Techniques for depth acquisition and enhancement of depth perception

Liao, Jingtang

DOI
10.4233/uuid:e6ab025b-6acf-4d4c-a583-ad1e39a8caa7

Publication date
2017

Document Version
Final published version

Citation (APA)
Liao, J. (2017). *Techniques for depth acquisition and enhancement of depth perception*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:e6ab025b-6acf-4d4c-a583-ad1e39a8caa7

TECHNIQUES FOR DEPTH ACQUISITION AND ENHANCEMENT OF DEPTH PERCEPTION

TECHNIQUES FOR DEPTH ACQUISITION AND ENHANCEMENT OF DEPTH PERCEPTION

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 12 december 2017 om 15:00 uur

door

Jingtang LIAO

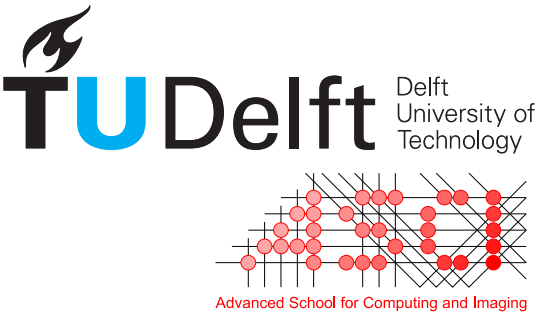
Master of Science in Aerospace Engineering,
Beihang University, Beijing, China,
geboren te Sichuan, China.

Dit proefschrift is goedgekeurd door de

Promotor: Prof. Dr. E. Eisemann

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. Dr. E. Eisemann,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. Dr. A. Hanjalic,	Technische Universiteit Delft
Prof. Dr. S.C. Pont,	Technische Universiteit Delft
Prof. Dr. F.W. Jansen,	Technische Universiteit Delft
Prof. Dr. T. Weyrich,	University College London, UK
Dr. A. Bousseau,	Inria Sophia-Antipolis, France
Dr. J.C. van Gemert,	Technische Universiteit Delft
Overige leden:	
Dr. J.-M Thiery,	Telecom-ParisTech, France



This research is supported by China Scholarship Council (CSC).
This work was carried out in the ASCI graduate school. ASCI dissertation series number: 384.

ISBN 978-94-92516-90-9

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

CONTENTS

SUMMARY

Depth plays an essential role in computer graphics for the sorting of primitives. The related data representation, the depth map, is useful for many different applications. In this dissertation, we present solutions for creating depth maps with the goal of using these maps to enhance the depth perception in the original images. Regarding the generation of depth maps, we propose two solutions, a reconstruction method via near-light photometric stereo (PS) and a depth map design tool via user guidance. Additionally, we present several techniques for image enhancement of depth perception based on depth information. In the following, we give a short summary of the dissertation.

Chapter 2 introduces a solution for reconstructing indoor scenes using near-light PS. This solution overcomes limitations of previous methods, which were restricted to albedo variations, shadowing, perspective projections, or limited in effectiveness by noise. Our method makes use of a video sequence captured of a moving light source in the scene. Additionally, we rely on specular spheres, which are detected via a perspective-correcting Hough transform, to perform a light calibration. We then apply an optimization process to robustly solve the calibrated near-light PS problem. In contrast to previous approaches, our solution reconstructs depth, relative albedo and normals simultaneously and faithfully, and is tested on both synthetic and real-world scenes.

Chapter 3 presents a solution to support a user in generating an approximate depth map for a single image. We show that the resulting map can be used as input for various image depth perception enhancement effects. In this context, the depth maps do not have to be perfect, but should rather support the desired depth-based image enhancement effect. To this end, the depth map is generated in a semi-automatic manner through a diffusion process by integrating user interaction and image features. The user has the freedom to control the depth diffusion process by various tools, such as global depth adjustments and relative depth indications. We demonstrate a variety of 3D effects using the derived depth maps, including wiggle stereoscopy and unsharp masking.

In Chapter 4, we show that the impression of depth ordering can be enhanced by split-depth images, which rely on an optical illusion and have not been much explored so far. By introducing white bars into the scene, which separate fore- and background, image elements appear more distant in depth. We study different factors of this illusion via visual perception experiments and propose a perceptual model to optimize and generate such images automatically. Our method provides practical guidelines to create such images and is verified by a validation study.

In general, we believe that depth maps can become a natural extension to standard image content, but this representation is not widespread yet. Our work makes a step into the direction of filling this gap with automatic and manual solutions. Access to depth information is of great benefit as it gives more flexibility in the depiction of depth as an image cue. We show various techniques to enhance depth perception, which can be chosen depending on preference, context or modality of the display devices.

SAMENVATTING

Diepte speelt een essentiële rol in computergraphics voor het sorteren van geometrische primitieven. De gerelateerde datarepresentatie, de depth map, is nuttig voor vele verschillende applicaties. In dit proefschrift presenteren we oplossingen voor het creëren van depth maps met het doel om deze te gebruiken om perceptie van diepte te verbeteren in de originele afbeeldingen. Aangaande de generatie van depth maps stellen we twee oplossingen voor: een reconstructiemethode die gebruik maakt van near-light photometric stereo (PS) en een depth map ontwerptool middels gebruikersondersteuning. Tevens presenteren we meerdere technieken gebaseerd op diepte-informatie voor een verbeterde perceptie van diepte in afbeeldingen. In wat volgt geven we een korte samenvatting van het proefschrift.

Hoofdstuk 2 introduceert een oplossing voor het reconstrueren van indoorscènes door middel van near-light PS. Deze oplossing komt beperkingen van eerdere methodes te boven, die gelimiteerd waren tot albedovariaties, schaduwen, en perspectiefprojecties, of beperkt werden door ruis. Onze methode maakt gebruik van een videoreeks van een bewegende lichtbron in de scène. Verder benutten we spiegelende bollen, die worden gedetecteerd door een perspectief-corrigerende Hough-transformatie om zo het licht te kalibreren. Vervolgens passen we een optimalisatieproces toe om het gekalibreerde near-light PS-probleem op robuuste wijze op te lossen. In tegenstelling tot eerdere technieken reconstrueert onze oplossing diepte, relatieve albedo en normals tegelijkertijd en waarheidsgetrouw, en is getest op zowel kunstmatige als echte scènes.

Hoofdstuk 3 presenteert een oplossing om gebruikers te ondersteunen bij het genereren van een benaderende depth map voor een enkele afbeelding. We laten zien dat de resulterende depth map gebruikt kan worden als invoer voor meerdere effecten die de perceptie van diepte verbeteren. In deze context hoeven de depth maps niet perfect te zijn, maar moeten eerder de gewenste effecten ondersteunen. Hiertoe wordt de depth map op halfautomatische wijze gegenereerd door middel van een diffusieproces waarbij gebruikersinvoer en afbeeldingskenmerken geïntegreerd worden. De gebruiker heeft de vrijheid om dit proces te controleren middels meerdere tools, zoals globale diepteaanpassingen en relatieve diepte-indicaties. We demonstreren gevarieerde 3D-effecten met behulp van de afgeleide depth maps, waaronder wiggle stereoscopy en unsharp masking.

In hoofdstuk 4 laten we zien dat de impressie van diepterangschikking versterkt kan worden met split-depth afbeeldingen, welke vertrouwen op een optische illusie en nog niet veel onderzocht zijn. Door witte staven in de scène aan te brengen die de voor- en achtergrond scheiden, lijken elementen qua diepte verder uit elkaar te liggen. We bestuderen verschillende factoren van deze illusie middels visueel-perceptuele experimenten en stellen een perceptueel model voor om dergelijke afbeeldingen automatisch te optimaliseren en genereren. Onze methode biedt praktische richtlijnen om zulke afbeeldingen te creëren en wordt geverifieerd door een validatieonderzoek.

Over het algemeen geloven we dat depth maps een natuurlijke extensie van standaardafbeeldingen kunnen worden, maar deze representatie is nog niet wijd verspreid. Ons werk maakt een stap om dit gat te dichten met automatische en handmatige oplossingen. Toegang tot diepte-informatie is van groot voordeel, aangezien het meer flexibiliteit geeft voor het weergeven van diepte als een afbeeldingssignaal. We laten meerdere technieken zien om perceptie van diepte te versterken, die gekozen kunnen worden afhankelijk van voorkeur, context of modaliteit van de weergaveapparaten.

1

INTRODUCTION

In computer graphics, depth information is usually stored in an image, i. e., a depth map, which is analogous to the depth buffer, Z-buffer, Z-buffering and Z-depth. This dissertation presents novel techniques involving this representation, including scene depth reconstruction and enhancement of depth perception. In this chapter, we will discuss the motivation and contributions of our work.

1.1. MOTIVATION

Depth buffers appear in a variety of applications, and over the past decades, we have witnessed its impact in various fields. A few examples include:

- **Object digitization** : the digital content creation has gained much attention and has shown its wide usage in applications, including digital preservation in cultural heritage [WLGK16], physical simulation (e. g., flood simulation [LKT*15], scene lighting design [SLE17]), and virtual reality and augmented reality [KYS03].To produce a digital representation of a real-world object, a usual way is to rely on a set of photos and vision algorithms to derive a depth value per pixel.
- **Image segmentation** : many tasks such as face detection, content-based image retrieval, and foreground and background detection are associated with image segmentation. The use of a color image together with its depth map [RBF12] provides additional cues with respect to the scene geometry. It is beneficial to handle problems such as color camouflage and leads to a more accurate segmentation result.
- **Robotics** : applications like robot navigation and object tracking rely on depth maps to judge an object's distance from the camera system. One solution is to use stereo cameras [ML00], in which a depth map is derived from stereo images.

A more recent development is the use of depth information for image-based effects, such as deriving a stereo image pair from a single input [ZCW*15], depth of field [LES09], haze, or ambient occlusion[BSD08]. These effects are very successful in improving the

1

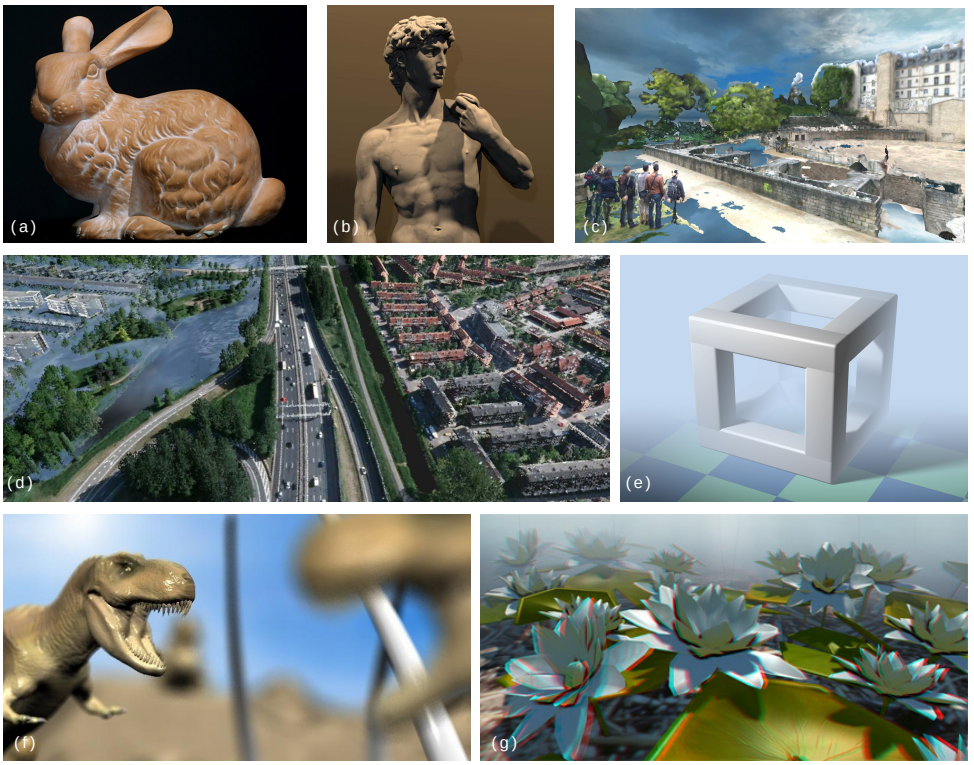


Figure 1.1: Examples of depth related applications. Object digitization: (a) Stanford Bunny, (b) Michelangelo's David, (c) Reconstructed park model, (d) Flood simulation; Depth-based effects: (e) haze, (f) depth of field [LES09], (g) stereo [DRE*11].

appearance of images, offer stylization possibilities, and help in understanding the scene layout by enhancing depth perception. The benefit of such depth-based effects has also been shown for photos[LCD06], but it requires an underlying depth map. Depth perception and preference varies on an individual basis and depends on the used display devices or viewing distance. Being able to add cues in a postprocess, makes it possible to achieve a personalized result. Having this additional flexibility to adapt content as needed, is likely to become a future trend [CED*16].

In this dissertation, we will address the problem of deriving a depth map in two ways, a reconstruction method via near-light photometric stereo and a depth-design method. When using depth-based image manipulation techniques, as outlined above, our depth-design solution provides a possibility to control the outcome. We will present several techniques to enhance depth perception in the original images by relying on an underlying depth map.

1.2. OVERVIEW OF THE DISSERTATION

Our main contributions will be described in Chapter 2-4. We give a brief overview of these main contributions below. The dissertation is rounded off by a conclusion in Chapter 5, which includes a discussion of future work.

NEAR-LIGHT PHOTOMETRIC STEREO

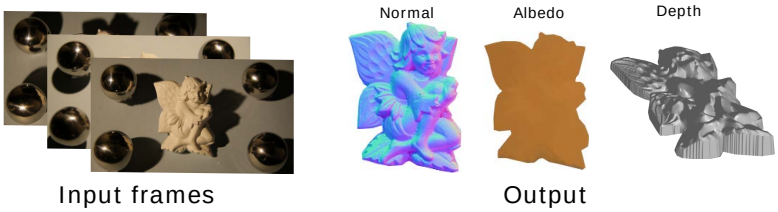


Figure 1.2: 3D reconstruction using our near-light PS approach. Our method can reconstruct indoor scene parameters including normal, albedo and depth simultaneously.

There exists a large body of literature on accurate depth acquisition for real scenes, which can be classified into two main categories: active and passive methods. Active methods usually utilize active 3D scanners. Kinect, LiDAR and time-of-flight cameras are examples. Each scanner comes with its own advantages and limitations. For example, the distance measurement accuracy of time-of-flight cameras is relatively low due to the high speed of light. Most depth sensors tend to be relatively noisy and more accuracy translates into significant costs of the devices, which are not always affordable for regular users. In contrast, passive methods of depth reconstruction rely on image understanding (e. g., scene illumination, texture) where one or more images or videos are collected for the targeted objects. Photometric stereo (PS) falls into the former category due to active illumination. PS [Woo80] is a technique to estimate surface orientation which relies on two or more images captured at a fixed viewpoint with varying illuminations. Most of the current existing PS approaches [AZK08, ASC11, CAK07] are based on quite a few assumptions, such as uniform albedos, orthogonal projections, the absence of shadowing and noise. One often exploited assumption is that light sources are at a far-away position which is not applicable due to the fact that the scene dimensions are usually not significantly smaller in comparison with the distance between the light source and objects. In contrast, near-light PS can resolve the far-away light source assumption. However existing near-light PS methods [MWBK14, MQ*16] require special setups, and the data capturing process is a tedious task that constrains the light sources to follow predefined paths. These limitations motivated our work, which enables a more robust reconstruction and lifts many of the existing constraints. In Chapter 2, we will introduce our algorithm to estimate a depth map for an indoor scene via near-light PS with a carefully-designed but low-cost acquisition setup.

We present a solution to deal with the near-light PS problem, which is robust to the aforementioned limitations, such as albedo variations, perspective projections and the presence of shadows and noise. To achieve this goal, our method utilizes a video se-

1

quence of varying lighting conditions captured with a simple, uncalibrated, and affordable setup using specular spheres. The sphere positions are detected with a perspective-correcting Hough transform, with which the light calibration is robustly performed by analyzing the light's reflection via a least-squares approach in the presence of outliers. Given the estimated light positions, the near-light PS problem can be robustly solved, leading to a simultaneours reconstruction of scene parameters including depth, relative albedo and normal.

The proposed method was published in: *Indoor Scene Reconstruction Using Near-Light Photometric Stereo*
Jingtang Liao, Bert Buchholz, Jean-Marc Thiery, Pablo Bauszat and Elmar Eisemann
IEEE Transactions on Image Processing, 2017

USER-GUIDED DEPTH MAP DESIGN

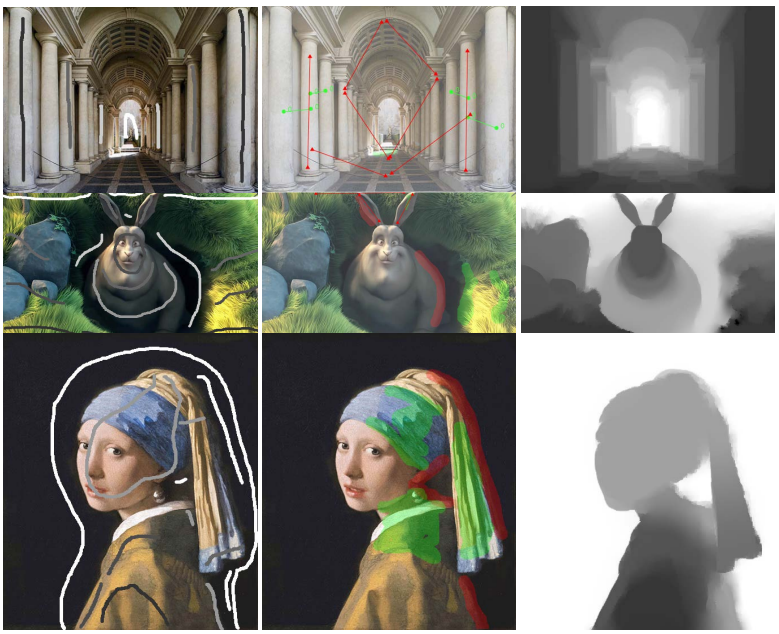


Figure 1.3: Example depth maps generated by our approach.

User-guided depth map design is an important element, when using a depth map for image manipulation. For artistic purposes, a realistic depth map is not always the best choice [LHW*10, DRE*11]. While previous work exists that builds upon user interaction in the form of sparse scribbles [WLF*11] or points [LGG14], these solutions have mostly been made for realistic or plausible depth-map generation. In Chapter 3, we focus on providing a variety of new interactive tools to guide the depth-design process that serves as an input to an artistic filter. Our depth-map generation relies on a diffusion process,

initialized by a set of input scribbles placed by the user. Additionally, the process can be influenced by additional annotation tools such as a non-linear depth mapping, directionality, emphasis, or reduction of the influence of image cues. We show that the derived depth maps can be directly applied to an application for various depth-based effects, including wiggle stereoscopy, depth-of-field, and unsharp masking.

This work was presented in Graphics Interface Conference 2017 and an extended version was referred to Computer & Graphics.

Depth Map Design and Depth-based Effects With a Single Image
Jingtang Liao, Shuheng Shen and Elmar Eisemann
Graphics Interface, 2017

Depth Annotations: Designing Depth of a Single Image for Depth-based Effects
Jingtang Liao, Shuheng Shen and Elmar Eisemann:
Computers & Graphics, (submitted)

ENHANCEMENT OF DEPTH PERCEPTION VIA SPLIT-DEPTH IMAGES

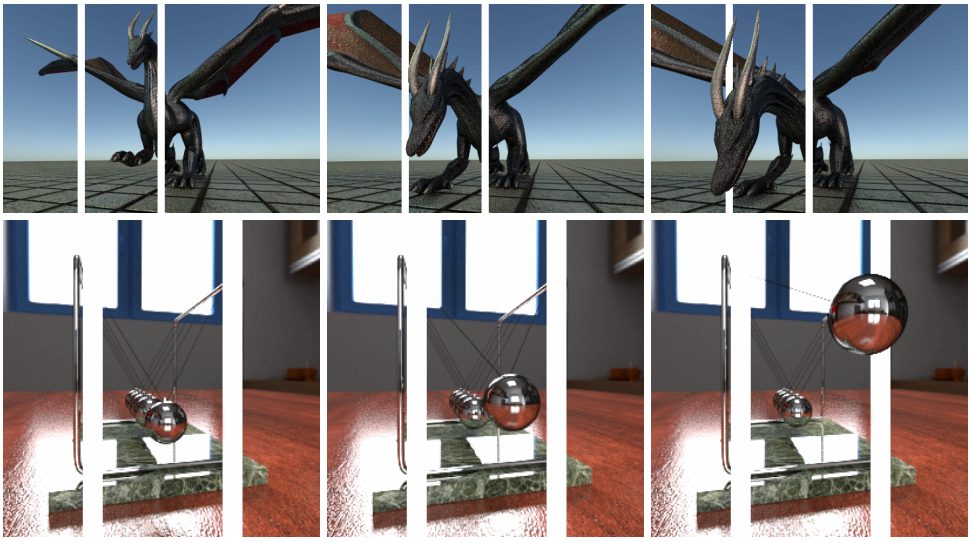


Figure 1.4: Example frames of split-depth images generated by our method.

In Chapter 4, we present a new opportunity to enhance depth perception. Occlusion is one of the strongest cues of the human visual systems to interpret the depth ordering. This observation is also exploited by artists when employing a passepartout - a paper or cardboard sheet with a cutout. We have seen methods [SCRS09, RTMS12] that propose to transfer this knowledge to digital images by adding virtual passepartouts, producing a strong "popping out" or "floating on the window" sensation. Our work handles also

dynamic scenes and focuses on a generalization of this idea, in form of split-depth images. Split-depth images utilize a similar principle as passepartouts, usually in the form of vertical or horizontal bars, to increase the 3D effect in a video clip. Hereby, a division between the mental fore- and background is created and an overlapping object is usually interpreted as moving out of the image.

