajiya. The rendering equation. In *Proc. ACM SIGGRAPH*, pages 143–150, 1986. doi: 10.1145/15922.15902. URL `http://doi.acm.org/10.1145/15922.15902`. → Cited on pages 71, 114, 117.

Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ACM Transactions on Graphics*, (3):32:1–32:15, 2014. → Cited on page 116.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, volume 3, 2015. URL `http://arxiv.org/abs/1412.6980`. → Cited on pages 75, 87.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 972–981, 2017. → Cited on page 76.

Sebastian B. Knorr and Daniel Kurz. Real-time illumination estimation from faces for coherent rendering. In *IEEE International Symposium on Mixed and Augmented Reality*, 2014. → Cited on page 116.

Balazs Kovacs, Sean Bell, Noah Snavely, and Kavita Bala. Shading annotations in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. → Cited on pages 101, 103, 115.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. → Cited on pages 73, 76.

Pankaj Kumar. Intrinsic image based moving object cast shadow removal in image sequences. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 410–415. IEEE, 2011. → Cited on page 65.

J. Lächele, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. SwarmSimX: Real-time simulation environment for multi-robot systems. In I. Noda, N. Ando, D. Brugali, and J.J. Kuffner, editors, *3rd Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, volume 7628 of *Lecture Notes in Computer Science*, pages 375–387. Springer, 2012. → Cited on page 51.

Pierre-Yves Laffont and Jean-Charles Bazin. Intrinsic decomposition of image sequences from local temporal variations. In *ICCV*, 2015. → Cited on page 82.

Alice Lam. Tacit knowledge, organizational learning and societal institutions: An integrated framework. *Organization studies*, 21(3):487–513, 2000. → Cited on page 73.

Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760. → Cited on page 71.

Edwin H Land and John McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971. → Cited on pages 81, 89, 92, 113.

Ha Le and Ioannis Kakadiaris. Illumination-invariant face recognition with deep relit face images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2146–2155. IEEE, 2019. → Cited on page 116.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999. → Cited on page 77.

Jinho Lee, Raghu Machiraju, Baback Moghaddam, and Hanspeter Pfister. Estimation of 3d faces and illumination from single photographs using a bilinear illumination model. In *Eurographics Conference on Rendering Techniques*, EGSR '05, pages 73–82, 2005. → Cited on page 116.

Andreas Lehrmann and Leonid Sigal. Non-parametric structured output networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4214–4224. Curran Associates, Inc., 2017. → Cited on page 77.

Louis Lettry, Kenneth Vanhoey, and Luc Van Gool. Darn: a deep adversial residual network for intrinsic image decomposition. *arXiv preprint arXiv:1612.07899*, 2016. → Cited on pages 82, 83.

Chen Li, Kun Zhou, and Stephen Lin. Intrinsic face image decomposition with human face priors. In *European Conference on Computer Vision*, pages 218–233, 2014. → Cited on page 116.

Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. → Cited on pages 101, 102, 103.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 37(6), November 2018. → Cited on page 115.

H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*, 24(3-4):353–363, 2001. → Cited on pages 28, 39, 42.

Stephen Lombardi and Ko Nishino. Reflectance and illumination recovery in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):129–141, 2016. → Cited on page 115.

Jundan Luo, Zhaoyang Huang, Yijin Li, Xiaowei Zhou, Guofeng Zhang, and Hujun Bao. NIID-Net: Adapting Surface Normal Knowledge for Intrinsic Image Decomposition in Indoor Scenes. *IEEE Transactions on Visualization and Computer Graphics*, 26 (12):3434–3445, 2020. → Cited on pages 102, 103.

N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997. ISBN 1558603484. → Cited on pages 28, 39.

Ernst Mach. Über die Wirkung der räumlichen Vertheilung des Lichtreizes auf die Netzhaut. *Sitzungsberichte der mathematisch-naturwissenschaftlichen Classe der kaiserlichen Akademie der Wissenschaften*, 52:303–322, 1865. → Cited on page 71.

Will Maddern, Alex Stewart, Colin McManus, Ben Upcroft, Winston Churchill, and Paul Newman. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, 2014. → Cited on page 65.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. The MIT Press, 1982. → Cited on pages 65, 72.