We study different factors that contribute to the illusion and propose a solution to generate split-depth images for a given RGB + depth image sequence. Based on a motion summarization of an object of interest through space and time, we can formulate the bar positioning as an energy-minimization problem. The energy formulation is based on a number of visual perception experiments. We demonstrate the effectiveness of our approach on a variety of examples. Our study with novice users shows that our method allows them to quickly create satisfying results even for complex animations.

This work was presented in Pacific Graphics, 2017 and published in an issue of the Computer Graphics Forum (CGF), the journal of the Eurographics Association:
Split-Depth Image Generation and Optimization
Jingtang Liao, Martin Eisemann, and Elmar Eisemann
Computer Graphics Forum, 2017

I

This part of the dissertation presents our first solution to obtain a depth map using near-light PS. Our approach removes the constraints appearing in previous methods and reconstructs a depth map for an indoor scene truthfully by a carefully-designed but low-cost acquisition setup, which we describe in Chapter 2.

2

INDOOR SCENE RECONSTRUCTION USING NEAR-LIGHT PHOTOMETRIC STEREO

We propose a novel framework for photometric stereo (PS) under low-light conditions using uncalibrated near-light illumination. It operates on free-form video sequences captured with a minimalistic and affordable setup. We address issues such as albedo variations, shadowing, perspective projections and camera noise. Our method uses specular spheres detected with a perspective-correcting Hough transform to robustly triangulate light positions in the presence of outliers via a least-squares approach. Furthermore, we propose an iterative reweighting scheme in combination with an ℓ_p-norm minimizer to robustly solve the calibrated near-light PS problem. In contrast to other approaches, our framework reconstructs depth, albedo (relative to light source intensity) and normals simultaneously and is demonstrated on synthetic and real-world scenes.

This chapter is based on the following publication: **Jingtang Liao**, Bert Buchholz, Jean-Marc Thiery, Pablo Bauszat and Elmar Eisemann, "Indoor Scene Reconstruction Using Near-Light Photometric Stereo", *IEEE Transactions on Image Processing*, vol.26, no. 3, pages 1089–1101, 2017.

2.1. INTRODUCTION

Photometric stereo (PS) [Woo80] is a technique to determine surface orientation from two or more images with a fixed viewpoint but differing lighting conditions. It is widely used in computer vision and graphics, e.g., for 3D scene reconstruction or geometry-based image relighting.

Current PS approaches impose a significant number of restricting constraints on the scene and illumination, such as a uniform albedo, orthographic projection, or absence of shadows. An often employed assumption is that light arrives from a distant source (i. e., parallel light rays), leading to the same incident light direction and radiance for each scene point. Such a constraint usually forces the scene to be small-scale, as the assumption does not hold if the distance to the light source is not significantly larger than the scene dimensions. Furthermore, generalized bas-relief (GBR) [BKY99] coupled with the constraint of integrability can solve only up to three scene parameters and leaves room for geometric ambiguity. In contrast, near-light PS models can reconstruct entire indoor scenes, but typically require careful light calibration to be successful. This step often involves specialized equipment and complex setups. An advantage is that the added illumination even makes a capture in badly lit environments possible, where pure stereo reconstructions can fail.

In this chapter, we propose a new approach to PS that aims at relaxing as many of the previously-mentioned assumptions as possible and recovers scene parameters (depth, albedo, and normal) simultaneously involving a cheap, uncalibrated, and simple setup. Our framework reconstructs indoor scenes by solving the near-light PS problem from a sequence of images extracted from a captured video. During the capture, a light source is moved through the scene while the camera's viewpoint is kept fixed. Several reflective spheres are arbitrarily placed in the scene beforehand for a robust, yet effortless light calibration. While this setup has been employed before in several existing approaches [Nay89, ASSS14, RDL*15], it typically suffers from two issues. First, the unknown locations of the spheres have to be robustly estimated from the input images and even small deviations can lead to significant errors in the light triangulation. Second, highlights on spheres can potentially be reflected in other spheres and are assumed to come from a perfect point light, which is not true in practice where the light source typically has area. Our framework addresses these issues and improves the robustness of traditional light calibration approaches by several means. By acquiring a video, we can choose a reliable set of frames to make a robust estimate possible. Similarly, by testing multiple light configurations in combination with a trimmed least-squares approach, we can successfully triangulate its position and obtain the light's center with a significantly-reduced reconstruction error. Additionally, we propose to use a novel sphere detection approach based on a perspective-correct, closed-form parameterization which is suitable for Hough transform. Finally, we can solve for various scene parameters (normal, albedo and depth) simultaneously by using an energy formulation derived from the calibrated near-light PS model.

Overall, our work on the near-light PS problem considering perspective projection and light attenuation makes the following technical contributions:

An efficient minimization of our weighted ℓ_p-norm energy more robust to noise and outliers compared to the traditional ℓ_2-norm;

A robust and elegant sphere position estimation based on the Hough transform to handle perspective projections.

A simple light calibration setup using uncalibrated specular spheres with unknown positions.

2.2. RELATED WORK

We will first briefly discuss related work for sphere detection and light calibration, as well as near-light photometric stereo.

2.2.1. SPHERE DETECTION

It is crucial to estimate the positions of the reference spheres in the scene from the input images to reconstruct the light position. Unfortunately, a simple circle detection is not accurate, because the projections of the spheres onto the image plane are affected by the perspective projection leading to ellipsoids. A general method to detect ellipses has been proposed by D.H. Ballard [Bal81], which uses a Hough transform into a 5-dimensional parameter space. However, using five parameters is computationally expensive and various modifications were proposed to maintain robustness and reduce computational complexity by exploiting ellipse symmetry [TM78, LW99], randomization [McL98], special acceleration techniques [XJ02], or reduction to a one-dimensional parametric space [CLER07]. Additionally, directly estimating the sphere's center from the orientation point of its ellipsoid projection obtained from these approaches is inaccurate and, hence, these approaches are not directly suitable candidates for the required 3D sphere reconstruction. The practical problem of stable sphere localization under perspective projection is underrepresented in the literature, which usually requires manual work [CPM*16] or many views [Len04, YZ06] to localize the sphere center. In contrast, we propose a modified Hough transformation, which robustly computes the sphere's location from only one view and incorporates perspective projections, but only requires a 3-dimensional parameter space (Sec. 2.4.1).

2.2.2. LIGHT CALIBRATION

Light calibration often requires specialized non-portable equipment [WWH05, DHT*00] or relies on constraints regarding the varying light positions; such as fully controlled light paths [Cla92] or restricted locations (e.g., a roughly hemispherical pattern, for which the light position can be determined by dimensionality reduction [WMTG05]). For general light positions, reference spheres can be used for the localization process. Nayar [Nay89] proposed the *Sphereo* method which triangulates the position of the light based on its reflection in two reflective spheres and has been used in several recent approaches [ASSS14, RDL*15]. While the detection of the light reflection is eased with a calibrated setup (including known sphere positions and geometry) [PSG01], in practice, highlight detection is prone to noise and interreflection, in particular when relying on low-dynamic range imagery, which is typically acquired in a video setup. Ackermann et. al's general light-calibration method minimizes the image-space error of highlights reflected off specular spheres [AFG13], however, their method requires high-dynamic range images. Masselus et. al [MDA02] presented the *Free-form Light Stage*, which uses the shading patterns on

four diffuse spheres to estimate the illumination direction following Lambert's cosine law. However, their approach focuses on computing only the dominant light direction and cannot accurately estimate the light position. In our setup, we use multiple, simple reflective spheres with unknown position. Still, we robustly reconstruct the light location even in the presence of outliers and partial occlusion of the spheres.

2.2.3. NEAR-LIGHT PHOTOMETRIC STEREO

Traditional photometric stereo algorithms use a distant light model, with lots of efforts having been made to cope with perspective projection [TK05], albedo variations [AZK08, ASC11], shadows [CAK07, SZP10] and non-Lambertian corruptions such as specularities and noise [IWMA12]. Chandraker et al. [CBR13] present a comprehensive theory of photometric surface reconstruction from image derivatives in the presence of general, unknown isotropic BRDFs. However, the motion of the light source is constrained to circular motion around the camera axis and requires a specific acquisition setup. Recent studies [AHP12, HGZGL15] attempted PS reconstruction on outdoor data using the sun light for which the distant light source assumption holds. Nonetheless, a distant light model makes geometry reconstruction ambiguous.

To tackle this problem, Iwahori et al. [ISI90] introduced a near-light PS model to better recover depth details. However, their approach assumed a calibrated setup and perfectly uniform diffuse surfaces. It was later improved by detecting diffused maxima regions [ASSS14], but still ignored light attenuation. Uncalibrated near-light PS models often suffer from artifacts due to shadows in the input images [PF14] or restricting C^0-surface assumptions [XDW15], making it impossible to deal with depth discontinuities and varying object albedo. A calibrated nearlight PS model proposed by Mecca et al. [MWBK14, MQ*16] pays special attention to faithfully model perspective projection, the point light source and shadowing by exploiting the image ratios. However, they use a special setup to constrain the light positions, require the surfaces to be connected, and the existence of at least one reference point per surface. Some issues of near-light PS can be overcome by using multi-view PS [HMJI09], however, this requires a more costly and complex acquisition setup and is out of the scope of this chapter.

Compared with state-of-the-art methods for near-light PS, our approach has clear distinctions. First, we do not require any special setup besides several reflective spheres and an active light sources, and light sources are not constrained to move on restricted paths. Second, we use a large number of input frames and let our algorithm choose the observations that mostly correspond to diffuse reflectance, which allows us to estimate the result even in the presence of specularities and shadows. The selection is done automatically by, among others, minimizing a sparsity-inducing ℓ_p norm. Third, our model recovers the scene parameters (normal, depth, and albedo) simultaneously. Fourth, we use dedicated strategies to enforce local albedo and geometry smoothness.

2.3. OVERVIEW

Our approach is illustrated in Fig. 2.1. In a (not necessarily) dark room, the camera is placed at a fixed view point and reference spheres are distributed throughout the scene for calibrating the light position. Then the video acquisition starts. In the beginning of

the recording, i.e., *before* the light bulb is turned on, we record a few seconds to solely capture the ambient lighting. Using the average of these initial frames of the captured video clip, the constant ambient lighting map of the scene can be estimated and subtracted from the remaining frames. Then the light bulb is turned on and the user walks through the scene, illuminating it by waving the light bulb and covering as many light positions as possible. Only the frames in which the light bulb is turned on are used as input for the light calibration and scene reconstruction. Besides a gamma correction (response linearization) and subtraction of the ambient lighting, no further processing is applied to the frames.

We seek to recover the scene parameters including normal, albedo and depth for a given scene. To this extent, we first estimate the reference spheres' position once via a perspective-correcting Hough transform cone detection (Sec. 2.4.1). We triangulate the light position for each frame using the rays reflected towards the light from its reflection on the reference spheres. To handle wrongly detected or distorted highlights on the spheres robustly, we compute the light positions via a trimmed linear least-squares optimization (Sec. 2.4.2). Finally, we recover the scene parameters by extracting a subset of reliable observations for each pixel and employing an ℓ_p-norm minimizer combined with a reweighting scheme that is designed to robustly handle noise and occlusions (Sec. 2.5). We will demonstrate our approach on rendered scenes (to have access to a reference reconstruction), recorded real-world scenes, and compare our solution to existing work (Sec. 2.6).

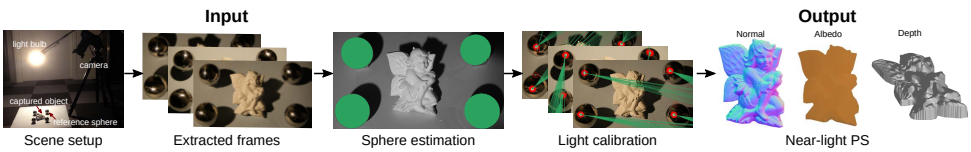


Figure 2.1: **From left to right:** We first capture a free-form video using a minimal setup consisting only of a regular camera and light bulb, as well as a set of reference spheres. We extract frames from the captured video (second image) and estimate the reference spheres' positions (third image) to calculate the light position for each input frame using the light's reflection (fourth image). Finally, the scene parameters for normal, albedo, and depth are computed by solving the calibrated near-light photometric stereo problem using a robust reweighting scheme.

2.4. LIGHT CALIBRATION

Our goal is to estimate the light positions for each input frame based on the reference spheres in the scene. The spheres can be placed arbitrarily, but should be well distributed around the acquisition area, as this has been proven to work well in existing calibration setups. We will first discuss the detection of the reference spheres without any prior knowledge about their position. The only user input is the world radius r of these spheres to fix the absolute scale of the scene. Later, we will show how to robustly triangulate the light position for each frame using the highlights (the light's reflections) on the spheres.

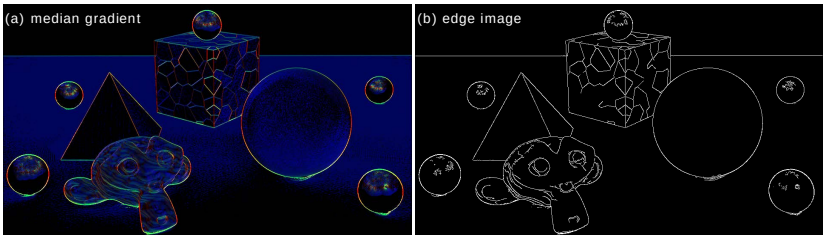


Figure 2.2: Robust edge detection using all frames. **(a)** The median gradient image over all input frames. **(b)** Edge detection result computed by thresholding the median gradient image.

2.4.1. SPHERE POSITION ESTIMATION

We aim at reconstructing the positions of the spheres in world coordinates (with the camera at the origin) using all input frames. By detecting the shape of the sphere's projection in the image plane, we derive its position using the projected center and known sphere extent. We use a Hough transform for the shape detection, which finds the most likely parameters for the shape model. Typically, the parameter space of the shape model (e.g., for circle detection, one would use the 2D center and radius) is subdivided into candidate bins. For each candidate bin, the corresponding shape is tested against the detected edges of the input image. The candidate with the most (normalized) edge-pixel consistency on the shape's boundary is assumed to be the best parameter estimate. Consequently, a robust edge detection in the input image is a key component. Directly applying a Canny edge detection on a randomly chosen input frame leads to unreliable results, because edges often are ignored (due to low-illumination regions and occlusions) or introduced by cast shadows. Therefore, we propose to first estimate the gradient images of all input frames separately, and then compute the median of the gradients for each pixel, which is a robust estimate that can be used as input to the edge detection (Fig. 2.2). To additionally avoid the rare case that almost all observations of a pixel are shadowed or over-saturated, we perform the median gradient calculation on a per-pixel level and exclude too bright or too dark observations. In practice, we exclude the brightest 20% of the brightest pixel (each channel) and 10% of the darkest pixel observations, which is a reasonable assumption for roughly uniform illumination directions. In all examples, 0.2 and 0.5 are used for the Canny edge detection double thresholding.

In our situation, using a Hough circle detection is not suitable. The projection of a sphere onto the image plane corresponds to an intersection of a plane and a cone with apex at the view point and defined by the sphere's silhouette, which is generally a conic section (Fig. 2.3). Only if the sphere's center projects to the very center of the image plane, we obtain a sphere. In most PS algorithms with reference spheres [ASSS14, WMTG05], the projection is inaccurately considered to be a circle, resulting in errors when the sphere is placed in image corners where the elliptical shape is most pronounced. Although traditional ellipse-detection methods could be used to account for perspective distortions, the resulting ellipses cannot be used directly to estimate the sphere position because the projected sphere's center is typically not the orientation point of the ellipse.

In consequence, we propose a novel parameter model which correctly takes the perspective distortion into account. We parameterize a cone using the half opening angle

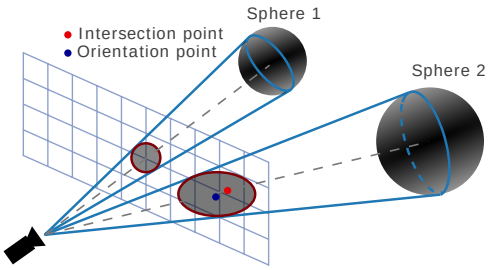


Figure 2.3: Conic intersection between a sphere and the image plane. A sphere's projection on the image plane only resembles a circle at the very center of the image plane (Sphere 1) and is typically an ellipse (Sphere 2) due to perspective distortion. Using the orientation point (blue dot) of the ellipse is not an accurate estimate of the sphere's world center, since it does typically not correspond to the intersection point between a view ray from the camera to the sphere's center (red dot).

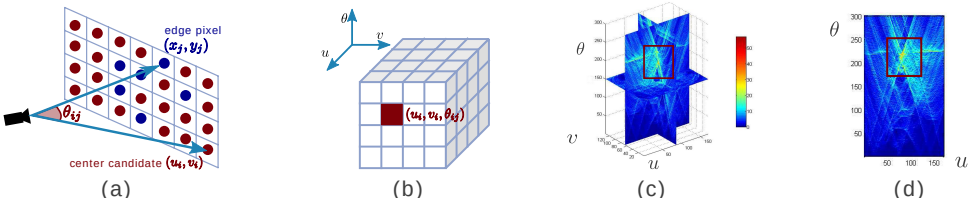


Figure 2.4: An overview of the cone-based Hough transform model. **(a)-(b)** Each image pixel (u_i, v_i) is considered as potential candidate and for each edge pixel, the cone angle θ_{ij} is computed and the bin (u_i, v_i, θ_{ij}) in the Hough parameter space is increased. **(c)-(d)** A 3D visualization and a θ-u slice of the filled parameter space (u,v,θ) show that the most-likely candidates (color-coded with blue to red) lie in the red dashed region.

θ and the image coordinates (u, v) for the intersection between the cone's axis and the image plane (Fig. 2.4). Assuming the camera focal length f, the sensor size w_s, h_s and image resolution w, h are known, it is possible to construct the cone in world coordinates, as its axis orientation is given by $A_i := (u - w/2, v - h/2, fh/h_s)$. Note that this defines a 3-dimensional parameter space. We discretize the parameter space and define uniform bins $\mathscr{B}_{uv\theta}$, each representing a possible cone candidate. Each detected edge pixel $P_j := (x_j, y_j)$ will increase a counter in all bins (u_i, v_i, θ_{ij}) whose corresponding candidate shape contains P_j on its boundary (Fig. 2.4). In this setup, (u_i, v_i) is the center of the candidate cone and θ_{ij} is the opening angle of the candidate cone (the angle between the rays going through (u_i, v_i) and P_j). After treating all edge pixels, we normalize the bins by the circumference of the represented ellipse and for n spheres in the scene, we choose the n bins with the highest votes to retrieve their location. Having determined a candidate, the position of the corresponding sphere can be computed by $\mathbf{c} = \mathbf{a}\frac{r}{\sin\theta}$ where $\mathbf{a}$ is the normalized camera ray pointing from the camera to the intersection point and r the world radius of the sphere. To avoid a bias towards small spheres (e.g., with size of a single pixel) or wrongly detected sphere-like objects in the scene, we ask the user to provide a rough size interval. Alternatively, a user can also drag bounding boxes around the spheres to indicate their rough locations in the image, further accelerating the detection process. A precise indication of the spheres is not needed.

2.4.2. LIGHT POSITION ESTIMATION

Once the locations of the spheres (which are constant over all frames) are known, the world position of the light source can be estimated for each input frame. By using the light's reflection on the spheres (specular highlights), rays from the eye reflected off the spheres and towards the light source can be computed. The light position is then defined as the point closest to the reflected rays. Note that each frame is only required to have at least two spheres with a reflective highlight. Frames which do not meet this requirement are discarded.

The first step is to detect the light's reflection on each reference sphere in image space. For low-dynamic range images, we consider the pixels whose intensity is above 95% as highlights. Since the light source is not a perfect point light in practice, its reflection is typically an irregularly shaped highlight. A standard solution is to calculate the averaged pixel position within the highlight blob as the light reflection on the spheres [ASSS14]. However, it is potentially inaccurate since a discrepancy of one or two pixels can immediately lead to larger errors for the light-ray reconstruction. Instead, our approach uses all pixels associated with highlights during reconstruction as candidates. We will later show how to prune this set. Moreover, our method is able to consider sub-pixel level precision to reduce the influence of the limited image resolution.

A candidate light ray for a pixel representing a highlight can directly be constructed from its coordinates. Given the i-th sphere with center $\mathbf{c_i}$ and radius r, and the pixel coordinate (hl_x, hl_y), the 3D point on the sphere $\mathbf{p_{hl}}$ is simply given by $\mathbf{p_{hl}} = \lambda_{hl}\mathbf{a}$ where $\mathbf{a}$ is the unit vector pointing from the camera to the highlight and λ_{hl} is the camera distance to the point. By verifying $||\lambda_{hl}\mathbf{a} - \mathbf{c_i}||^2 = r^2$, the camera distance can be written as $\lambda_{hl} = \mathbf{a}\cdot\mathbf{c_i} - \sqrt{(\mathbf{a}\cdot\mathbf{c_i})^2 + r^2 - ||\mathbf{c_i}||^2}$. The sphere normal $\mathbf{n_{hl}}$ at this point is $\overrightarrow{\mathbf{c_i p_{hl}}}/||\overrightarrow{\mathbf{c_i p_{hl}}}||$, which finally leads to the reflected ray direction $\mathbf{l} = \mathbf{a} - 2(\mathbf{a}\cdot\mathbf{n_{hl}})\mathbf{n_{hl}}$.

Trimmed least-squares approach Given a set of N candidate light rays, we will derive the light source position $\mathbf{b}$ as the closest point to the actual reflected rays. One problem for light calibration in real-world scenes is that spheres inter-reflect among each other leading to wrong highlight assumptions. Hence, we first discard light rays, which intersect with other reference spheres. Still, even the remaining candidate rays are not all reliable due to noise (or extended highlights) and we propose a weighted trimmed least-squares approach to address this problem. Initially, an estimate of the light position is found using regular least-squares fitting using all rays. In the next step, we perform multiple refinement iterations, each time removing one or more rays with the largest residual error for each sphere, until k rays remain (k is the number of spheres with rays). The least-squares problem for the set of rays $\mathbf{r} = (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_N)$, is defined via an energy function consisting of the sum of squared distances to these rays:

$$\mathfrak{C}(\mathbf{b}) = \sum_{i=1}^{N} \omega_i d(\mathbf{b}, \mathbf{r_i})^2,$$

where $d(\mathbf{b}, \mathbf{r_i})^2$ is the squared distance between the light position and the ray. One can see that $d(\mathbf{b}, \mathbf{r_i})^2$ is a quadric[1] with respect to $\mathbf{b}$, and therefore $\mathfrak{C}(\mathbf{b})$ is also a quadric

[1] $d(\mathbf{b}, (q_i; \vec{v_i}))^2 = \mathbf{b}^t \cdot {A_i}^t \cdot A_i \cdot \mathbf{b} - 2{B_i}^t \cdot \mathbf{b} + \text{const}$, with
$A_i := I - \vec{v_i} \cdot \vec{v_i}^t$, $B_i := {A_i}^t \cdot A_i \cdot q_i$, where $\vec{v_i}$ is the unit direction of the ray and q_i is its basis 3D point.

with respect to **b** and can be minimized efficiently. The weighting factor ω_i defines the *reliability* of the ray $\mathbf{r_i}$. Since the normal variation towards the edge of a projected sphere is larger than in its center, we regard highlights closer to the center as more reliable. Small errors in highlight position estimation have a significantly higher impact close to the edge. Therefore, we use the angle θ_i between the ray from the camera to the sphere center, and the ray from the highlight to the sphere center to weigh the ray's contribution. When more than one highlight (and therefore ray) is detected for one sphere, we further weigh the ray by the total number of rays for that sphere, denoted by M. The weight for a ray $\mathbf{r_i}$ is thus given by $w_i = \frac{\cos\theta_i}{M}$.

2.5. VIRTUAL SCENE RECONSTRUCTION

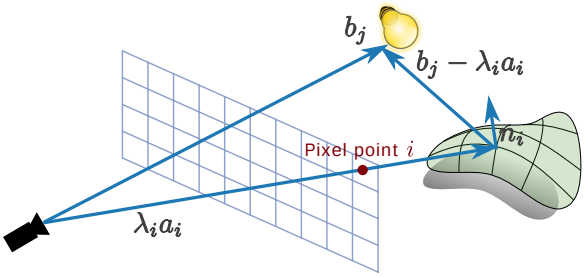


Figure 2.5: The near-light photometric stereo model describes the color of a pixel i as the light arriving from a scene point given by the pixel's λ_i and the view ray $\mathbf{a}_i$ with normal $\mathbf{n}_i$. The point is assumed to be illuminated by a point-light source at $\mathbf{b}_j$ which varies through all input frames.

After the light positions have been estimated for each frame, our goal is now to recover the scene parameters using the near-light PS model. The near-light PS model (Fig. 2.5) relates albedo $\rho_\mathbf{i}$, normal $\mathbf{n}_i$, and depth λ_i for each pixel i and is defined as

$$\mathbf{m_{ij}} = \frac{\rho_\mathbf{i}\big(\mathbf{n}_i \cdot (\mathbf{b}_j - \lambda_i \mathbf{a}_i)\big)}{\|\mathbf{b}_j - \lambda_i \mathbf{a}_i\|^3}$$

where $\mathbf{m_{ij}}$ is the observation (color) for pixel i in frame j, $\mathbf{b}_j$ the light position at frame j, and $\mathbf{a}_i$ the normalized vector pointing from the camera to the pixel's 3D position, which is $\lambda_i \mathbf{a}_i$ (compare to Sec. 2.4.1). Note that the given formulation of the near-light PS model respects perspective projection and light attenuation. While the model does only account for diffuse material, we can still obtain a robust reconstruction in the presence of specular materials with a simple strategy that chooses input frames which are more likely to represent a diffuse response, which is the advantage of having a large set of input frames available. The scene parameters are typically found using energy minimization, where the energy is defined as the difference between the current near-light model's state and the observed pixel color. The input to our reconstruction approach is the set of pixel observations $(\mathbf{m_{i1}}, \mathbf{m_{i2}}, \ldots, \mathbf{m_{ij}})$ for each of the $1 \ldots j$ video frames with corresponding light positions $(\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_j})$. We first perform a pixel-based frame selection to exclude observations that are outliers due to specularities, over-saturation, and

shadowing. Then, we formulate the problem of recovering the scene parameters as an ℓ_p-norm optimization problem combined with a reweighting scheme based on different characteristics of the data set. Although, an ℓ_p-norm optimization is known to be computationally involved, it can be efficiently solved using an iterative Newton procedure. Further, we add three extensions, which relax the assumption of a fully local reconstruction; exploiting spatial coherence for improved convergence, a smoother albedo reconstruction, and a robust handling of pixels with insufficient observations. Finally, we show how to iteratively refine the light positions obtained in Sec. 2.4 using the reconstructed scene parameters results.

2.5.1. PIXEL-BASED FRAME SELECTION

Since some pixel observations correspond to outliers and should be ignored during reconstruction (e.g., occlusion due to the person moving the light, cast shadows, specular reflections, and over-saturation), we select a reliable subset of observations for each pixel as a first step. Specularities and over-saturations are usually sparse, but appear significantly brighter when the light source is situated along the reflection direction and usually share the light's white color. In order to reconstruct the scene, we opt at eliminating such outliers, obtaining an observed diffuse behavior. We apply a two-step process. First, we exclude observations that are too bright or too dark in the same way as for the computation of the median gradient image (Sec. 2.4.1). The purpose of this approach is solely to remove strong outliers defined by the range of LDR images and thus, the thresholds are robust to small changes. In a second step, we remove observations that are smaller than 70% of the median value of the remaining observations after the first step. While the first step removes outliers at absolute boundaries, the second step defines outliers relative to the remaining pixel observations.

2.5.2. REWEIGHTED OPTIMIZATION USING THE ℓ_p-NORM

With a large number of observations and a few unknowns, we have an overdetermined problem, which we cast into an energy minimization problem and first solve for each pixel independently. Additionally, we propose to use an iterative scheme [LFDF07] to change the influence of certain observations based on the current solution, exploiting observed intensities, as well as the known geometric distribution of the light. In the following, we detail the reconstruction.

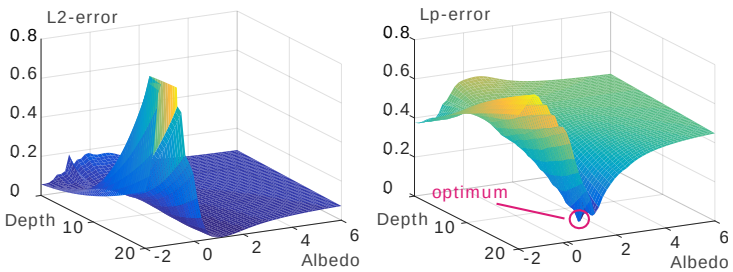


Figure 2.6: Energy error profiles using the ℓ_2-norm (on the left) and the ℓ_p-norm (on the right) with $p = 0.5$ for a single pixel with varying depth and albedo. The optimum in the ℓ_p-norm is more pronounced.

Unfortunately, the energy function can still be distorted by wrong observations (e.g., from camera noise). To provide a robust reconstruction in the presence of outliers, we employ the ℓ_p-norm [BTP13] with $p <= 1$ instead of the ℓ_2-norm, this choice is known to robustly handle significant amounts of noise. Fig. 2.6 compares the energy profile of a single pixel with changing depth and albedo, while keeping the normal fixed, using the ℓ_2-norm and ℓ_p-norm (with $p = 0.5$). This example is typical and illustrates intuitively why the minimizer is easier to identify using a sparsity-inducing norm such as the ℓ_p-norm, even if this energy function is not convex. We use $p = 0.5$ for all examples. The energy function of a pixel i is given by

$$F_i(\mathbf{n}_i, \lambda_i, \rho_\mathbf{i}) = \sum_j \omega_{ij} E_{ij}(\mathbf{n}_i, \lambda_i, \rho_\mathbf{i}), \tag{2.1}$$

where the error function E_{ij} is based on the near-light PS model and is defined as

$$E_{ij}(\mathbf{n}_i, \lambda_i, \rho_\mathbf{i}) := \left\| \mathbf{m_{ij}} - \frac{\rho_\mathbf{i}\big(\mathbf{n}_i \cdot (\mathbf{b}_j - \lambda_i \mathbf{a}_i)\big)}{\|\mathbf{b}_j - \lambda_i \mathbf{a}_i\|^3} \right\|^p. \tag{2.2}$$

Each observation is multiplied by a weight ω_{ij}, which is composed of three individual weights, and addresses further outlier handling, non-uniform light distributions, and geometric properties of the current reconstruction state:

$$\omega_{ij} = \omega_{ij}^{\text{ld}} \cdot \omega_{ij}^{\text{outl}} \cdot \omega_{ij}^{\text{hs}}.$$

Light-Distribution Weight (ω_{ij}^{ld}) The distribution of light positions over a scene point is an important factor for ensuring convergence. E.g., lights distributed along a line in direction of a scene point, would only lead to attenuation changes at this location, which is insufficient. Furthermore, depending on the movement of the light source, some directions potentially receive significantly more observations than others. For instance, Fig. 2.7 (left) illustrates an exemplary non-uniform light distribution over a hemisphere of a scene point and it can be observed that area A and B exhibit a dense light sampling. We propose to balance the importance of the directional sampling by setting ω_{ij}^{ld} to be the inverse of the light's density. Since the input is a discrete set of observations, we estimate an approximate density by subdividing the directional sphere around a scene point in equally-sized regions. For this task, we employ HEALPix (Hierarchical Equal Area iso-Latitude Pixelization) [GHB*05], which is a suitable approach to discretize the surrounding sphere into N_s equal areas with similar shape (Fig. 2.7). In our implementation, we use $N_s = 30$, which gives satisfying results.

Outlier Weight ($\omega_{ij}^{\text{outl}}$) Even after the initial pixel-based frame selection, some pixel observations might still correspond to outliers and should be ignored during scene reconstruction. When an observation has a significantly larger error compared to the average error of all observations, we assume that this observation is an outlier and reduce its importance. For pixel i at frame j, we compute the outlier weight as a relation of its error E_{ij} to the mean error $\overline{E_i}$ for all observations in i and set $\omega_{ij}^{\text{outl}} := e^{-\frac{E_{ij}}{E_{mean}}}$.

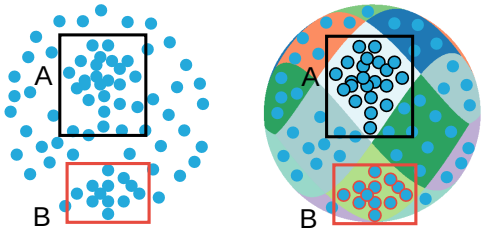


Figure 2.7: The distribution over spherical directions can be non-uniform depending on the captured light positions (e.g., area A and B are more densely sampled). Using HEALPix, the density is approximately described by a set of discrete equally-sized regions. The observations are then reweighted with the inverse of the density to simulate a uniform light distribution.

Half-space Weight ($\omega_{ij}^{\mathbf{hs}}$) When a light is in the opposite side of the plane defined by a point's normal, it implies that the dot product of the normal and the vector from the point towards the light is negative, hence, it cannot contribute to the points illumination. In this case, we want to set the observation's weight to 0 (otherwise to 1). Theoretically, as shown in Fig. 2.8, frames for which a pixel is in shadow (Fig. 2.8, middle) are excluded by our pixel-based frame-selection technique (Sec.2.5.1) and only the ones for which the light source illuminates the pixel are kept (Fig. 2.8, left). However, in practice, due to light reflections, some frames might be kept, even though the light source does not illuminate the corresponding point directly (Fig. 2.8, right). The "half-space weight" penalizes light positions behind the plane described by the pixel's position and normal (in this configuration, the light cannot illuminate the pixel directly).

Although we rely on a rough estimate of the normal and could potentially ignore valid observations, the initial solution and the large amount of frames prove sufficient in practice. An alternative would be to use the half-space weight only after a certain number of iterations when the normal estimate is more stable.

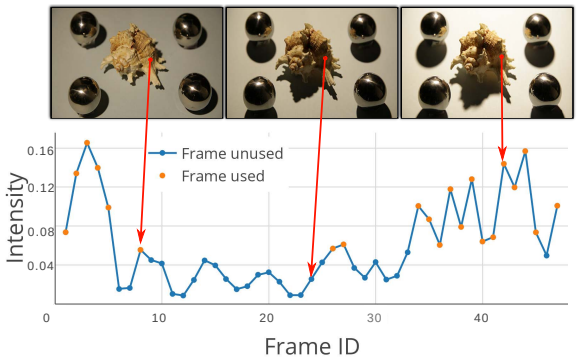


Figure 2.8: Motivation of the half-plane weight. Blue (resp. orange) dots correspond to frames which were discarded (resp. kept) by our pixel-based frame selection technique (Sec.2.5.1). The half-space weight can help further discard observations which are in strong global illumination, though the lights are in the opposite side of the plane defined by a point's normal.

2.5.3. NUMERICAL SOLVING

To solve the energy function in Eq. 2.1, we employ a Newton procedure (Alg. 1). The six scene parameters can be divided into two categories, the color parameters $(\rho_{ri}, \rho_{gi}, \rho_{bi})$ and the geometric parameters $(\theta_i, \phi_i, \lambda_i)$. Here, we express the normal $\mathbf{n}_i$ using spherical coordinates (θ_i, ϕ_i) in a local frame based on the camera ray $\mathbf{a}_i$ to reduce the number of parameters. To create local frames that vary smoothly across the image, we define the two vectors orthogonal to $\mathbf{a}_i$ as $\mathbf{e}_{i1} := \mathbf{a}_i \times \mathbf{t}$ and $\mathbf{e}_{i2} := \mathbf{a}_i \times \mathbf{e}_{i1}$ with $\mathbf{t} = (0,1,0)$.

For each image point, we first initialize the parameters (line 1-3) by setting the albedo to $[1,1,1]$ and the normal to $[0,0]$ (expressed in the local coordinate frame and, hence, aligned with the camera view). For the depth parameters, we use the average depth of the reference spheres detected during the light calibration process. For each iteration, we update the weights (line 6) and compute the 6×6 Hessian matrix $\mathscr{H}$ (line 7). The inverse matrix of $\mathscr{H}$ is computed by solving the system of 6×6 linear equations using Gaussian elimination (line 8). At the end of each iteration, we constrain the normals to face towards the camera (line 9-11). We iterate this process around 200 iterations, which usually ensures a good convergence as shown in Fig.2.15.

Newton method in ℓ_p-norm Since the function $f(x) = x^p$ is non-differentiable in 0 ($\partial_x f(x) = p x^{p-1}$) for $p < 2$, standard Newton and gradient-descent methods are usually not suitable, and often an *alternating direction method of multipliers* is used instead. Instead, we chose to reformulate the Newton method by approximating the first and second order of the function $f_p : X \mapsto |F|^p$ (which we rewrite as $f_p : X \mapsto (|F|^2)^{(p/2)}$) in Eq. 2.2 as

$$\partial_x f_p \approx \frac{p}{2} |F+\epsilon|^{p-2} \partial_x F$$
$$\partial^2_{xy} f_p \approx \frac{p}{2} \frac{p-2}{2} |F+\epsilon|^{p-4} \partial_x F \partial_y F + \frac{p}{2} |F+\epsilon|^{p-2} \partial_{xy} F$$

This approach delivers stability and maps efficiently to graphics hardware.

2.5.4. SPATIAL COHERENCE EXTENSIONS

Instead of simply iterating the Newton process, we can use partially-derived results to guide the convergence process. Typically, natural images consist of several patches, which are mostly consistent or only vary slowly. We exploit this property in several ways. We frequently check neighboring-pixel parameters during the Newton procedure for faster convergence and we derive consistent albedo patches to regularize the optimization. Further, we improve depth parameters for pixels with insufficient numbers of observations by normal integration [BJK07].

Specifically, for each pixel, we test if the use of their parameters for neighboring pixels leads to a reduced error (and vice versa), in which case the values are copied over, similar to [AWL15]. This does not affect the optimization in a mathematical way, but is merely used to improve convergence. We test four different parameter-transfer combinations regarding error reduction; with or without using the color parameters, and with or without using the geometric parameters. To exploit albedo consistence, the process is slightly more involved. We observe that albedo changes will exhibit strong gradients in

Algorithm 1 Virtual scene reconstruction algorithm

for each pixel point i **do**
  Initialize $X := (\rho_{ri}, \rho_{gi}, \rho_{bi}, \theta_i, \phi_i, \lambda_i)$
end for
for each pixel point i **do**
  for each iteration **do**
    Update $\omega_{ij}^{\text{hs}} \omega_{ij}^{\text{outl}} \omega_{ij}^{\text{ld}}$
    Compute gradient g and Hessian matrix $\mathscr{H}$.
    $X \leftarrow X - (\mathscr{H} + \epsilon \mathrm{I})^{-1} \cdot g$ (Gaussian elimination)
    if $(\mathbf{n}_i(\theta_i, \phi_i) \cdot \mathbf{a}_i) > 0$ **then**
      $\theta_i = -\theta_i$
    end if
  end for
end for

the median gradient image. In consequence, we define the energy for optimization with albedo constraints as

$$F_i(\mathbf{n}_i, \lambda_i, \rho_\mathbf{i}) = \sum_j \omega_{ij} E_{ij}(\mathbf{n}_i, \lambda_i, \rho_\mathbf{i}) + \gamma \sum_{k \in \mathscr{N}_i} \omega_{ik} A_{ik}(\rho_\mathbf{i})$$

where ω_{ik} is set to 0 or 1 depending on the edge image obtained in Sec. 2.5.2, $\mathscr{N}_i$ is a 3×3 patch centered around pixel i, and $A_{ik}(\rho_\mathbf{i})$ is the albedo difference between a pixel i and a neighboring pixel k:

$$A_{ik}(\rho_\mathbf{i}) := \left\| \rho_\mathbf{k} - \rho_\mathbf{i} \right\|^p .$$

Note that, the ℓ_p-norm is again used for measuring the difference. The user parameter γ can be used to control the influence of the regularization (increasing γ leads to a smoother albedo). Since the value range of the regularizer depends on the light source power, γ should be adjusted accordingly. In our case, we use $\gamma = 0.01$ for all our real-world data sets. For faster convergence, we first solve an initial solution without regularization and use the result as an initialization for the regularized problem.

Finally, depth is known to require more observations due to its non-linearity and weaker influence on the error term than the other parameters. In consequence, if noise is present, it first manifests itself in the depth values. In all examples, we recompute the depth of 20% of the pixels having the lowest number of used observations via normal integration [BJK07], using the remaining depth values as constraints. Note that depth and normal are indeed linked: the normal is the cross product of gradients of the depth map in smooth regions. However, the scenes we handle feature many objects, producing depth discontinuities and occlusions. This situation prevents us from robustly recovering the geometry from normal integration alone (which would, additionally, require knowledge of one depth value per smooth region). Our approach estimates both depth and normal based on shading, finds consistencies in the reconstructed data automatically, and detects depth discontinuities otherwise.

2.5.5. LIGHT POSITION OPTIMIZATION

The light and scene estimation are both estimation processes but should lead to a consistent result. In consequence, the light positions obtained in Sec. 2.4 can be refined using the scene reconstruction result $(\rho_{\mathbf{i}}, \mathbf{n}_i, \lambda_i)$ and vice versa. By alternating the two optimization steps, we can refine the solution. The light position of a frame j can be optimized by minimizing the energy

$$L_j(\mathbf{b}_j) = \sum_i \left\| \mathbf{m_{ij}} - \frac{\rho_{\mathbf{i}}\left(\mathbf{n}_i \cdot (\mathbf{b}_j - \lambda_i \mathbf{a}_i)\right)}{\|\mathbf{b}_j - \lambda_i \mathbf{a}_i\|^3} \right\|^p .$$

While the frame j is fixed, the sum iterates over the pixels and we only consider the valid observations used in the scene parameter reconstruction.

Again, we solve the problem using the Newton method (Alg. 2). In the beginning, the light positions of all frames are directly initialized from the light calibration. For each iteration, we compute the light position gradient $\nabla_{\mathbf{b}_j} L_j$ and Hessian matrix $\mathscr{H}_{\mathbf{b}_|\mathbf{b}_|}$. Finally, the light position is updated until a local minimum is reached.

Algorithm 2 Light position refinement algorithm

```
for each frame j do
    Initialize b_j := b_recon
end for
for each frame j do
    for each iteration do
        Compute gradient ∇_{b_j} L_j from valid pixels
        Compute Hessian matrix H_{b_| b_|} from valid pixels
        b_j ← b_j − (H_{b_| b_|} + εI)^{-1} · ∇_{b_j} L_j
    end for
end for
```

2.6. RESULTS

We have implemented our framework in OpenGL/C++ on a desktop computer with an Intel Core i7 3.7 GHz CPU and a GeForce GTX TITAN GPU. The scene parameter reconstruction was implemented in parallel on the GPU, while the light calibration and optimization was implemented on the CPU. In the following, we evaluate our framework on synthetic data sets as well as real-world captures.

2.6.1. EVALUATION ON SYNTHETIC DATASETS

We evaluate our method on synthetic datasets (generated in Blender 2.73 Cycles) enabling a ground-truth comparison. Our first experimental scene MONKEY is a compilation of several objects with different properties: a set of planes, a pyramid and sphere with uniform albedo, and a cube as well as the Blender Suzanne monkey head model with varying albedo from textures. We added five reflective spheres for light calibration and generated different illumination situations for 150 randomly-chosen light positions.

Our second scene KITCHEN is a more complex synthetic indoor scene with several objects of different albedo and shape. Four reflective spheres are placed for light calibration and 199 light positions are chosen following a spiral-like path. The dimensions of scene MONKEY and KITCHEN are about $10 \times 10 \times 8$ and $4 \times 4 \times 3$, respectively.