C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Interactive planning of persistent trajectories for human-assisted navigation of mobile robots. In *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2641–2648, Vilamoura, Portugal, Oct. 2012. → Cited on page 39.

Wojciech Matusik. *A data-driven reflectance model*. PhD thesis, Massachusetts Institute of Technology, 2003. → Cited on page 117.

Abhimitra Meka, Maxim Maximov, Michael Zollhoefer, Avishek Chatterjee, Hans-Peter Seidel, Christian Richardt, and Christian Theobalt. LIME: Live intrinsic material estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. → Cited on page 115.

Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhoefer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, Peter Denny, Sofien Bouaziz, Peter Lincoln, Matt Whalen, Geoff Harvey, Jonathan Taylor, Shahram Izadi, Andrea Tagliasacchi, Paul Debevec, Christian Theobalt, Julien Valentin, and Christoph Rhemann. Deep reflectance fields - high-quality facial reflectance field inference from color gradient illumination. *ACM Transactions on Graphics (SIGGRAPH)*, 38(4), July 2019. → Cited on page 117.

Marvin Minsky, Seymour A Papert, and Léon Bottou. *Perceptrons: An introduction to computational geometry*. MIT press, 1969. → Cited on page 73.

V. Mistler, A. Benallegue, and N. K. M'Sirdi. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In *10th IEEE Int. Symp. on Robots and Human Interactive Communications*, pages 586–593, Bordeaux, Paris, France, Sep. 2001. → Cited on page 56.

A. R. Mosteo, L. Montano, and M. G. Lagoudakis. Guaranteed-performance multi-robot routing under limited communication range. In *9th Int. Symp. on Distributed Autonomous Robotic Systems*, pages 491–502, Tsukuba, Japan, Nov. 2008. → Cited on page 30.

Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. A dataset of multi-illumination images in the wild. In *IEEE International Conference on Computer Vision*, 2019. → Cited on page 116.

R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. Erkmen. Search and rescue robotics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1151–1173. Springer, 2008. → Cited on page 28.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. → Cited on page 76.

Takuya Narihira, Michael Maire, and Stella X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *International Conference on Computer Vision (ICCV)*, 2015a. → Cited on pages 80, 82, 83.

Takuya Narihira, Michael Maire, and Stella X Yu. Learning lightness from human judgement on relative reflectance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2973, 2015b. → Cited on pages 80, 81, 82, 83, 85, 86, 87, 89, 98, 99, 101, 143.

Thomas Nestmeyer. Decentralized multi-target exploration and connectivity maintenance with a multi-robot system. Master's thesis, Eberhard Karls University Tübingen, 10 2012. → Cited on page 27.

Thomas Nestmeyer and Peter V Gehler. Reflectance adaptive filtering improves intrinsic image estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6789–6798, 2017. → Cited on pages 79, 95, 97, 101, 102, 103.

Thomas Nestmeyer, Martin Riedel, Johannes Lächele, Simon Hartmann, Fiete Botschen, Paolo Robuffo Giordano, and A. Franchi. Interactive demo: Haptic remote control of multiple uavs with autonomous cohesive behavior. In *Int. Work. on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration, at 2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013a. → Cited on page 58.

Thomas Nestmeyer, Paolo Robuffo Giordano, and Antonio Franchi. Multi-target simultaneous exploration with continual connectivity. In *2nd Int. Work. on Crossing the Reality Gap - From Single to Multi- to Many Robot Systems, at 2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013b. → Cited on page 27.

Thomas Nestmeyer, Paolo Robuffo Giordano, and Antonio Franchi. Human-assisted parallel multi-target visiting in a connected topology. In *6th Int. Work. on Human-Friendly Robotics*, Rome, Italy, Oct. 2013c. → Cited on page 27.

Thomas Nestmeyer, Paolo Robuffo Giordano, Heinrich H Bülthoff, and Antonio Franchi. Decentralized simultaneous multi-target exploration using a connected network of multiple robots. *Autonomous Robots*, 41(4):989–1011, 2017. → Cited on page 27.

Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, and Andreas Lehrmann. Learning physics-guided face relighting under directional light. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5124–5133, 2020. → Cited on pages 105, 113.

Ko Nishino and Shree K Nayar. Eyes for relighting. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3):704–711, July 2004. → Cited on page 116.

John of Salisbury. *Metalogicon*. 1159. → Cited on page 20.

R. Olfati-Saber and R. M. Murray. Consensus protocols for networks of dynamic agents. In *2003 American Control Conference*, pages 951–956, Denver, CO, Jun. 2003. → Cited on page 48.

R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, 49(9):1520–1533, 2004. → Cited on page 28.

I. Omer and M. Werman. Color lines: Image specific color representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. → Cited on pages 81, 95.

F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Trans. on Robotics*, 28(3): 592–606, 2012. → Cited on pages 28, 29, 33.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. → Cited on page 121.

Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. In *ACM Transactions on Graphics (TOG)*, volume 26, page 52. ACM, 2007. → Cited on page 113.

Y. Pei and M. W. Mutka. Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration. In *2012 IEEE Int. Conf. on Robotics and Automation*, pages 1551–1556, St. Paul, MN, May 2012. → Cited on pages 28, 29.

Y. Pei, M. W. Mutka, and N. Xi. Coordinated multi-robot real-time exploration with connectivity and bandwidth awareness. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 5460–5465, Anchorage, AK, May 2010. → Cited on page 29.

Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM transactions on graphics (TOG)*, 23(3):664–672, 2004. → Cited on page 96.

Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A Efros, and George Drettakis. Multi-view Relighting Using a Geometry-aware Network. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. → Cited on page 116.

Michael Polanyi. *The Tacit Dimension*. London: Routledge and Kegan Paul, 1966. → Cited on page 72.

Charles Poynton. *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012. → Cited on page 92.

Yanlin Qian, Miaojing Shi, Joni-Kristian Kämäräinen, and Jiri Matas. Fast fourier intrinsic network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3169–3178, January 2021. → Cited on pages 102, 103.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. doi: 10.1162/neco_a_00990. PMID: 28599112. → Cited on pages 76, 77.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM Trans. Graph.*, 34(4):111:1–111:12, July 2015. ISSN 0730-0301. doi: 10.1145/2766899. → Cited on page 116.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. → Cited on page 74.

P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. A passivity-based decentralized strategy for generalized connectivity maintenance. *The International Journal of Robotics Research*, 32(3):299–323, 2013. → Cited on pages 27, 28, 30, 31, 32, 34, 38, 139, 140.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. → Cited on pages 88, 121.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. → Cited on pages 73, 75.

SB Sells and Richard S Fixott. Evaluation of research on effects of visual training on visual functions. *American journal of ophthalmology*, 44(2):230–236, 1957. → Cited on page 21.

Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *Computer Vision and Pattern Regognition (CVPR)*, 2018. → Cited on pages 113, 114, 115, 116, 123.

Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W. Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *International Conference on Computer Vision*, 2019. → Cited on pages 116, 118.

Davoud Shahlaei and Volker Blanz. Realistic inverse lighting from a single 2d image of a face, taken under unknown and complex lighting. In *IEEE International Conference on Automatic Face and Gesture Recognition*, jul 2015. → Cited on page 116.

Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *International Conference on Machine Learning*, pages 2217–2225, 2016. → Cited on page 76.

Li Shen and Chuohao Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 697–704. IEEE, 2011. → Cited on pages 81, 83, 89, 95.

Hyunjung Shim. Faces as light probes for relighting. *Optical Engineering*, 51(7): 077002–1, 2012. → Cited on page 116.

Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5444–5453. IEEE, 2017. → Cited on pages 115, 116.

Zhixin Shu, Sunil Hadap, Eli Shechtman, Kalyan Sunkavalli, Sylvain Paris, and Dimitris Samaras. Portrait lighting transfer using a mass transport approach. *ACM Transactions on Graphics (TOG)*, 37(1):2, 2018. → Cited on pages 117, 126, 127.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. → Cited on page 20.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017. → Cited on page 20.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. → Cited on page 20.