Sphere Detection We first evaluate our sphere detection method and compare it to the traditional approaches based on circle detection. In Fig. 2.9 (top images), we visualize the projection of the reconstructed spheres for both methods. Overall, our approach is more accurate and detects spheres further away from the image center more robustly. E.g., the spheres marked 1 in the MONKEY scene and marked 2 in the KITCHEN scene are clearly misclassified if perspective distortion is not considered. The increased accuracy of our solution is also evident when comparing the world distances of both approaches to the ground truth (Fig. 2.9, bar plots).

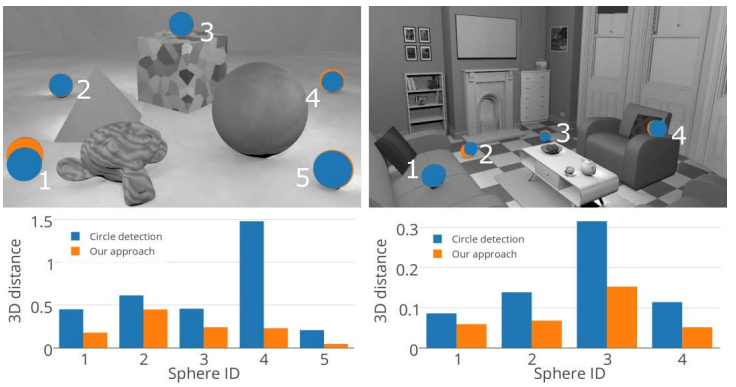


Figure 2.9: Comparison of the 3D distance (error) w.r.t. reference for sphere detection between traditional circle detection (blue) and our method using the cone-based model (orange). The diameter of the spheres in world-space is 0.46 for MONKEY scene ($10 \times 10 \times 8$) and 0.2 for KITCHEN scene ($4 \times 4 \times 3$). It can be seen that our approach accurately detects spheres which are closer to the image border and exhibit perspective distortions.

Light Calibration We compare in Fig. 2.10 our light calibration method to the method, which shoots a ray from the blob center only. Our method results in smaller average error, and, as mentioned in Sec. 2.4, can locate the highlight on sub-pixel level. Furthermore, no parameters are needed to tweak the blob-center detection. On the other hand, we evaluate the robustness of our light-position estimation in Fig. 2.11 for the first 10% of the frames of the MONKEY scene. We show results for the initial light calibration as well as two further optimizations alternating with the scene reconstruction. It can be seen that the light positions are improved for most frames that are not already close to ground truth. For the other frames, which are already estimated well during the initial calibration, only small fluctuations occur.

We investigate the influence of the alternating optimization of the light positions in more detail in Fig. 2.12. The table shows the median angular error for the reconstructed normal map and two insets in the MONKEY scene for up to three light refinement it-

erations. It can be seen that the error is constantly reduced by each iteration. In practice, typically 1-2 iterations are sufficient, which provides a reasonable trade-off between computation time and resulting error.

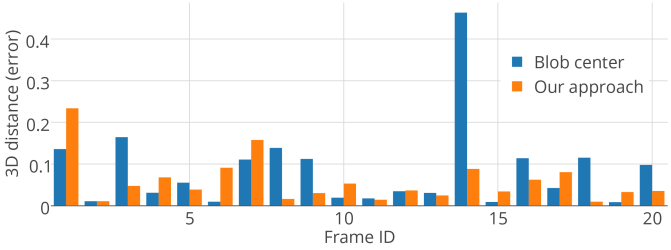


Figure 2.10: 3D error of light position w.r.t. reference for light calibration in MONKEY scene (10 × 10 × 8) using blob centers only (mean error: 0.086) and our approach (mean error: 0.058).

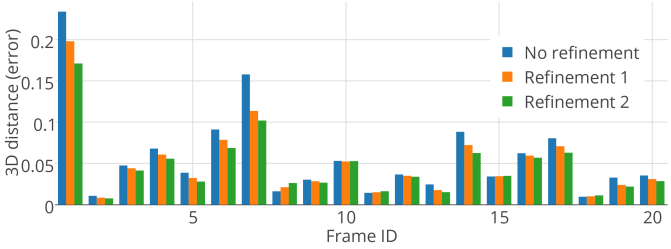


Figure 2.11: 3D error of light position w.r.t. reference for alternating between light optimization and scene reconstruction in MONKEY scene (10 × 10 × 8). It can be seen that for most frames the estimated light position gets more accurate.

	IMAGE	INSET A	INSET B
No refinement	6.0486	6.4845	5.6150
Refinement 1	5.9133	6.2298	5.5788
Refinement 2	5.8445	6.1481	5.4505
Refinement 3	5.8445	6.1481	5.4132
Perfect lights	4.4041	5.3945	4.3347

Figure 2.12: Median angular error (in degrees) for the full normal map and two selected regions (shown on the left) in the MONKEY scene after various light position refinement steps. Overall, the error continuously decreases with each iteration.

Scene Reconstruction We first investigate different values of p for the ℓ_p-norm minimizer. The result is shown in Fig. 2.13. The ℓ_p-norm minimizer ($p <= 1.0$) converges better and also faster than the ℓ_2-norm minimizer as it can be seen in Fig. 2.14. The convergence of the energy error, normal, and depth during the optimization is illustrated in Fig. 2.15.

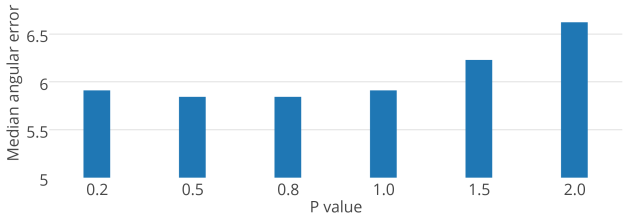


Figure 2.13: Median angular error (in degrees) using different values of p for the ℓ_p-norm minimizer. Using an ℓ_p-norm minimizer ($p <= 1.0$) achieves smaller errors.

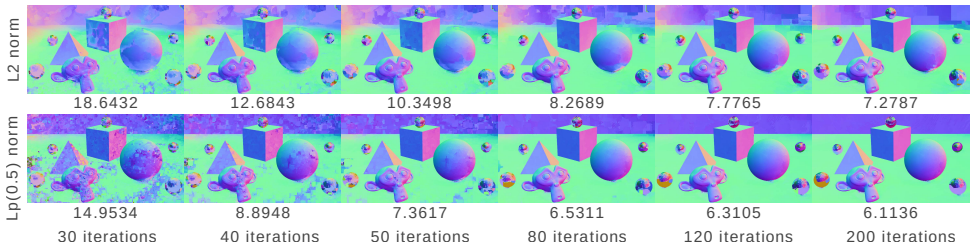


Figure 2.14: Comparing the convergence of the median angular error (in degrees) using an ℓ_2-norm and ℓ_p-norm minimizer. The use of ℓ_p-norm minimizer allows for faster and better convergence.

Fig. 2.16 shows the reconstructed scene parameters (normal, albedo, and depth) for the MONKEY and the KITCHEN scene. Our method achieves accurate results with small median angular errors and rel. median depth errors (w.r.t. the maximum z-extent of the scene) after around 100 iterations. A single iteration in the MONKEY scene (150 frames, resolution of 960×540) requires 2.22 seconds, and 2.34 seconds in the KITCHEN scene (199 frames, resolution of 720×405). Our approach scales linearly with respect to the resolution as well as the number of frames of the video.

We compare our method with the near-light PS algorithms from Ahmad et al. [ASSS14] and Mecca et al. [MQ*16]. The first approach computes object distances from local diffused maxima regions from which they derive the per-pixel light vectors. However, they rely on the assumption that all objects of interest are roughly in the same distance plane,

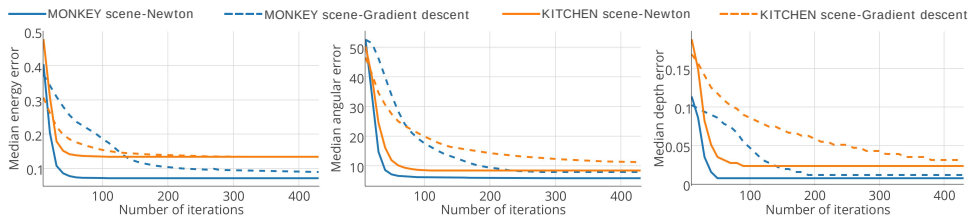


Figure 2.15: Convergence of median energy value, median angular error (in degrees), and median depth error using gradient descent and Newton method for both synthetic scenes. Our modified Newton method provides faster and better convergence than the gradient descent.

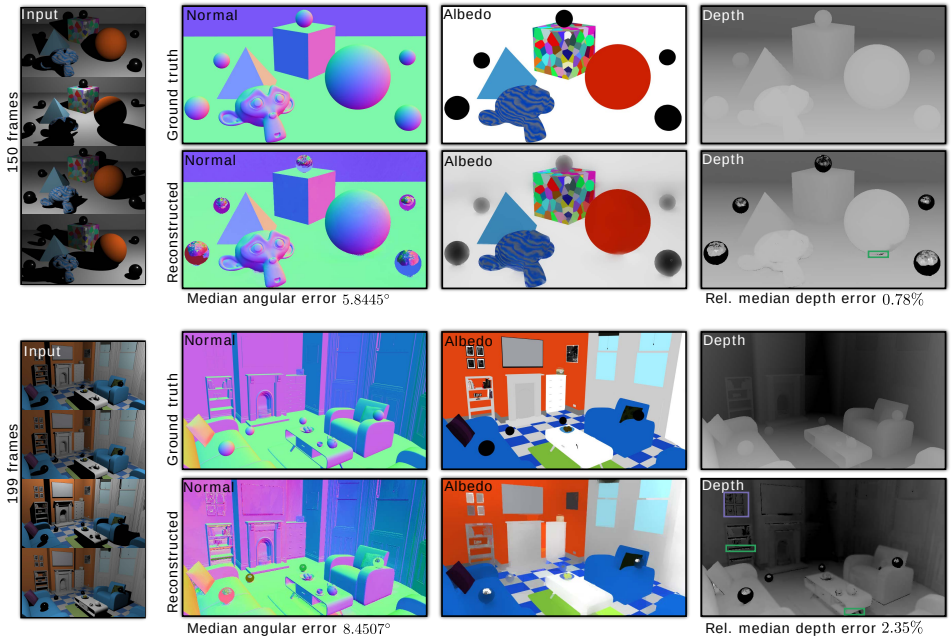


Figure 2.16: Scene parameters reconstructions of two synthetic data sets with dimensions of about $10 \times 10 \times 8$ (top) and $4 \times 4 \times 3$ (bottom) using our approach comparing against ground truth: Our approach recovers the normal map, albedo (relative to light source) map, and absolute depth map of a given scene simultaneously. Overall, we achieve a low median angular error and rel. median depth errors (w.r.t. the maximum z-extent of the scene). Smaller artifacts can occur from insufficient observations (green dashed areas) and surfaces with almost black albedo (purple dashed areas).

which does not hold for larger scenes with large depth discontinuities, as the ones we address. This limitation results in artifacts for our test scenes (Fig. 2.17, first image). The approach from Mecca et al. is more closely related to our approach and formulates the near-light PS problem globally. However, they do not consider shadows, leading to unpleasant artifacts in regions, which are partially shadowed over the video sequence (Fig. 2.17, second image). In comparison, our method can achieve a robust scene reconstructions in the presence of large discontinuities and shadowed regions (Fig. 2.17, third image.)

To illustrate robustness against noise, we generate another three data sets by contaminating the input frames with different levels of additive Gaussian white noise with zero mean. The used standard deviations are 0.01, 0.02 and 0.04 respectively. The results in Fig. 2.18 (shown exemplarily for the normal map) illustrate that our method can ensure robust reconstruction with small median angular errors even for noise levels, which are typical of low-cost camera systems.

We also evaluate the influence of constraining the albedo values. As shown in Fig. 2.19, the constrained optimization outperforms the unconstrained one for the textured objects in the MONKEY and leads to an overall smoother albedo appearance.

Our approach is not without limitations, but in the virtual data set, reconstruction

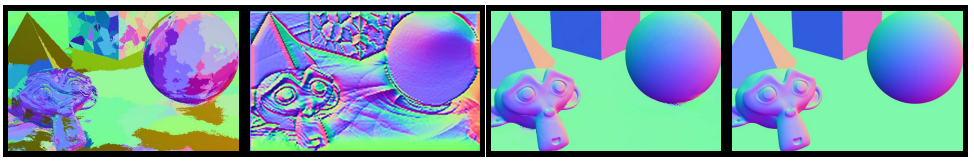


Figure 2.17: Comparison (normal map) of our approach with the two state-of-the-art near-light PS algorithms from Ahmad et al. [ASSS14] and Mecca et al. [MQ*16] for the MONKEY scene. Please note that shadows and discontinuities in our input makes the data already unsuitable for these algorithms, hence it is obvious that their reconstructions fail for most parts.

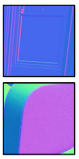


	IMAGE	INSET A	INSET B
$\sigma = 0.0$	8.4507	8.8606	5.1056
$\sigma = 0.01$	8.4627	9.1311	5.1056
$\sigma = 0.02$	8.6405	9.3830	5.5425
$\sigma = 0.04$	8.7916	9.5012	6.6085

Figure 2.18: Median angular error (in degrees) for the full normal map and two selected regions (shown on the left) in the KITCHEN scene for different levels of artificially additive Gaussian white noise. Even when the input is corrupted with strong noise our approach faithfully reconstructs the scene parameters.

failures mostly arose from an insufficient number of observations. Parts like the bottom of the sphere and a part of the monkey head's ear, as illustrated in the green dashed area in Fig. 2.16, are problematic because of being almost always in shadow. In a real-world scenario, it implies that a user should take special care to exhibit all parts of the scene to the light source well enough to avoid reconstruction issues. Additionally, objects with black or very dark albedo need special treatment or can otherwise introduce localized artifacts in the reconstruction (Fig. 2.16, purple dashed area).

2.6.2. EVALUATION ON REAL-WORLD SCENE DATASET

We reconstructed five real-world scenes including a complex and large-scale office scene ($2 \times 2 \times 2$ meter) and four small scale object scenes with different shapes and colors using

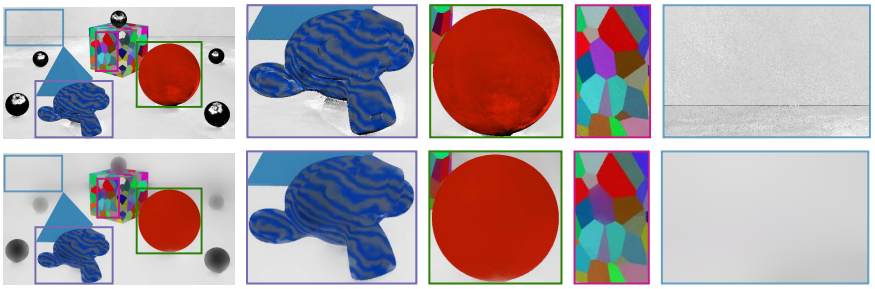


Figure 2.19: Comparing the reconstructed albedo without (**top row**) and with (**bottom row**) smoothness constraints. Optimizing with constraints leads to overall smoother albedo results.

our framework. Since our goal was to support a cheap and minimal capturing setup, we used four customary Christmas balls of radius 5.0 cm (big) or 2.0 cm (small) including clear imperfections as our reference spheres. For the light source, we used a hand-held standard light bulb attached to a stick to facilitate the light movement as illustrated in Fig 4.2. After setting up the scene, we recorded a video using a Cannon 5D II camera, while the user walked around in the scene moving the light source arbitrarily. Fig. 2.20 demonstrates (for the office scene) that the light positions can be arbitrarily distributed, which makes the data capture convenient for the user. No post-processing was applied to the captured video before reconstruction and our framework automatically handles frames where the light source and/or the user accidentally appear.

Fig. 2.21 illustrates the robust reconstruction of an office scene using our approach. It is worth noting again that artifacts occur mainly in areas with black material, such as the black adjusting handle of the chair (blue dashed area), the parts that the light hardly illuminates, e.g., the background behind the computer and the area behind the chair (red dashed area), and the regions with interreflections, such as the camera-facing side of the cardboard. More results are provided in Fig. 2.22, showing that our approach is able to truthfully recover all scene parameters of the tested real-world scenes. Please note that the scenes in Fig. 2.22 contain strong depth discontinuities. These discontinuities might be less visible in the 3D rendering, as the rendering process we chose considers a height field.

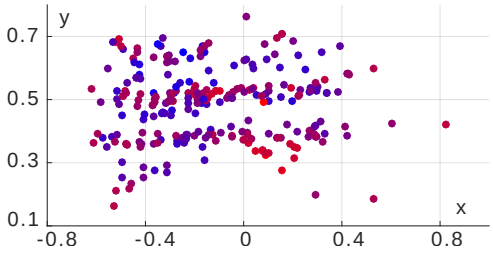


Figure 2.20: Visualization (in world-space coordinates centered at the camera position) of the captured light positions in the office scene. We map the depth minimum and maximum value from blue to red. Our capture setup allows the user to move the light source arbitrarily.

Figure 2.21: Reconstruction results of the scene parameters for a complex, large scale real-world data set of an office work space. While a few artifacts occur in areas which are barely lit (red), feature very dark materials (blue) or interreflection (the camera-facing side of the cardboard), most parts are truthfully reconstructed by our method.

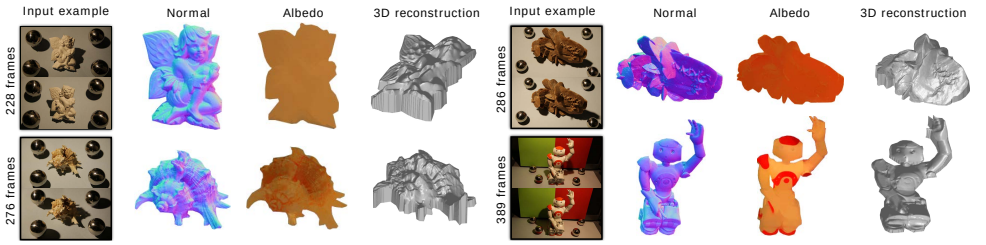


Figure 2.22: **Near-light photometric stereo results** of our approach on various real-world datasets. Even with a minimalistic setup our framework reconstructs the normal, albedo, and depth scene parameters truthfully.

2.7. CONCLUSION

We presented a framework for indoor scene reconstruction that solves the near-light PS problem from a set of video frames exhibiting multiple illumination conditions. The capturing setup is cheap and convenient for users, and only depends on a few uncalibrated reflective spheres. We proposed a novel light calibration approach that uses a cone-based Hough transform to find the spheres in the scene and triangulates the light position accurately via a trimmed least-squares approach. A benefit of our light calibration is that it can handle irregular highlights as well as inter-reflections between reference spheres, which both occur frequently in real-world scenarios. We introduced an ℓ_p-minimizer and reweighting scheme to robustly reconstruct the scene's normal, albedo, and depth parameters in an optimization framework based on the near-light PS model. Our method was demonstrated on both synthetic and real-world datasets. Hereby, we demonstrated that our method is able to handle perspective projection, noise, and albedo variations. Our approach shows that near-light photometric stereo is a feasible option for scene reconstruction.

Several interesting extensions could be investigated in the future. The temporal consistency of the light movement could be exploited during the light calibration. Further, the placement and number of reference spheres is an interesting problem. Nonetheless, our approach does integrate multiple spheres robustly and handles outliers carefully, making a precise placement less crucial.

II

In this part of the dissertation, we present a tool related to the second solution of depth map derivation: depth map design via user guidance. By providing several interactive tools for the users described in Chapter 3, we allow users more freedom to manipulate the depth design process and the possibility to control the result with respect to a desired depth-based effect, which is shown by a connection to a direct image depth perception enhancement application, including wiggle stereoscopy and unsharp masking.

3

Depth Annotations: Designing Depth of a Single Image for Depth-based Effects

We present a novel pipeline to generate a depth map from a single image that can be used as input for a variety of artistic depth-based effects. In such a context, the depth maps do not have to be perfect but are rather designed with respect to a desired result. Consequently, our solution centers around user interaction and relies on scribble-based depth editing. The annotations can be sparse, as the depth map is generated by a diffusion process, which is guided by image features. We support a variety of controls, such as a non-linear depth mapping, a steering mechanism for the diffusion (e.g., directionality, emphasis, or reduction of the influence of image cues), and besides absolute, we also support relative depth indications. In case that a depth estimate is available from an automatic solution, we illustrate how this information can be integrated in form of a depth palette, that allows the user to transfer depth values via a painting metaphor. We demonstrate a variety of artistic 3D results, including wiggle stereoscopy, artistic abstractions, haze, unsharp masking, and depth of field.

This chapter is based on the following publications: **Jingtang Liao**, Shuheng Shen and Elmar Eisemann, "Depth Map Design and Depth-based Effects With a Single Image", *Graphics Interface, pages 57–64*, 2017; **Jingtang Liao**, Shuheng Shen and Elmar Eisemann, "Depth Annotations: Designing Depth of a Single Image for Depth-based Effects", *Computers & Graphics*, submitted.

3.1. INTRODUCTION

Representing 3D content on a standard 2D display is difficult. This topic has been of much interest to artists, who learned over centuries how to use effective pictorial cues to enhance depth perception on a canvas. On a computer display, it is also possible to add animation for the purpose of an increased depth perception. The *Ken Burns effect* is a simple example that combines zooming and panning effects and is widely used in screen savers. For television and movie productions, this technique can be obtained by a rostrum camera to animate a still picture or object. In its modern variant, the foreground is often separated from the background, which requires a rudimentary segmentation. The resulting parallax effect leads to a strong depth cue, when the viewpoint is changing (Fig. 3.1). Today, with the help of image-manipulation software, such effects can be easily produced. However, the picture elements are only translated, which is very restrictive and leads to a reduced effectiveness.

When several views are available, image-based view interpolation [GGSC96] is more general. The perceived motion of the objects helps in estimating spatial relationships. Nonetheless, these techniques often require a special acquisition setup or a carefully produced input. Wiggle stereoscopy can be seen as a particular case of view interpolation, which simply loops left and right images of a stereo pair and can result in a striking parallax perception despite its simplicity (Fig. 3.2). These techniques all avoid special equipment, e.g., 3D glasses, and they even work for people with limited or no vision in one eye. Alternatively, it is possible to use a single input image and warp it based on a depth map to produce stereo pairs. Yet, computing depth maps for a monocular image is an ill-posed problem. While important advances have been made [EPF14, LBRF12, SCN05, SSN09], the methods are not failsafe. Furthermore, many depth-based effects require the possibility for manual adjustments, such as remapping the disparity range of stereoscopic images and video in production, live broadcast, and consumption of 3D content [LHW*10], or to modify a depth-of-field effect in an artistic manner [LES10], which is why we focus on a semi-automatic solution. We will show that a depth estimate, if available, can be beneficial as a starting point for our interactive depth-map design.

In this chapter, we propose a new framework to generate a depth map for a single

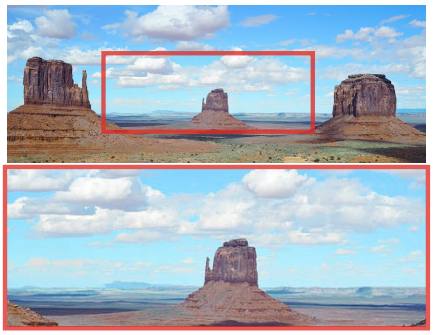


Figure 3.1: Ken Burns effect. Panning and zooming on still images (Image source: `http://maxpixel.freegreatpicture.com`).

Figure 3.2: Wiggle stereoscopy. Looping a left/right image pair (Image source: Wikimedia Commons).

input image with the goal of supporting artistic depth-based effects to illustrate the spatial information in the image. We build upon the insight that a depth map does not have to be perfect for such applications but should be easily adjustable by a user, as this option allows fine-tuning of the artistic effect. Our results are illustrated with a variety of examples, ranging from depth-of-field focus control to wiggle stereoscopy. Additionally, with such a depth map at hand, it is possible to produce image pairs for 3D viewing without (e.g., via establishing a cross-eyed view) or with specialized equipment (e.g., stereo glasses). Our approach builds upon the assumption that depth varies mostly smoothly over surfaces and only exhibits discontinuities where image gradients also tend to be large. In consequence, we follow previous work and require only coarse annotations, such as sparse scribbles [GDA*11, LTW14, WLF*11] or points [LGG14]. These annotations form hard constraints in an optimization system that leads to a diffusion process, taking the image content into account. We focus on the control of this process and our method offers ways to influence the result via local and global constraints. Defining relative depth differences, a non-linear depth diffusion by assigning a strength to scribbles, or privileged diffusion directions are examples. We ensure that all these elements can be formulated in a linear optimization problem to ensure a fast solving step. We additionally show a selection of effects in our results.

Overall, our work makes the following contributions:

- A fast depth-map creation solution from a single image;
- Various tools to refine the depth map;
- A depth design tool, in form of a depth palette if an estimated depth map is available;
- A selection of effective effects, including wiggle stereography, unsharp masking, haze, or artistic abstractions.

Furthermore, we describe interface decisions to ease the creation of the depth map and facilitate the choice of adequate depth values.

3.2. RELATED WORK

Depth perception helps us perceive the world in 3D using various depth cues, classified into binocular and monocular cues. In an image, we typically encounter monocular cues — depth information that can be perceived with just one eye. Motion parallax [KDR*16], size, texture gradient [BL76], contrast, perspective, occlusion [LEE17], and shadows [BG07] are examples of these. Motion parallax and occlusion are particularly strong [Cut95]. Parallax arises due to the non-linear displacement relative to the depth when shifting the viewpoint of a perspective projection. In order to add such an effect, one can warp an image based on a depth map, which associates to each pixel the distance to the camera.

Depth estimation for a single image is a well-known problem in computer graphics and computer vision that received much attention. Recent approaches [EPF14, LBRF12, SCN05, SSN09, LSL15, KLK14] are based on learning techniques. They enable an automatic conversion from a photo to a depth map. Nonetheless, the quality depends on the variety of the training data set and provided ground-truth exemplars. Additionally, in practice some manual segmentation is needed and the methods are not failsafe, as problematic elements are quite common (e.g., the reflections in a mirror or a flat image hanging on the wall). Even if accurate depth is obtainable, it is not always optimal for artistic purposes [LHW*10, DRE*11], which is our focus.

Depth from defocus (DFD) is another approach where the amount of blur in different areas of a captured image is utilized to estimate the depth [Pen87]. Methods for single DFD from conventional aperture are usually based on such assumptions. Aslantas et al. [Asl07] assumed defocus blur to be the convolution of a sharp image with a 2D Gaussian function whose spread parameter is related to the object depth. Lin et al. [LJXD13] designed aperture filters based on texture sharpness. Zhu et al. [ZCSM13] took smoothness and color edge information into consideration to generate a coherent blur map for each pixel. Shi et al. [STXJ15] inferred depth information from photos by proposing a non-parametric matching prior with their constructed edgelet dataset, based on small small-scale defocus blur inherent in an optical lens. Their method is limited to photos in their original resolution and does not resolve ambiguities due to smooth edges. A general disadvantage of single-image DFD methods is that they cannot distinguish between defocus in front and behind the focal plane. Coded-aperture setups [LFDF07] address this issue by using a specially-designed aperture filter in the camera. Sellent et al. [SF14] proposed an asymmetric aperture, which results in unique blurs for all distances from the camera. All these coded latter methods require camera modifications and have limitations regarding precision and image quality.

In our approach, the depth map will be designed by the user in a semi-automatic way. Hereby, also artistic modifications are kept possible. Early interactive techniques [CRZ00, LCZ99], and their extensions [LHK09], focused on scenes containing objects with straight edges to reconstruct a 3D model by geometric reasoning and finding the best fitting model to line segments. In general, the use of edges is a good choice, as many natural scenes consist of smooth patches separated by object boundaries. Gerrits et al. [GDA*11] introduced a stroke-based user iterative framework in which users can draw a few sparse strokes to indicate depths as well as normals. Their technique optimizes for a smooth depth map in an edge-aware fashion, which is typically applied to photographs contain-

ing large planar geometry. Lin et al. [LTW14] focused mainly on recovering depth maps for 2D paintings, where the 2D paintings have to be segmented into areas based on input strokes and the depth values are only propagated locally based on the color difference. Wang et al. [WLF*11] proposed a work flow for stereoscopic 2D to 3D conversion, where users draw only a few sparse scribbles, which together with an edge image (computed from the input image) propagate the depth smoothly, while producing discontinuities at edges. Similarly, Lopez et al. [LGG14] used points instead of scribbles to indicate depths and made additional definitions available for the user, such as depth equalities and inequalities, as well as perspective indications. Tools for the definition of equalities and inequalities [YSHSH13, SSJ*10] can help reduce the amount of user intervention. Our work follows similar principles, but offers additional possibilities with the goal of a direct application to artistic depth-based effects. Our work builds upon depth propagation via a diffusion process, similar to diffusion curves[OBW*08] and their extensions [BEDT10].

3.3. OUR APPROACH

Our approach is illustrated in Fig. 3.3. Given a single image as input, e.g., a photograph or even a drawing, we seek to create a depth map and show how it can be used as input to various depth-based effects. Consequently, we first describe the depth-map generation via the diffusion process, then discuss additional tools provided to the user (Sec. 3.3.1), before illustrating our implementation of various depth-based effects (Sec. 3.3.2). Finally, we discuss the results (Sec. 3.4) before concluding (Sec. 3.5).

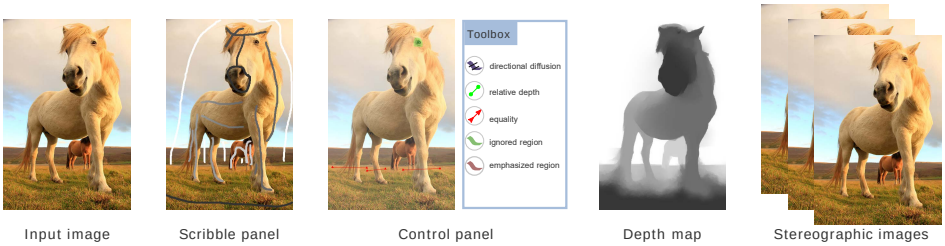


Figure 3.3: Overview: From left to right, starting from a monocular image, the user draws scribbles, which spread via a diffusion process to define a resulting depth map. The interface allows for constant or gradient-color scribbles, the definition of a diffusion strength, brushes to ignore or emphasize gradients in regions or Bézier curves to direct the diffusion process. Further, relative depth differences and equalities can be annotated. (Image source: ©Robert Postma/Design Pics), used with permission.

3.3.1. DEPTH MAP ESTIMATION

The basic input by the user are a few depth indications in form of scribbles. These scribbles will be considered hard constraints that should be present in the final depth map. The rest of the depth map will be solved via an optimization procedure. In order to ensure acceptable performance, we cast our problem into a constrained linear system. This initial setup is identical to *Diffusion Curves* [OBW*08], based on Poisson diffusion [PGB03], except the scribbles take the role of the diffusion curves.

POISSON DIFFUSION

Given the image $I := \{I_{i,j} \mid i \in 1...w, j \in 1...h\}$, where $I_{i,j}$ are brightness or color values at pixel (i, j), we aim at creating a depth map $D := \{D_{i,j} \mid i \in 1...w, j \in 1...h\}$, given a set of scribbles with associated values $\{S_{i,j} \mid i \in 1...w, j \in 1...h\}$ on these scribbles. The depth map D is then implicitly defined:

$$\Delta D = 0$$
$$\text{subject to: } D_{i,j} = S_{i,j}, \forall (i, j) \in I.$$

where Δ is the Laplace operator. The discretized version for a pixel (i, j) of the first equation is:

$$4D_{i,j} - D_{i+1,j} - D_{i-1,j} - D_{i,j+1} - D_{i,j-1} = 0 \tag{3.1}$$

The depth map can, thus, be constructed by solving a constrained linear system. A result is shown in Fig. 3.4 (middle). It can be seen that the colors on scribbles smoothly diffuse across the whole image. The absolute depth values defined by the scribbles are useful to roughly associate depth ranges to different objects or parts of the scene. This is common in practice [Men09] where a coarse layout of the scene depth is defined in the preprocess of the 3D design.

Figure 3.4: Depth estimation from scribbles. Scribble input (left), only using the scribble input results in a smooth depth map lacking discontinuities (middle), by involving the input image gradients, the depth propagation is improved (right). (Image source: www.pixabay.com)

ANISOTROPIC DIFFUSION

Eq. 3.1 implies that each pixel's depth is related to its four neighbor pixels in an equal way. Consequently, the map is smooth and free of discontinuities. Nonetheless, discontinuities can be crucial for depth effects at object boundaries. Hence, we want to involve the image gradients in the guidance of the diffusion process and, basically, stop the diffusion at object boundaries [PM90]. To this extent, we will rely on the difference of neighboring input-image pixels to steer the diffusion, transforming the Laplace equation into a set of constraints. For a pixel k and its 4-pixel neighborhood $N(k)$, we obtain:

$$\sum_{l \in N(k)} \omega_{kl}(D_k - D_l) = 0, \tag{3.2}$$

where ω_{kl} is the first order difference for the two neighboring pixels $\omega_{kl} = \exp(-\beta|I_k - I_l|)$. At the border of an object, ω_{kl} is often close to 0 because the pixel values typically differ. In consequence, the impact of the constraint is reduced, which, in turn, relaxes the smoothness condition. Hence, depth discontinuities will start to occur at boundaries. Fig. 3.4 (right) shows the effect of integrating the image gradient.

Ignored-gradient Region While object boundaries are useful barriers for the diffusion, some gradients (e. g., shadows, reflections etc.) in the image may introduce unwanted depth discontinuities. For example, Fig. 3.5(top row) exhibits shadowed areas, which produce strong gradients that lead to artifacts on the floor, although it should actually have been smooth. For automated methods [LSL15], a user might also want to tweak the resulting depth map. For example, reflections from a mirror in Fig. 3.5(bottom row) might lead to artifacts, which can be addressed with an interactively designed depth map. To this extent, we provide the user with the possibility to use a simple brush to annotate regions where gradients should be ignored. For pixels which were selected in this way, the corresponding diffusion constraint would change back to Eq. 3.1. Fig. 3.5 shows a comparison with and without this annotation.

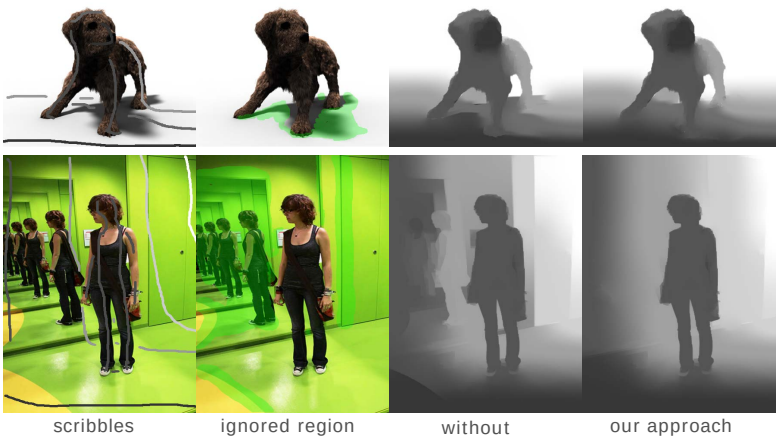


Figure 3.5: Ignored-gradient region. Shadows and reflections introduce unwanted large gradients, which hinder the depth diffusion and lead to discontinuities. Using the ignored-gradient region brush, these gradients can be excluded from the depth derivation. (Top image: courtesy of Erik Sintorn; bottom image: Flickr - salendron)

Emphasized-gradient Region Contrary to the previous case, depth discontinuities might also need a boost in other areas. Consequently, we also allow the user to emphasize gradients. The gradient of the brushed pixels is enlarged by a scale factor (two in all examples). This tool is of great use when refining depth maps (Fig. 3.6), as it helps to involve even subtle gradients when needed. As illustrated in Fig. 3.6, there is no clear boundary at the highlighted (red and blue rectangles) locations. With this tool, the depth discontinuities at these image areas could be well pronounced.

Directional Guidance While the previous methods stop or accelerate diffusion, its directionality remains unaffected. Still, in some cases, the intended diffusion direction might be relatively clear, e.g., along a winding road to the horizon. In order to integrate a directional diffusion in the linear equation system, we let the user provide a directional vector field and remove the gradient constraints orthogonal to the indicated direction,

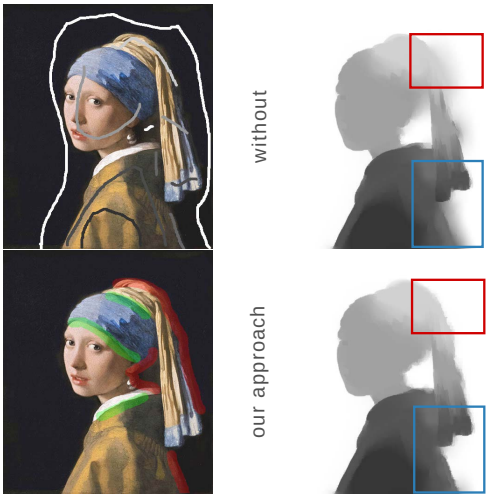


Figure 3.6: Emphasized-gradient region. Weak gradients can be enhanced to induce depth discontinuities. Here, it ensures a better separation between the foreground and background. (Image source: "Girl with a Pearl Earring" by Johannes Vermeer)

following [BEDT10]. For an arbitrary direction $\boldsymbol{d} := (\cos\theta, \sin\theta)$, the derivative of an image I along direction $\boldsymbol{d}$ is given by $\nabla I \boldsymbol{d}$. In consequence, the constraints for pixel (i, j) are replaced by:

$$\cos\theta \cdot \omega_{ijx}(D_{i+1,j} - D_{i,j}) - \sin\theta \cdot \omega_{ijy}(D_{i,j+1} - D_{i,j}) = 0 \tag{3.3}$$

where $\omega_{ijx} = \exp(-\beta|D_{i+1,j} - D_{i,j}|)$ and $\omega_{ijy} = \exp(-\beta|D_{i,j+1} - D_{i,j}|)$. Here, the diffusion will then only occur along direction $\boldsymbol{d}$.

To define the vector field, we first ask the user to indicate the region, where to apply the directional guidance with a brush. To specify the directions, the user can then draw Bézier curves. The tangent of a point on the curve is defining the diffusion orientation that is to be used for the underlying pixel. To propagate the information from the Bézier curves to the entire region, we let the direction vector itself be diffused over the marked region using Eq. 3.1. To avoid singularities, we diffuse the cosine and sine values of the direction and normalize the result after diffusion. Fig. 3.7 (middle, top) shows the curves and brushed region in which the diffusion is guided, as well as the diffused direction information for each pixel of the region (Fig. 3.7 (right,top)).

It is possible to reduce the directionality by adding a constraint for the direction orthogonal to the diffusion direction (i.e., $\boldsymbol{d} := (-\sin\theta, \cos\theta)$). If we do not apply a scale factor to the constraint, the resulting diffusion would go back to a uniform diffusion. The scale factor could be chosen by the user, but we also propose a default behavior based on the image content. The idea is that the user indicates a direction because it is connected to the input image's content. We thus analyze the input image's gradient, and compute the angle θ between gradient and provided diffusion direction to derive an adaptive scale factor $1 - |\cos\theta|$.

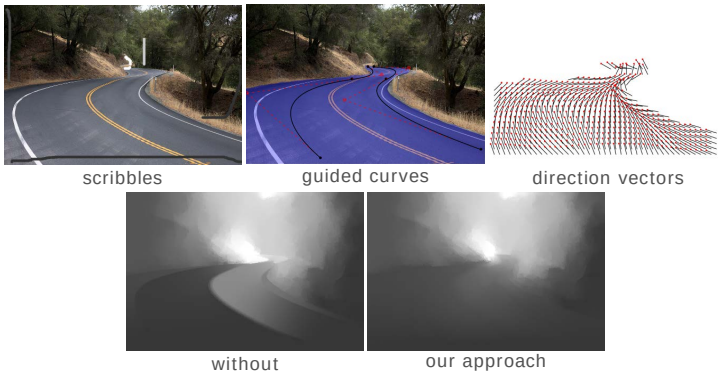


Figure 3.7: Diffusion guidance. A user brushes the region and draws the direct curves to define the direction in which he or she is interested in. Our approach can direct the diffusion mainly happens in this direction. (Image source: `http://maxpixel.freegreatpicture.com`)

DEPTH DIFFUSION STRENGTH

Perspective projection can result in a non-linear depth mapping, e.g., via foreshortening. Furthermore, surfaces might not always be mostly planar but exhibit a convex or concave bent surface. For these situations, we want to provide the user with a way to influence the diffusion strength. Following [BEDT10], diffusion strength can be added by introducing an additional component to the vector value that is diffused; besides a depth value d, we will have a strength α. For two such elements $(d_1, \alpha_1), (d_2, \alpha_2)$, a mix is assumed to yield:

$$\frac{\alpha_1 d_1 + \alpha_2 d_2}{\alpha_1 + \alpha_2}. \tag{3.4}$$

The higher the strength, the higher the influence of the associated depth value on the final result. Fig. 3.8 demonstrates a simple example with two input scribbles, a darker scribble on the left and lighter scribble on the right. We obtain a result where the two values uniformly spread across the image when using equal strength (Fig. 3.8, left). When selecting a bigger influence on the right part by assigning a higher strength to the left scribble its influence on the result is increased (Fig. 3.8, right). This equation directly

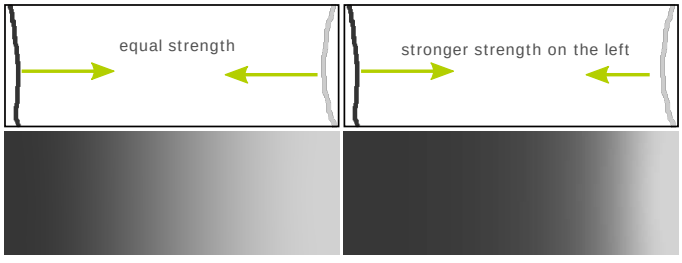


Figure 3.8: Scribble strength. Equal strength (middle); strength of left scribble stronger than the one on the right.

extends to many depth values:

$$\frac{\sum \alpha_i d_i}{\sum \alpha_i} \tag{3.5}$$

This insight makes it possible to formulate this behavior in our linear optimization system — we now solve for two maps, containing values of type αd and α. Once the diffusion converged, we can divide the first map's values by the second, establishing the result of Eq. 3.5. Fig. 3.9 shows the influence of the diffusion strength for different values.

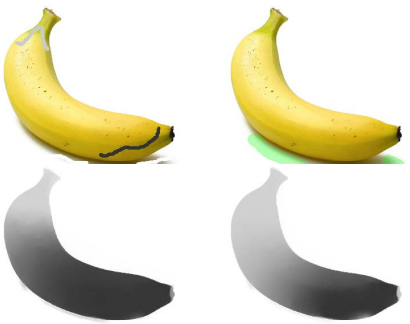


Figure 3.9: Non-linear depth mapping. Assigning a strength to different scribbles can be used to influence the diffusion speed. Green region is the ignored-gradient area.