E. Stump, A. Jadbabaie, and V. Kumar. Connectivity management in mobile robot teams. In *2008 IEEE Int. Conf. on Robotics and Automation*, pages 1525–1530, Pasadena, CA, May 2008. → Cited on pages 28, 29.

E. Stump, N. Michael, V. Kumar, and V. Isler. Visibility-based deployment of robot formations for communication maintenance. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 4489–4505, Shanghai, China, May. 2011. → Cited on pages 28, 29.

Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics (SIGGRAPH)*, 38(4):79, 2019. → Cited on pages 117, 127.

S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S.M. Seitz. Total moving face reconstruction. In *ECCV*, 2014. → Cited on page 116.

D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel, and L. Montano. Enforcing network connectivity in robot team missions. *The International Journal of Robotics Research*, 29(4):460–480, 2010. → Cited on page 30.

Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998. → Cited on page 96.

Alan Mathison Turing. I.—computing machinery and intelligence. *Mind*, LIX(236): 433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL `https://doi.org/10.1093/mind/LIX.236.433`. → Cited on page 20.

Han LJ Van der Maas, Kees-Jan Kan, and Denny Borsboom. Intelligence is what the intelligence test measures. seriously. *Journal of Intelligence*, 2(1):12–15, 2014. → Cited on page 19.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1993. → Cited on page 86.

Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics (TOG)*, 37(4):64, 2018. → Cited on pages 123, 133.

Yang Wang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and D Samaras. Face re-lighting from a single image under harsh lighting conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. → Cited on page 116.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. → Cited on page 121.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. → Cited on page 121.

Yair Weiss. Deriving intrinsic images from image sequences. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 68–75. IEEE, 2001. → Cited on page 82.

Zhen Wen, Zicheng Liu, and T S Huang. Face relighting with radiance environment maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003. → Cited on page 116.

A. D. Wissner-Gross and C. E. Freer. Causal entropic forces. *Phys. Rev. Lett.*, 110: 168702, Apr 2013. doi: 10.1103/PhysRevLett.110.168702. URL `https://link.aps.org/doi/10.1103/PhysRevLett.110.168702`. → Cited on page 20.

Ying Xiong, Ayan Chakrabarti, Ronen Basri, Steven J. Gortler, David W. Jacobs, and Todd Zickler. From shading to local shape. In *PAMI*, 2014. → Cited on pages 111, 112, 118, 119, 122.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. → Cited on page 76.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):126, 2018. → Cited on page 116.

P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2):390–396, 2010. → Cited on page 140.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. → Cited on page 89.

Ye Yu and William AP Smith. InverseRenderNet: Learning Single Image Inverse Rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. → Cited on page 116.

M. M. Zavlanos and G. J. Pappas. Potential fields for maintaining connectivity of mobile networks. *IEEE Trans. on Robotics*, 23(4):812–816, 2007. → Cited on page 30.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. → Cited on page 75.

Edward Zhang, Michael F. Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics*, 35(6), 2016. → Cited on page 116.

Edward Zhang, Michael F Cohen, and Brian Curless. Discovering point lights with intensity distance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6635–6643, 2018a. → Cited on page 116.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018b. → Cited on pages 121, 122, 126.

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999. → Cited on page 115.

Qi Zhao, Ping Tan, Qiang Dai, Li Shen, Enhua Wu, and Stephen Lin. A closed-form solution to retinex with nonlocal texture constraints. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1437–1444, 2012. → Cited on pages 83, 89.

Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *IEEE International Conference on Computer Vision*, 2019a. → Cited on pages 116, 127.

Hao Zhou, Xiang Yu, and David W. Jacobs. Glosh: Global-local spherical harmonics for intrinsic image decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019b. → Cited on pages 102, 103.

Tinghui Zhou, Philipp Krähenbühl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *ICCV*, 2015. → Cited on pages 80, 81, 82, 83, 85, 87, 89, 92, 101.

Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. Learning ordinal relationships for mid-level vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 388–396, 2015. → Cited on pages 80, 81, 82, 83, 85, 90, 91, 92, 98, 99, 100, 101, 103, 143, 144, 145, 146.