EQUAL AND RELATIVE DEPTHS

It can be useful to indicate that two objects are located at the same depth, without providing an absolute value. Given our constraint system, this goal can be achieved by adding a constraint of the form $D_k = D_l$, similar to [BEDT10]. This possibility is quite useful for images containing symmetric features, as shown in Fig. 3.10, where pixels on the pillars, which are at the same depth, can be linked. There are also cases in which it may be hard for a user to choose adequate depth values for scribbles. Fig. 3.11 shows an example, in which drawing scribbles with absolute values for each gap inside the wheels would be very difficult, as the correct value depends on the background. With our tool, we can link the background to other regions. It is worth noting that many pixels can be connected at the same time.

We also introduce a new feature to describe relative depth relationships; let D_1, D_2,D_3 and D_4 be four locations in the depth map. If the user wants the distance of D_1 to D_2 equal to the distance of D_3 and D_4, we can add the constraint $D_1 - D_2 = D_3 - D_4$. For the pillar example, the relative depth indications can be used to ensure the equivalent distances between pillars. Again, this solution can be extended to multiple relative points.

DEPTH PALETTE

With the advent of single-image depth estimation, automated approaches [LSL15, KLK14] can provide useful information to initiate the depth map design. Unfortunately, there can be multiple depth inconsistencies or noise, as shown in Fig. 3.12 (top row, left) highlighted (red rectangles) regions. Additionally, the resulting depth might not be adequate for artistic purposes [WLF*11]. Hence, directly using the resulting depth as an input

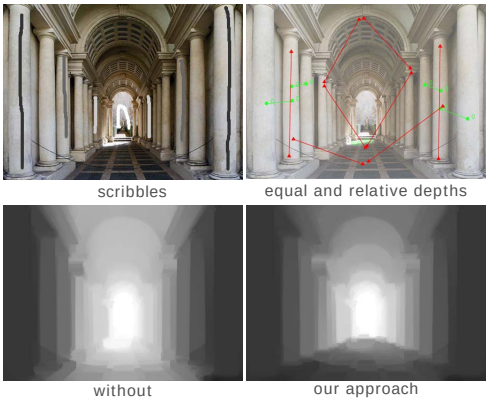


Figure 3.10: Depth equality and relativity We connect depths from different places together via depth equality and relativity to globally influence the depth estimation. (Image source: wikipedia)

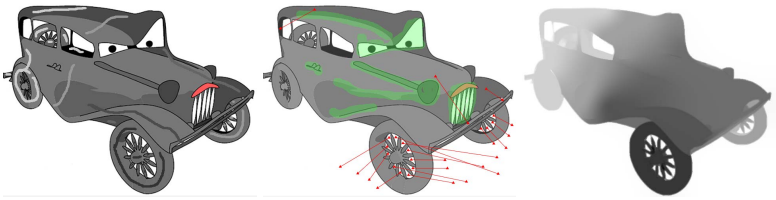


Figure 3.11: Equal constraints. Connecting depth from different places via depth equality can reduce the user interventions. (Image source: [EPD09])

for the 3D effects, e. g., wiggle stereography, could cause visible artifacts (please refer to supplementary video[1]). However, the initial depth maps can serve as a good starting point for the depth-map design. Similar to a color or normal palette, a user can transfer depth values directly from the reconstruction. For this purpose, a position is chosen in the reference depth image. The selected depth value can then be used to draw a scribble with the corresponding value, which will generate a corresponding hard constraint. While drawing the scribble, the value can either be held constant or the values of the corresponding pixels from the reference could be transferred. With only a few depth transfers, it is possible to improve the depth-map quality using our solution.

ADDITIONAL INTERFACE ELEMENTS

Our framework offers the possibility to globally adjust the resulting depth map. We provide the user with a mapping curve, similar to a gamma curve, to specify a non-linear remapping. We use an interpolating spline, adjusted via control points. A result is illustrated in Fig. 3.13 (left), where the depth appearance of the scene is globally influenced to obtain Fig. 3.10. Global adjustments are particularly useful for stereo-based effects, as they allow the user to influence the global disparity range. In this context, we provide a simple user interaction to control the 3D effect on the canvas. Instead of defining the

[1] The video can be accessed at `https://graphics.tudelft.nl/Publications-new/2017/LSE17/`

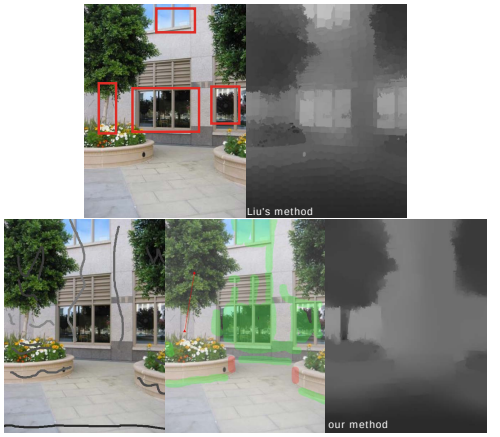


Figure 3.12: Depth palette. Using the result of automated methods [LSL15] as depth palette can ease the depth creation.

stroke values by choosing from a palette, the user can also simply drag the mouse to indicate a disparity baseline that then corresponds to a depth value that is automatically transferred to the stroke. This process makes it easy to control warping effects, in case the depth map is used to derive a stereo pair. Please also refer to supplementary video.

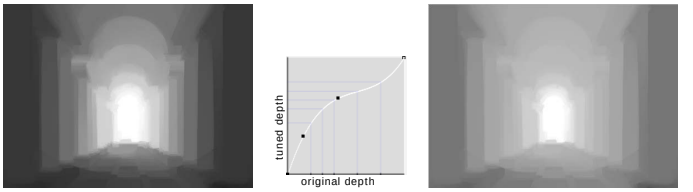


Figure 3.13: Depth adjustment. Depth map can be globally adjusted using a mapping curve.

3.3.2. 3D EFFECTS

In this section, we illustrate a few of the 3D effects that can be introduced in the input image, when relying on the derived depth map, whose values we assume normalized between zero and one.

COLOR-BASED DEPTH CUES

Given the depth map, we can easily add an aerial perspective to the result. An easy solution is to apply a desaturation depending on the distance as shown in Fig. 3.14. Alternatively, we can convert the distance to a fog density and apply it as an overlay on the image [Wil87].

DEPTH-OF-FIELD EFFECTS

It is possible to simulate lens blur to refocus on different parts of the scene. Fig. 3.15 (right) shows an example.

Figure 3.14: Distance-based desaturation and haze.

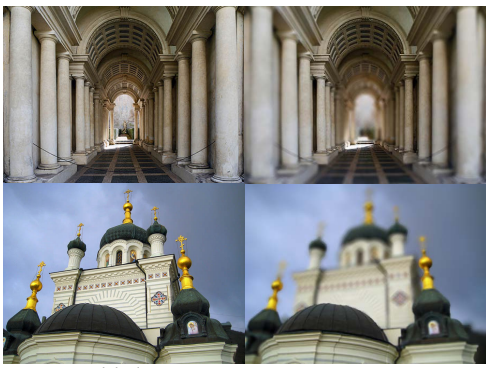


Figure 3.15: Image refocusing based on the depth values.

UNSHARP MASKING

Textures and colors in images can be enhanced by utilizing unsharp masks. Contrary to what its name may indicate, unsharp masks are used to sharpen the images. An unsharp mask is created by subtracting a low-pass filtered (usually Gaussian filter) copy from the original image. The mask is then added back to the original image to get a local contrast enhancement. While typically applied for color images, the involvement of depth enables us to well separate different elements from each other [LCD06]. Note that even when colors are similar (color of the puppy's hair and the background), involving the depth map makes sure that the depth difference becomes more evident.

Luft et al. [LCD06] proposed a depth-based unsharp-masking method. Assuming that a depth map D is available, the unsharp-masking process is applied to the depth buffer: $\Delta D = G * D - D$, with $G * D$ is the convolution of a Gaussian filter. The resulting high frequency ΔD is then used to alter the original image I to achieve a sharpening or a local contrast enhancement: $I' = I + \Delta D \cdot \lambda$, where λ is a user defined gain parameter. Thus, the greater the spatial difference, the higher the local enhancement.

We found that in some cases, distant elements receive an overly strong enhancement. In consequence, we propose an adaptive gain value and Gaussian kernel size. Based on the observation in [RSI*08] that unsharp masking can be performed in 3D instead of image space, we propose a hybrid approach. We adapt the kernel size depending on the depth map values, i. e., the farther away, the smaller the kernel size. Specifically, we define the kernel size as: $\delta_{Adapt} = \delta(1-0.5D)$, with δ being 2% of the image diagonal. Moreover, we apply a bilateral filter instead of a Gaussian filter, to ensure that elements from different depths do not mix and, hereby, keep the contrast of edges. To avoid oversaturation, all operations are executed in CIELAB color space.

Figure 3.16: Unsharp masking using a depth buffer. It can enhance the depth arrangement in the scene and make a dull appearance more interesting.

STEREOGRAPHIC IMAGE SEQUENCE

When adding motion parallax to the input image, the resulting images can be used as stereo pairs, for wiggle stereoscopy, or even as an interactive application that can be steered with the mouse position. Please also refer to our supplemental material for looping videos, of which a few frames are shown in Fig. 3.17.

Figure 3.17: Examples of looping videos.

For a given displacement direction γ and a maximum pixel traversal distance S, the newly-derived image N, in which nearer pixels are shifted more strongly than far-away pixels, is given by:

$$N(i + (1.0 - d_{ij})\cos(\gamma)S, j + (1.0 - d_{ij})\sin(\gamma)S) := I(i, j)$$

Unfortunately, the definition of N is imperfect, as several pixels may end up in the same location or holes occur (no pixel projects to this location). The first case can be easily solved; as our motion direction does not affect depth, we can, similar to a depth buffer, keep the reprojected pixel with the smallest depth value. To address holes, we rely on a post-processing step. We search from a hole in N along the opposite direction of γ, until we find the first non-hole pixel. Its value is then copied over to the hole location. Fig. 3.18 shows the comparison with and without hole filling. Note that our hole filling method is not suitable for big motions.

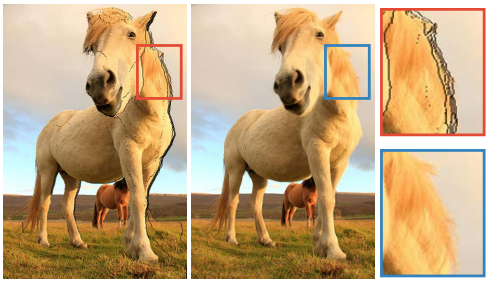


Figure 3.18: Hole filling. Holes due to reprojection (left) are filled (right).

ARTISTIC EFFECTS

Besides changing the viewpoint, the derived depth map can also be used to apply artistic filters. First, we illustrate the use for movement and show a special rotation, where the radius depends on the distance. Second, there are many depth-based abstraction filters and we show an example, based on the work by Jodeus `http://jodeus.tumblr.com/post/131437406357`. Here, discs are used to replace a subset of the pixels to achieve an abstract look (Fig. 3.19). These effects are best illustrated in the accompanying video.

3.4. RESULTS

We have implemented our framework in Java on a desktop computer with an Intel Core i7 3.7 GHz CPU. The linear solver is implemented in Matlab and called from within the Java program. To make the solver more efficient, we build up an image pyramid for the input of the solver and solve each layer from low to high resolution, while using the result of the previous layer as the input for the current layer. It takes about 30 seconds to compute a depth map of 600×500. Nonetheless, we did not optimize our approach and it could be possible to achieve even real-time rates via a GPU implementation. Furthermore, the approach would lend itself well to upsampling strategies. For now, we provide a small-resolution preview to the user, which is interactive.

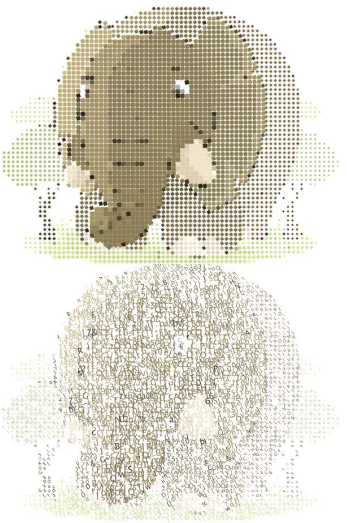


Figure 3.19: Example frames of different depth-based abstractions (i. e., square and letter).

We tested our depth estimation on various datasets (e.g., Fig. 3.20). It works for real photographs, paintings, but also cartoons. All results and all sequences shown in the video have been produced by a user in less than 3 minutes.

We did not conduct a user study to investigate the effectiveness of our tools. In practice, we received positive feedback from three test users. Nonetheless, expertise in image editing is definitely an advantage. This is similar to novice users applying advanced tools in software, such as Photoshop or Gimp. A certain amount of training is also helpful to gain familiarity. Increasing user friendliness further could be an interesting direction for future work.

3.5. CONCLUSION

We presented a pipeline for integrating depth-based effects into a single-image input. We proposed editing tools to facilitate the depth-map creation by influencing a depth-diffusion process. We demonstrated that our solution enables users to generate depth maps very rapidly and presented various examples for depth-based enhancements. In the future, we want to increase performance, which could be achieved via a sparse GPU linear solver.

It would also be interesting to apply our method for animations. One possible solution might be to design depth maps for several key frames and propagating the annotations, similar to rotoscoping [AHSS04].

Figure 3.20: Examples. We support a wide variety of inputs including real photographs, paintings and cartoon images. Image source: from top to bottom, row 1, 2, 3, 5 are from `https://pixabay.com/`; row 4 is from Lone Pine Koala Sanctuary; row 6, 7 are from ©Blender open source movie *Big buck bunny* and *Monkaa*, respectively; row 8 is from [SS03]; row 9 is "Girl with a Pearl Earring" by Johannes Vermeer.

III

We have already discussed a few techniques regarding enhancing depth perception in original images in Chapter 3. In the last part of the dissertation, we take this discussion one step further by presenting a study on split-depth images. It is a little-explored optical illusion to enhance depth perception in a video clip. The goal is to achieve findings on its mechanism and hence find a solution to automatically generate such images, as described in Chapter 4.

4

SPLIT-DEPTH IMAGE GENERATION AND OPTIMIZATION

Split-depth images use an optical illusion, which can enhance the 3D impression of a 2D animation. In split-depth images (also often called split-depth GIFs due to the commonly used file format), static virtual occluders in form of vertical or horizontal bars are added to a video clip, which leads to occlusions that are interpreted by the observer as a depth cue. In this chapter, we study different factors that contribute to the illusion and propose a solution to generate split-depth images for a given RGB + depth image sequence. The presented solution builds upon a motion summarization of the object of interest (OOI) through space and time. It allows us to formulate the bar positioning as an energy-minimization problem, which we solve efficiently. We take a variety of important features into account, such as the changes of the 3D effect due to changes in the motion topology, occlusion, the proximity of bars or the OOI, and scene saliency. We conducted a number of visual perception experiments to derive an appropriate energy formulation. Our method helps in finding optimal positions for the bars and, thus, improves the 3D perception of the original animation. We demonstrate the effectiveness of our approach on a variety of examples. Our study with novice users shows that our approach allows them to quickly create satisfying results even for complex animations.

This chapter is based on the following publication: **Jingtang Liao**, Martin Eisemann and Elmar Eisemann, "Split-Depth Image Generation and Optimization", *Computer Graphics Forum*, vol.36, no.7, pages 175–182, 2017.

4.1. INTRODUCTION

Preserving or even enhancing the 3D impression of a scene on a 2D display can be a powerful means to attract attention, amplify scene layout, and enhance scene understanding, yet, it is difficult to achieve. Artists throughout the centuries developed techniques on how to use effective pictorial (or monocular) cues to enhance the depth perception on a canvas.

Occlusion is one of the strongest cues of the human visual system to interpret depth ordering. In consequence, it is also one of the main factors to exploit when conveying a 3D arrangement. One creative solution to exploit this effect for paintings is the use of a passepartout (a paper, more usually, cardboard sheet with a cutout). We are so accustomed to a passepartourt being not part of the image itself that incorporating it into the actual painting leads to a surprisingly convincing 3D effect, e. g., in "Escaping Criticism" by Pierre Borell del Caso as shown in Figure 4.1. With digital media, virtual passepartouts have become a popular variant for photography and static virtual scenes [SCRS09, RTMS12]. The resulting occlusion effect separates the image into a front and back layer, which produces a strong "popping out" sensation or a "floating on the window" illusion. Split-depth images utilize similar reference spaces, usually bars (but we will use the general term *splits* throughout the chapter), to increase the 3D effect of a short animation or movie clip. They have recently gained in popularity and are employed by an increasing number of companies to catch the consumer's attention and interest. This chapter will present a novel algorithm to help in the generation of split-depth images.

Figure 4.1: The use of reference spaces in paintings, images and animations increases 3D impression. a) "Escaping Criticisim" by Pierre Borell del Caso; b) Out of bounds photography [OOB15]; c) Virtual passepartouts [VP09]; d) Split-depth image.

As for passepartouts, splits induce a plane in the virtual scene, creating a division between the mental fore- and background. If an object overlaps this plane, it is considered in front. The same holds for animations where this information is usually interpreted as an object moving out of the image towards the viewer. Currently, the generation of such split-depth images is a purely manual and time-consuming task relying on image editing tools to segment each image in an animation and add the splits. Further, there

are no known rules for producing the effect and choices were made in an adhoc manner, although the quality of the resulting animations relies heavily on several factors such as position, width, scene content, physical correctness, etc. In consequence, designing split-depth images is a tedious and time-consuming task, which resulted in many low-quality examples on the internet. In our work we propose an approach to automatically create split-depth images using an RGBD (color plus depth) image sequence as input. We investigate the possible factors, which contribute to the enhanced 3D perception and build a computational model to automatically generate splits that lead to a convincing result. Overall, we make the following contributions:

- A perceptual study to investigate the contributions of different factors to the split-depth illusion;
- A multi-objective split-optimization procedure respecting various perceptual cues, such as occlusion, split proximity, and scene saliency;
- A framework to support the split-depth image generation.

4.2. RELATED WORK

In this section, we will briefly discuss the related work of optical illusions in relation to depth.

Optical illusions. The research of optical illusions has a long history in vision science. Michael Bach provides a vast collection of optical illusions on `http://www.michaelbach.de/ot/`, which use different perceptual cues such as motion (dotted line [IAC09], reverse Phi illusion [AR86]), luminance and contrast (Hermann grid [Spi94], the pyramid illusion [RMR83]), color (color fan[ZECL12, RE12]), geometric (Zöllner illusion, disjointed arch), size constancy (moon illusion [RP02]), etc. Gregory et al. [Gre97] classify the phenomena of illusions into four main classes: ambiguities (Necker cube), distortions (Ponzo figure), paradoxes (Tri-bar impossible figure) and fictions (Kanizsa square). Among them, some optical illusions already have a long history, whose mechanisms are well studied while others still lack a successful explanation [Oli06]. It has been only a short time that split-depth illusions are produced and little investigation was done in automating this process.

Occlusion depth cue. Depth perception helps us perceive the world in 3D and there are various kinds of depth cues, which are typically classified into binocular cues and monocular cues. Without 3D devices, we typically encounter monocular cues in animations - depth information that can be perceived with just one eye. Motion parallax [KDR*16, LSE17], size, texture gradient [BL76], contrast, perspective, occlusion [PBL07], and shadows [BG07] are examples of these. Occlusion is a particularly strong depth cue [Cut95], which can be used for various purposes, such as depth recovering [SCN08, SSN07], or depth enhancement. In this work, we focus on the latter one. Ritschel et al. [RTMS12] provided a framework to improve the perceived 3D effect by adding a virtual passepartout to RGBD images. Later, Zheng et al. [ZZS13] extended this work by incorporating scene saliency into the optimization. A similar work [SCRS09] presented an

intuitive user interface for fast "Out of Bounds" prototyping by adding 3D frames to 2D photographs. These approaches are mostly restricted to static images. Finding the optimal splitting plane in animations is difficult and often error prone as the object's motion is in general unrestricted and the 3D impression is quickly reduced by a non optimal placement. Nonetheless, well-placed simple splits in the form of horizontal or vertical bars can provide a strong occlusion depth cue. In this chapter, we investigate the creation of split-depth images and various factors that contribute to their effectiveness.

4.3. OVERVIEW

Our framework is illustrated in Figure 4.2. Given an RGBD image sequence, a mask to encode the OOI, and a choice for the number of splits and their width, we seek to generate optimal splits (potentially, in combination with virtual passepartouts). Video input with depth has become wide-spread (e.g., Kinect). In this case, an OOI can be extracted using segmentation and rotoscoping.

The core of our solution builds upon an energy optimization. To this extent, we first conduct a few visual perception experiments to validate assumptions that we will then integrate into our energy formulation (Sec. 4.4). We summarize the motion of the OOI through space and time (Sec. 4.5.1), which serves as a hard constraint and basis of our approach. Given these elements, we build a formulation for the optimal positioning of splits (Sec. 4.5.2). We then demonstrate our approach on various examples and show its effectiveness via an evaluation with novice users (Sec. 4.6) on complex datasets, before concluding (Sec. 4.7).

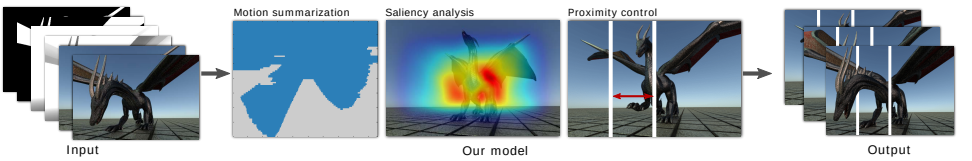


Figure 4.2: Overview: Given an RGBD image with a mask channel to indicate an object of interest, we summarize the motion spatially and temporally. This summary, together with other factors, such as saliency and proximity, will guide the split finding process.

4.4. PRELIMINARY STUDY

In this section, we study the assumptions that will guide our optimization. Throughout our experiments, unless otherwise stated, we used two splits, as it is the most common setup found in the many examples that are presented online, although there are a few cases where the number of splits varies or non-vertical bars are used. We formulate the following hypotheses, which we base on our observations from various online examples:

- Preference for split width varies on an individual basis;
- Splits should be at the same depth layer, otherwise the result might seem implausible;
- People prefer splits with a narrower gate (bar distance to the object is smaller);

- Main features in the scene should not be obstructed by the splits.

For the first assumptions, we deliberately avoid content influence. We thus investigate scenes using abstract cubes before testing two more complex scenes. In total, we performed four experiments in this preliminary study and involved 45 users with normal or corrected-to-normal vision. During the training session, we introduce split-depth images to the participants by showing them some previously collected examples. Once the concept was clear, we started the experiments. To avoid absolute scales for preference, we used a forced paired comparisons method [MTM12, LCTS05]. Here, a preference choice has to be made between two shown exemplars. We avoid biased results by randomizing the tests.

Experiment 1: Testing preference for split width. We used two different scenes from which we created three split-depth images with different split width (small (2% screen width), middle (4% screen width), and big (8% screen width)). For each pair, we asked participants to choose the one which looks as if the object is moving closer to them.

The result is recorded in the preference matrix shown in Table 4.1. In total, it records $270 = 45 * 3 * 2$ evaluations, used as a *Score*. The numbers indicate the number of times that the corresponding image sequence was preferred. E.g., the cell in row *Small*, column Middle has a value of 52 indicating that 52 times Small was preferred to Middle in a direct comparison.

The results are not entirely conclusive, even though the splits with the biggest width score the highest, there is no clear decreasing or increasing tendency shown as the split width increases. Our assumption that width is based on personal preference seems thus valid. In consequence, we made the split width a user-defined parameter.

Table 4.1: Preference matrix for split width. For each condition, there are 90 evaluations; for example, small (38) vs middle (52).

	Small	Middle	Big	Score
Small	–	52	35	**87**
Middle	38	–	34	**72**
Big	55	56	–	**111**

Experiment 2: Splits should share the same depth to avoid an implausible appearance. Again, we used two scenes, one with splits at the same depth and one with different depth layers but with the same distance of the splits in 3D. Note that due to perspective foreshortening, the distance between the splits that are located at different depths appears smaller. For each pair, we asked, which one is more plausible. The result is illustrated in Figure 4.3. 74 out of 90 (binomial test, $p < .000001$) choices favor the ones, where splits are placed at the same depth layer. Even though these results are statistically significant, we cannot exclude that other factors, such as motion direction, might influence a user's preference, which should be investigated in further experiments.

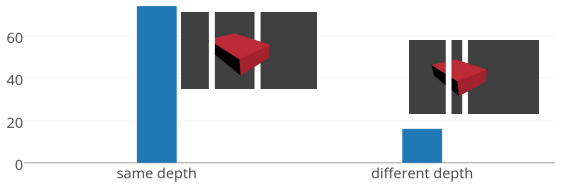


Figure 4.3: Preference comparison between bar placed at the same depth and different depth.

Experiment 3: Narrower gates are preferred. We used two scenes and created three split-depth image sequences each, where the opening between the gates ranges in width (narrow (around 15% screen width), middle (around 30% screen width), wide (around 45% screen width)), respectively. Again, participants perform 6 pair comparisons and were asked to choose the one, which they perceive as having a stronger depth.

The result is recorded in Table 4.2. The study shows that people preferred narrower gates as shown in the last column - the score increases when bars get closer. While not entirely conclusive, the results indicate that placing the splits as close as possible to the OOI is generally preferable if no other factors, such as scene content influence the perception, see Experiment 4.

Table 4.2: Preference matrix for split proximity.

	Narrow	Middle	Wide	Total
Narrow	–	54	56	**110**
Middle	36	–	57	**93**
Wide	34	33	–	**67**

Experiment 4: Main features should not be obstructed by bars. We do want to maintain the visibility of the main features in the animation. In practice, these would have to be estimated or otherwise derived (e.g., eye tracking). This constraint may conflict with the expected preference for a narrow opening between splits. In this study we want to validate that in some cases proximity of splits are more important than the scene saliency and vice versa. We present two scenes (Dragon, Balls) and placed the bars at five different locations having different proximity of the bars and covering differently salient regions, Fig. 4.4. For each animation, we generated five split-depth image sequences and inquired regarding preference. As there are many factors that can play a role, we asked for the reasoning via a textbox.

As shown in Figure 4.5, for the dragon scene 23 out of 45 (binomial test, $p < .000001$) participants chose the version with the narrow gate, whereas for the ball scene 17 out of 45 (binomial test, $p = .002594$) chose the one where the main object is more visible. It is important to note that for the dragon scene, which has a simplistic background, 8 participants mentioned in their reasoning that they preferred the increased depth perception due to the narrow gates. For the ball scene, which has a more complex background, 12 participants mentioned that they preferred a wider gate not because it provided a stronger 3D impression but because the narrow gate occluded salient parts of the scene.

These findings illustrate the complexity of the problem, as it indicates that it is scene-dependent. The preferred balance between covered salient elements and proximity of the bars can thus vary. For this reason, our approach lets the user determine the balancing between these two factors, e.g., if salient parts are hidden the impact of saliency preservation can be adjusted by increasing the according parameter in our framework. Future studies with more diverse scenes could give more insight into the impact of individual features on the perceived result.

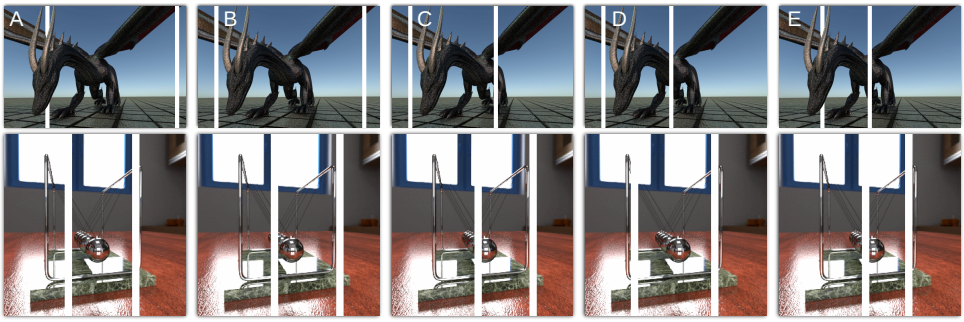


Figure 4.4: Scenes (dragon and ball scenes) used in experiment 4.

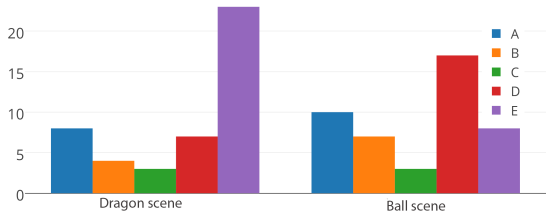


Figure 4.5: Votes of people's preference in experiment 4.

4.5. OUR APPROACH

We will now use the findings of our preliminary study to develop an optimization procedure for the placement of splits. First, in Sec. 4.5.1, we will explain our motion summarization, which is used to ensure that no intersections between the scene and the splits occur. Then we derive an energy formulation that is used in the actual optimization process in Sec. 4.5.2. The input to our framework are a set of RGB images $(\mathbf{I_1}, \mathbf{I_2}, ..., \mathbf{I_n})$, as well as corresponding depth images $(\mathbf{D_1}, \mathbf{D_2}, ..., \mathbf{D_n})$, and masking images of the OOI $(\mathbf{M_1}, \mathbf{M_2}, ..., \mathbf{M_n})$. All images have size $w \times h$. Two user-defined parameters that will be involved in the optimization immediately are the number N and width w_b of the splits.

4.5.1. MOTION SUMMARIZATION

Motion in split-depth images can be arbitrary and parts or entire objects might change. Consequently, object centroids or similar approximations will not well characterize the animation. Instead, we summarize the spatial and temporal information via a 3D histogram approach.

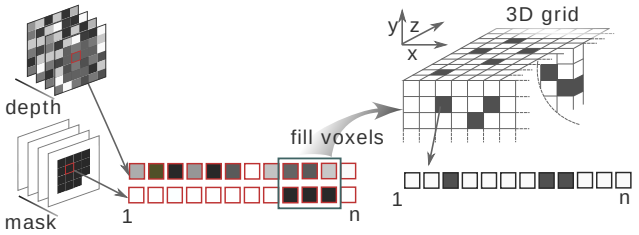
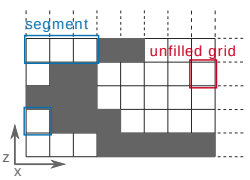


Figure 4.6: Motion summarization scheme. For each voxel inside the 3D grid, we record the contained depth values of the OOI.

As shown in Figure 4.6, we discretize our scene into a 3D voxel grid with dimensions of $w \times h \times k$, where k is a number of bins between the object's minimum and maximum extent along the z direction during the animation. As the intersection with the static scene would be simple to test and as it is not as disturbing as intersecting a moving object, we typically only insert the depth values of the OOI into this 3D grid. All unfilled voxels indicate room to add potential splits. In consequence, it becomes possible to enumerate all options and test an energy function, derived in Sec. 4.5.2.



To accelerate computations, we rely on a strategy from path finding and compute the Minkowski sum between the inversed 2D split shape and the 3D grid. Voxels that remain empty after this convolution will exactly correspond to valid positions of the split [dBvKOS00]. If the splits are represented by axis-aligned bars, we can project the entire grid on this axis and reduce the problem's dimensionality. To facilitate explanations, we will assume this case in the following. The figure to the right illustrates the resulting representation. The Minkowski sum is similar to a convolution and has the positive side effect to reduce noise. Figure 4.7 illustrates this effect for different split widths.

To enumerate all possible split configurations, we should remember our findings from Sec. 4.4. Splits should form gates through which the OOI moves in order to induce a 3D effect. Additionally, these splits should share the same depth. In consequence, if we slice the motion summarization grid with a plane at a certain depth, the splits should separate the OOI intersections with this plane.

For example, if we assume the user indicated that two splits should be used and a given depth layer leads to three connected regions that indicate potential placements, $(\mathbf{A},\mathbf{B},\mathbf{C})$, we will test the combinations $(\mathbf{A},\mathbf{B})$, $(\mathbf{A},\mathbf{C})$, and $(\mathbf{B},\mathbf{C})$. Generally, there will be $\binom{K}{N} = \frac{K!}{(K-N)!N!}$ possible solutions, where N is the number of splits and K the number of regions for possible placements. If a given depth does not allow for the user defined number of splits to form gates, we can proceed to the next discrete depth level. A last condition is that splits need to form gates, which the OOI traverses. To ensure this condition, we can test the OOI against each split along the sequence and two consecutively overlapped splits form a gate if the OOI is in front of the first, then behind the second split, or vice versa. Only if all splits form gates, the configuration is tested.

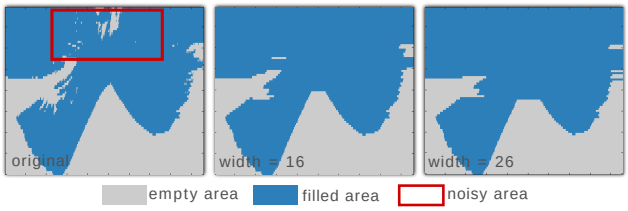


Figure 4.7: Larger bars lead to fewer potential placements, but reduce noise in the sequence, as highlighted in the red rectangle.

4.5.2. SPLIT OPTIMIZATION

We define an energy functional to encode and optimize various energy terms, denoted as E_{fb}, E_t, E_p, and E_s:

$$\min(\lambda_{fb}E_{fb} + \lambda_t E_t + \lambda_p E_p + \lambda_s E_s). \tag{4.1}$$

E_{fb}, E_t relates to the visibility of the occlusion by the splits, E_p to the proximity of the splits, and E_s to the saliency in the scene. By default, we propose the parameters $\lambda_{fb} = 1.0, \lambda_t = 0.1, \lambda_p = 0.5, \lambda_s = 0.5$, which work usually well in practice. Nonetheless, the user has the possibility to adjust settings, as Sec. 4.4 showed that some elements, such as preference for proximity and covering of salient elements, vary from scene to scene and individual to individual. This energy functional is optimized by evaluating various split configurations.

Occlusion cue. Occlusion is key in producing the depth effect. In consequence, a user will want to make sure that the occlusion is well noticed by the observer. We translate this condition into how many pixels the OOI is actually in front or behind the given splits over the duration of the video.

To compute this result, we calculate the number of pixels T_i of the OOI that overlap with the split i for the current configuration. Let $T_{\max}$ be the maximal number among all splits, then we define the energy:

$$E_{fb} = \sum_i 1 - |T_i / T_{\max}| \tag{4.2}$$

A related energy E_t, for trailing, will ensure that the OOI is not hidden in the beginning of the sequence, as it will otherwise not be visible to the observer and the frames would be useless for the animation. As not all sequences will allow us to fulfill a hard time constraint, we formulate this condition as an energy as follows:

$$E_t = \sum_i 1 - \min(V/U, 1), \tag{4.3}$$

where U is a user-defined constant (the desired number of frames that the OOI is not occluded by the split), V is a constant (the actual number of frames that the OOI is visible, thus, not occluded). A similar condition can be added for the end of the sequence, if no overlap is wanted.

Saliency cue. Main features of the scene, even if they are not part of the OOI, should not be blocked by the splits. As importance is difficult to derive, we use a mean saliency term as an estimate. To this extent, we compute the saliency of all input RGB frames and sum the contributions [HKP06], Fig. 4.8. The energy is then defined as:

$$E_s = \sum_i S_i \tag{4.4}$$

where S_i is the integrated mean saliency distribution underneath split i. In other words, split placements will be preferred that cover less salient regions.

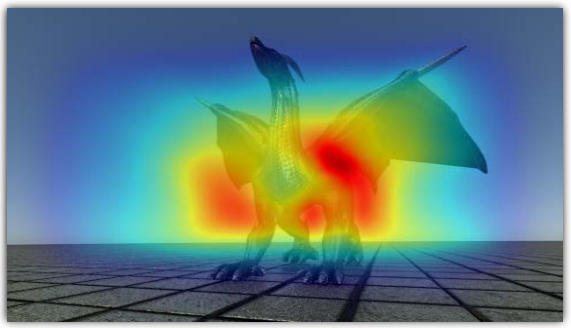


Figure 4.8: Mean saliency map for the input RGB frames overlaid with a single frame from the input video for visualization purposes.

Proximity cue To integrate a preference for narrower gates, as was investigated in our study in Sec. 4.4, we measure the distance between the two neighboring splits and encode it in the following energy formulation:

$$E_p = \sum_i D_i \tag{4.5}$$

where D_i is the distance between two neighboring splits (for parallel bars, it is just their distance, for general splits, one could use the Hausdorff distance). The terms are normalized by the screen width. As the splits are not allowed to intersect with the OOI there is a natural lower bound for D_i, namely the width of the OOI. If the user wants to only use one bar, this term is ignored.

4.6. RESULTS AND DISCUSSION

We have implemented our framework in Matlab on a desktop computer with an Intel Core i7 3.7 GHz CPU. We did not optimize the code for performance. The timing for the motion summarization is linear in the number of input images, ranging from a few seconds to several minutes, but could be easily parallelized. The optimization to look for the bar positions can be done within a few seconds.

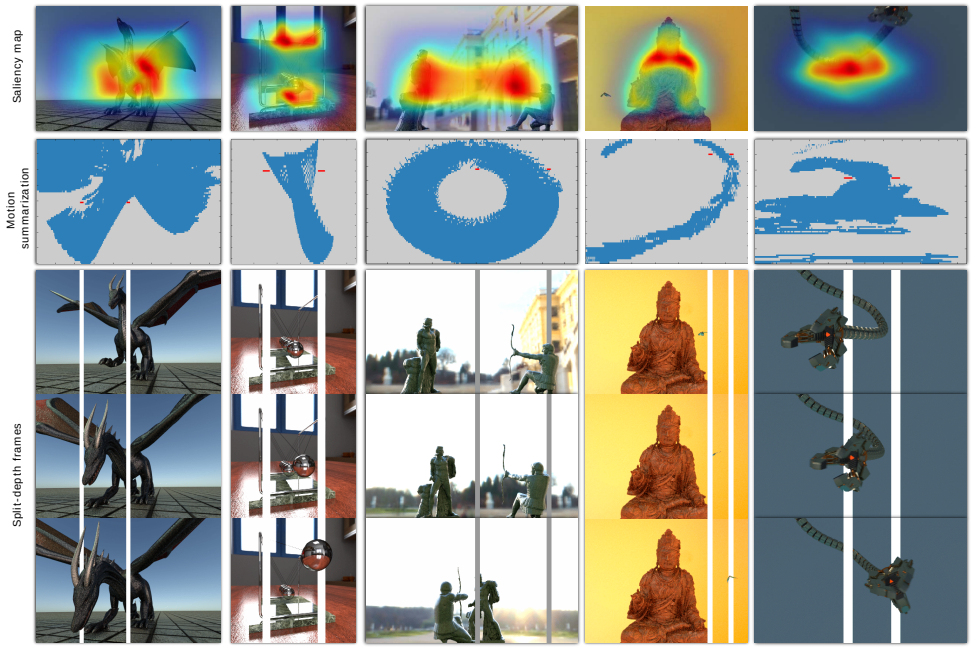


Figure 4.9: Examples of our approach on various data sets. Please refer to the supplemental videos for the animations. From top to bottom, row 1: mean scene saliency map; row 2: projection of the motion summarization with optimal positions for splits in red; row 3 - 5: example split-depth frames.

4.6.1. SPLIT-DEPTH GIFS RESULTS

We demonstrate our results on a range of scenes. Please refer to our supplemental material for several split-depth images, of which we show a few example frames in Figure 4.9 (row 3 - 5). We also visualize the final optimized position on the motion summarization map, Figure 4.9 (row 2). We created scenes with similar motion to the most popular split-depth gifs currently available. Interestingly the artists avoid much clutter in the background or strong camera motion. Strong camera motion would also contradict with the static positioning of the splits and affect depth perception.

The improvement using split-depth images is diminished if white splits are used in front of a bright background. We, therefore, offer the option to adapt the color of the splits to different gray-scales, Fig. 4.10. The user is also free to choose the number of splits for each scene, although 2 is the default. 1 or 3 is rarely needed and we encountered no case, where more than 3 was beneficial. A useful maximal number of splits can be derived directly from the topology of the summarization.

An interesting 3D effect can also be achieved by combining the split-depth images with virtual passepartouts. Our system automatically proposes to enlarge the bar towards the image boundary if the integrated saliency is comparably low (per default the splits should not cover more than 15% of the total saliency in the image), Fig. 4.11.

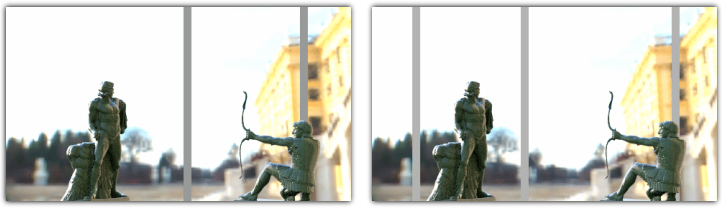


Figure 4.10: Adaption of color and number of splits.

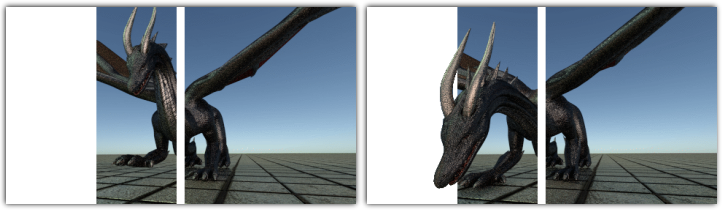


Figure 4.11: Combination using bars and virtual passepatouts

4.6.2. USER VALIDATION

To test the applicability of our model in scenes with comparably complex animations, we have performed a validation user study to compare our automatic method with manual split placement. As it is a young artform, professional split-depth image artists are

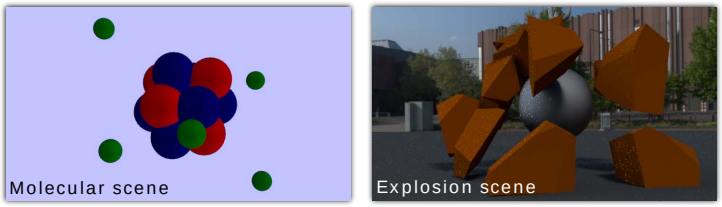


Figure 4.12: Example frames of two data sets we used in our validation study.

scarce. Additionally, as our work targets users with little experience, we conducted our study with three users, novice to creating split-depth images but with differing knowledge on image editing. Their task was to manually place the splits in the 3D scenes shown in Fig. 4.12. Occlusion was then automatically derived from the corresponding depth values. Before the study, we showed the participants examples of well done split-depth images. During the experiment, they could position the splits freely in 3D until they were pleased with the results. Only vertical splits were allowed. There was no time limitation throughout the whole study. During the experiment, we kept track of the number of split position adjustments and the result is shown in Table 4.3. It is worth mentioning that, in all test sequences, participants had to experiment with several positions to get to their final result.

Figure 4.14 row 1 shows the final selected position of each participant and the result

Table 4.3: Number of bar adjustments.

	User A	User B	User C
Explosion scene	6	8	8
Molecular scene	10	20	17

of our algorithm in the scene summarization. Row 2 to 5 depicts some example frames for each user. In most of the users' results, the scene objects penetrated the splits resulting in implausible scene constellations, whereas our framework found acceptable positions.

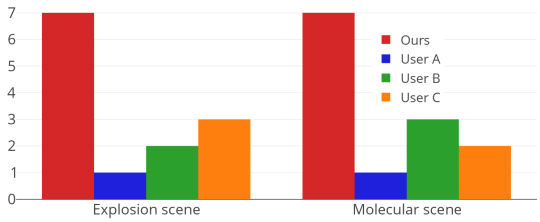


Figure 4.13: Comparison of people's preference between novice users' results and ours.

To further verify if our framework can generate more appealing results than those created by novice users, we conducted an additional user study with 13 subjects with normal or corrected-to-normal vision and asked for their preference by showing them the results of the novice users and ours. The user study setup is similar to that in Sec. 4.4. Figure 4.13 shows the results of people's preference. For both datasets, 7 out of 13 (binomial test, p = .018686) subjects liked ours best. The other results were comparably similar and no clear preference exists.

4.7. CONCLUSION AND FUTURE WORK

We presented an algorithm for automatic split-depth image creation from RGBD image sequences, which takes important factors, such as spatial and temporal motion information, scene saliency and proximity of the splits to the object of interest into account. We also provided means to manually set and test different parameters, such as color and width of the splits, while the succeeding optimization is fully automatic, which enables an easy exploration of effective solutions. We validated the importance of these factors through a preliminary study and demonstrated the usefulness of our presented model and the optimization in a second user study.

Our method is subject to certain limitations. Imperfect masking or depth-of-field in the animation, invalidate our current input assumptions. Using natural image matting techniques, one could separate fore- and background, although (especially for videos) these techniques are still highly error-prone and require substantial manual effort. As mentioned in Sec. 4.6.1, scenes with much clutter in the background or strong camera motion are often avoided by split-depth image artists. Complex motion in itself is not a problem for our algorithm. However, many objects with complex motion can poten-

tially lead to a case where the motion summarization is filled and our algorithm fails to find good positions for the splits. This is, however, no real limitation, as it simply implies that there is no possible intersection-free position for the splits. In fact, it can be seen as a benefit of our algorithm, as it tells the user that the scene in its current form is not well suited for an effective split-depth image. A solution might be non-linear splits (e. g., circle, ellipse) or animated splits which fade in/out or move throughout the animation but this is left for future work as the effect on depth perception is unclear. Another fruitful direction for further research is the extension of our approach to plain RGB image sequences without depth information.

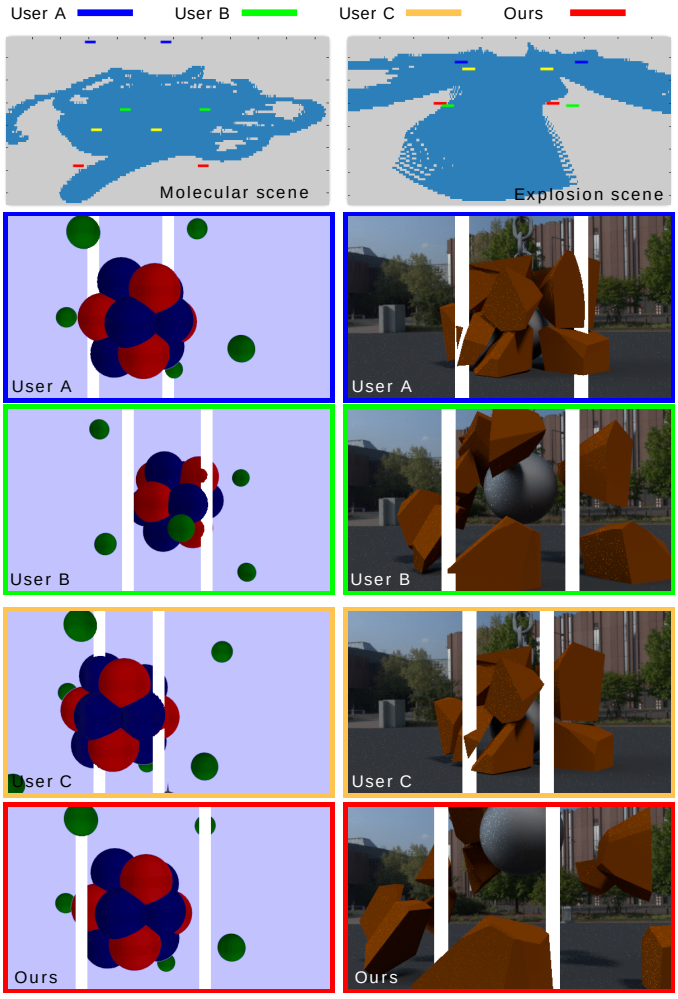


Figure 4.14: Results of novice users and example frames. Note the intersections in the results of novice users.

5

CONCLUSIONS

Depth is a powerful information for image-manipulation techniques. We are convinced that depth will increasingly find its way into common image representations. First indicators are the integration of additional sensors in modern cameras. Still, such portable hardware is in its infancy. More precise reconstructions require complex setups or restrict the type of scenes to be captured.

While future sensors will have higher resolution, increasing quality, and less cost, they will never be perfect and many ambiguous scene configurations exists, especially in the presence of specular elements, which are handled by our solution. In principle, we believe that a combination of various approaches, including sensor data, depth reconstruction via multiple images, or machine-learning-based solutions will be combined for highest effectiveness. This idea of *depth fusion* [OERW*15, EHH15] has been proposed earlier and first results indicate that a higher accuracy and resolution can be obtained by compensating the limitations of the various methods. Our work also connects to this idea by introducing *depth palettes* in Chapter 3, where input from different reconstruction methods can be integrated by the user.

Our goal of this work is to make a step towards a more wide-spread use of depth information in images. We presented solutions that provide a simpler and cheaper setup for depth reconstruction, as well as tools that enable a direct creation of a depth map. The latter solution is particularly interesting as it enables also the manipulation of depth maps and we show with various examples that such modifications can be beneficial. In this case, depth information does not have to be accurate but rather serves as a support for artistic control.

In this sense, this dissertation advocates to interpret depth not necessarily as an absolute measure. Depth usually relates to stereo, which is a cue that is hard to reproduce on a 2D medium. We showed with various means that even approximate depth can be used to manipulate the image content to add cues that support a better depth separation. Hereby, we can compensate to some degree for the otherwise lost depth perception.

A deeper understanding of human perception will allow us to build upon our findings and develop new algorithms that translate depth information into a visual support for

depth perception. Such an endeavor will be challenging but also exciting, as it will entail the combination of various factors of the human visual system. In this sense, our work contrasts with typical approaches that transform images into depth, as it instead adds depth cues into images. In the future, we would like to explore the many application scenarios that can benefit from such enhanced depth perception.

BIBLIOGRAPHY

[AFG13] ACKERMANN J., FUHRMANN S., GOESELE M.: Geometric point light source calibration. In *VMV* (2013), pp. 161–168.

[AHP12] ABRAMS A., HAWLEY C., PLESS R.: Heliometric stereo: Shape from sun position. In *Computer Vision–ECCV 2012*. Springer, 2012, pp. 357–370.

[AHSS04] AGARWALA A., HERTZMANN A., SALESIN D. H., SEITZ S. M.: Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (ToG) 23*, 3 (2004), 584–591.

[AR86] ANSTIS S. M., ROGERS B. J.: Illusory continuous motion from oscillating positive-negative patterns: implications for motion perception. *Perception 15*, 5 (1986), 627–640.

[ASC11] ANDERSON R., STENGER B., CIPOLLA R.: Color photometric stereo for multicolored surfaces. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 2182–2189.

[Asl07] ASLANTAS V.: A depth estimation algorithm with a single image. *Optics express 15*, 8 (2007), 5024–5029.

[ASSS14] AHMAD J., SUN J., SMITH L., SMITH M.: An improved photometric stereo through distance estimation and light vector optimization from diffused maxima region. *Pattern Recognition Letters 50* (2014), 15–22.

[AWL15] AITTALA M., WEYRICH T., LEHTINEN J.: Two-shot SVBRDF capture for stationary materials. *ACM Trans. Graph. 34*, 4 (2015), 110:1–110:13.

[AZK08] ALLDRIN N., ZICKLER T., KRIEGMAN D.: Photometric stereo with non-parametric and spatially-varying reflectance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[Bal81] BALLARD D. H.: Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition 13*, 2 (1981), 111–122.

[BEDT10] BEZERRA H., EISEMANN E., DECARLO D., THOLLOT J.: Diffusion constraints for vector graphics. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (2010), ACM, pp. 35–42.

[BG07] BRUCKNER S., GRÖLLER E.: Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1344–1351.

[BJK07] BASRI R., JACOBS D., KEMELMACHER I.: Photometric stereo with general, unknown lighting. *International Journal of Computer Vision 72*, 3 (2007), 239–257.

[BKY99] BELHUMEUR P. N., KRIEGMAN D. J., YUILLE A. L.: The bas-relief ambiguity. *International journal of computer vision 35*, 1 (1999), 33–44.

[BL76] BAJCSY R., LIEBERMAN L.: Texture gradient as a depth cue. *Computer Graphics and Image Processing 5*, 1 (1976), 52–67.

[BSD08] BAVOIL L., SAINZ M., DIMITROV R.: Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 Talks* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 22:1–22:1.

[BTP13] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Sparse iterative closest point. In *Computer graphics forum* (2013), vol. 32, Wiley Online Library, pp. 113–123.

[CAK07] CHANDRAKER M., AGARWAL S., KRIEGMAN D.: Shadowcuts: Photometric stereo with shadows. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8.

[CBR13] CHANDRAKER M., BAI J., RAMAMOORTHI R.: On differential photometric reconstruction for unknown, isotropic brdfs. *IEEE transactions on pattern analysis and machine intelligence 35*, 12 (2013), 2941–2955.

[CED*16] CALAGARI K., ELGAMAL T., DIAB K., TEMPLIN K., DIDYK P., MATUSIK W., HEFEEDA M.: Depth personalization and streaming of stereoscopic sports videos. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 12*, 3 (2016), 41.

[Cla92] CLARK J. J.: Active photometric stereo. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (1992), IEEE, pp. 29–34.

[CLER07] CHIA A. Y. S., LEUNG M. K., ENG H.-L., RAHARDJA S.: Ellipse detection with hough transform in one dimensional parametric space. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on* (2007), vol. 5, IEEE, pp. V–333.

[CPM*16] CIORTAN I., PINTUS R., MARCHIORO G., DAFFARA C., GIACHETTI A., GOBBETTI E., ET AL.: A practical reflectance transformation imaging pipeline for surface characterization in cultural heritage.

[CRZ00] CRIMINISI A., REID I., ZISSERMAN A.: Single view metrology. *International Journal of Computer Vision 40*, 2 (2000), 123–148.

[Cut95] CUTTING J. E.: Potency, and contextual use of different information about depth. *Perception of space and motion* (1995), 69.

[dBvKOS00] DE BERG M., VAN KREVELD M., OVERMARS M., SCHWARZKOPF O.: Computational geometry: Algorithms and applications. Second ed. Springer-Verlag, 2000, ch. 13, pp. 290–297.

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 145–156.

[DRE*11] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: A perceptual model for disparity. *ACM Trans. Graph. 30*, 4 (July 2011), 96:1–96:10.

[EHH15] EVANGELIDIS G. D., HANSARD M., HORAUD R.: Fusion of range and stereo data for high-resolution scene-modeling. *IEEE transactions on pattern analysis and machine intelligence 37*, 11 (2015), 2178–2192.

[EPD09] EISEMANN E., PARIS S., DURAND F.: A visibility algorithm for converting 3d meshes into editable 2d vector graphics. *ACM Trans. Graph. (Proc. of SIGGRAPH) 28* (July 2009), 83:1–83:8.

[EPF14] EIGEN D., PUHRSCH C., FERGUS R.: Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems* (2014), pp. 2366–2374.

[GDA*11] GERRITS M., DECKER B. D., ANCUTI C., HABER T., ANCUTI C., MERTENS T., BEKAERT P.: Stroke-based creation of depth maps. In *2011 IEEE International Conference on Multimedia and Expo* (July 2011), pp. 1–6.

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 43–54.

[GHB*05] GORSKI K. M., HIVON E., BANDAY A., WANDELT B. D., HANSEN F. K., REINECKE M., BARTELMANN M.: Healpix: a framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal 622*, 2 (2005), 759.

[Gre97] GREGORY R. L.: Visual illusions classified. *Trends in cognitive sciences 1*, 5 (1997), 190–194.

[HGZGL15] HOLD-GEOFFROY Y., ZHANG J., GOTARDO P. F., LALONDE J.-F.: x-hour outdoor photometric stereo. In *3D Vision (3DV), 2015 International Conference on* (2015), IEEE, pp. 28–36.

[HKP06] HAREL J., KOCH C., PERONA P.: Graph-based visual saliency. In *Proceedings of the 19th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2006), NIPS'06, MIT Press, pp. 545–552.

[HMJI09] HIGO T., MATSUSHITA Y., JOSHI N., IKEUCHI K.: A hand-held photometric stereo camera for 3-d modeling. In *2009 IEEE 12th International Conference on Computer Vision* (2009), IEEE, pp. 1234–1241.

[IAC09] ITO H., ANSTIS S., CAVANAGH P.: Illusory movement of dotted lines. *Perception 38*, 9 (2009), 1405–1409.

[ISI90] IWAHORI Y., SUGIE H., ISHII N.: Reconstructing shape from shading images under point light source illumination. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on* (1990), vol. 1, IEEE, pp. 83–87.

[IWMA12] IKEHATA S., WIPF D., MATSUSHITA Y., AIZAWA K.: Robust photometric stereo using sparse regression. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE, pp. 318–325.

[KDR*16] KELLNHOFER P., DIDYK P., RITSCHEL T., MASIA B., MYSZKOWSKI K., SEIDEL H.-P.: Motion parallax in stereo 3d: model and applications. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 176.

[KLK14] KARSCH K., LIU C., KANG S. B.: Depthtransfer: Depth extraction from video using non-parametric sampling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2014).

[KYS03] KIM H., YANG S.-J., SOHN K.: 3d reconstruction of stereo images for interaction between real and virtual worlds. In *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality* (Washington, DC, USA, 2003), ISMAR '03, IEEE Computer Society, pp. 169–176.

[LBRF12] LAI K., BO L., REN X., FOX D.: Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 1330–1337.

[LCD06] LUFT T., COLDITZ C., DEUSSEN O.: Image enhancement by unsharp masking the depth buffer. *ACM Trans. Graph. 25*, 3 (July 2006), 1206–1213.

[LCTS05] LEDDA P., CHALMERS A., TROSCIANKO T., SEETZEN H.: Evaluation of tone mapping operators using a high dynamic range display. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 640–648.

[LCZ99] LIEBOWITZ D., CRIMINISI A., ZISSERMAN A.: Creating architectural models from images. In *Computer Graphics Forum* (1999), vol. 18, Wiley Online Library, pp. 39–50.

[LEE17] LIAO J., EISEMANN M., EISEMANN E.: Split-depth image generation and optimization. *Computer Graphics Forum 36*, 7 (2017).

[Len04] LENSCH H. P.: *Efficient, Image-Based Appearance Acquisition of Real-World Objects*. Cuvillier Verlag, 2004.

[LES09] LEE S., EISEMANN E., SEIDEL H.-P.: Depth-of-field rendering with multiview synthesis. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia) 28*, 5 (2009).

[LES10] LEE S., EISEMANN E., SEIDEL H.-P.: Real-time lens blur effects and focus control. In *ACM Transactions on Graphics (TOG)* (2010), vol. 29, ACM, p. 65.

[LFDF07] LEVIN A., FERGUS R., DURAND F., FREEMAN W. T.: Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG) 26*, 3 (2007), 70.

[LGG14] LOPEZ A., GARCES E., GUTIERREZ D.: Depth from a Single Image Through User Interaction. In *Spanish Computer Graphics Conference (CEIG)* (2014), Munoz A., Vazquez P.-P., (Eds.), The Eurographics Association.

[LHK09] LEE D. C., HEBERT M., KANADE T.: Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 2136–2143.

[LHW*10] LANG M., HORNUNG A., WANG O., POULAKOS S., SMOLIC A., GROSS M.: Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. Graph. 29*, 4 (July 2010), 75:1–75:10.

[LJXD13] LIN J., JI X., XU W., DAI Q.: Absolute depth estimation from a single defocused image. *IEEE Transactions on Image Processing 22*, 11 (2013), 4545–4550.

[LKT*15] LESKENS J. G., KEHL C., TUTENEL T., KOL T. R., HAAN G. D., STELLING G. S., EISEMANN E.: An interactive simulation and visualization tool for flood analysis usable for practitioners. *Mitigation and Adaptation Strategies for Global Change* (May 2015), 1–18.

[LSE17] LIAO J., SHEN S., EISEMANN E.: Depth map design and depth-based effects with a single image. In *Proc. of Graphics Interface (GI)* (2017).

[LSL15] LIU F., SHEN C., LIN G.: Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5162–5170.

[LTW14] LIN Y.-H., TSAI M.-H., WU J.-L.: Depth sculpturing for 2d paintings: A progressive depth map completion framework. *J. Vis. Comun. Image Represent. 25*, 4 (May 2014), 670–678.

[LW99] LEI Y., WONG K. C.: Ellipse detection based on symmetry. *Pattern recognition letters 20*, 1 (1999), 41–47.

[McL98] MCLAUGHLIN R. A.: Randomized hough transform: improved ellipse detection with comparison. *Pattern Recognition Letters 19*, 3 (1998), 299–305.

[MDA02] MASSELUS V., DUTRÉ P., ANRYS F.: The free-form light stage. In *ACM SIGGRAPH 2002 conference abstracts and applications* (2002), ACM, pp. 262–262.

[Men09] MENDIBURU B.: Chapter 5 - 3d cinematography fundamentals. In *3D Movie Making*, Mendiburu B., (Ed.). Focal Press, Boston, 2009, pp. 73 – 90.

[ML00] MURRAY D., LITTLE J. J.: Using real-time stereo vision for mobile robot navigation. *Autonomous Robots 8*, 2 (Apr 2000), 161–171.

[MQ*16] MECCA R., QUÉAU Y., ET AL.: Unifying diffuse and specular reflections for the photometric stereo problem. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2016), IEEE, pp. 1–9.

[MTM12] MANTIUK R. K., TOMASZEWSKA A., MANTIUK R.: Comparison of four subjective methods for image quality assessment. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 2478–2491.

[MWBK14] MECCA R., WETZLER A., BRUCKSTEIN A. M., KIMMEL R.: Near field photometric stereo with point light sources. *SIAM Journal on Imaging Sciences 7*, 4 (2014), 2732–2770.

[Nay89] NAYAR S. K.: Sphereo: Determining depth using two specular spheres and a single camera. In *1988 Robotics Conferences* (1989), International Society for Optics and Photonics, pp. 245–254.

[OBW*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion curves: A vector representation for smooth-shaded images. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)* (2008), vol. 27.

[OERW*15] OR-EL R., ROSMAN G., WETZLER A., KIMMEL R., BRUCKSTEIN A. M.: Rgbd-fusion: Real-time high precision depth recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5407–5416.

[Oli06] OLIVER S.: Optical illusions and their causes: Examining differing explanations. *AHS Capstone Projects. 7.* (2006).

[OOB15] Out of bounds photography. `https://pixabay.com/en/out-of-bounds-image-editing-horses-940381/`, 2015. Accessed: 2017-07-01.

[PBL07] PALMER S. E., BROOKS J. L., LAI K. S.: The occlusion illusion: Partial modal completion or apparent distance? *Perception 36*, 5 (2007), 650–669.

[Pen87] PENTLAND A. P.: A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, 4 (1987), 523–531.

[PF14] PAPADHIMITRI T., FAVARO P.: Uncalibrated near-light photometric stereo. *Proceedings of the British Machine Vision Conference* (2014).

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 313–318.

[PM90] PERONA P., MALIK J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence 12*, 7 (1990), 629–639.

[PSG01] POWELL M. W., SARKAR S., GOLDGOF D.: A simple strategy for calibrating the geometry of light sources. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 9 (2001), 1022–1027.

[RBF12] REN X., BO L., FOX D.: Rgb-(d) scene labeling: Features and algorithms. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), pp. 2759–2766.

[RDL*15] REN P., DONG Y., LIN S., TONG X., GUO B.: Image based relighting using neural networks. *ACM Trans. Graph. 34*, 4 (July 2015), 111:1–111:12.

[RE12] RITSCHEL T., EISEMANN E.: A computational model of afterimages. *Comp. Graph. Forum (Proc. Eurographics 2012) 31*, 2 (2012).

[RMR83] RATLIFF F., MILKMAN N., RENNERT N.: Attenuation of mach bands by adjacent stimuli. *Proceedings of the National Academy of Sciences 80*, 14 (1983), 4554–4558.

[RP02] ROSS H., PLUG C.: *The Mystery of The Moon Illusion-Exploring Size Perception.* 2002.

[RSI*08] RITSCHEL T., SMITH K., IHRKE M., GROSCH T., MYSZKOWSKI K., SEIDEL H.-P.: 3d unsharp masking for scene coherent enhancement. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 90.

[RTMS12] RITSCHEL T., TEMPLIN K., MYSZKOWSKI K., SEIDEL H.-P.: Virtual passepartouts. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (2012), Eurographics Association, pp. 57–63.

[SCN05] SAXENA A., CHUNG S. H., NG A. Y.: Learning depth from single monocular images. In *Advances in Neural Information Processing Systems* (2005), pp. 1161–1168.

[SCN08] SAXENA A., CHUNG S. H., NG A. Y.: 3-d depth reconstruction from a single still image. *International Journal of Computer Vision 76*, 1 (2008), 53–69.

[SCRS09] SHESH A., CRIMINISI A., ROTHER C., SMYTH G.: 3d-aware image editing for out of bounds photography. In *Proceedings of Graphics Interface 2009* (2009), Canadian Information Processing Society, pp. 47–54.

[SF14] SELLENT A., FAVARO P.: Which side of the focal plane are you on? In *Computational Photography (ICCP), 2014 IEEE International Conference on* (2014), IEEE, pp. 1–8.

[SLE17] SALAMON N., LANCELLE M., EISEMANN E.: Computational light painting using a virtual exposure. In *Computer Graphics Forum (Proceedings of Eurographics)* (2017), vol. 36, Eurographics, John Wiley & Sons.

[Spi94] SPILLMANN L.: The hermann grid illusion: a tool for studying human perceptive field organization. *Perception 23*, 6 (1994), 691–708.

[SS03] SCHARSTEIN D., SZELISKI R.: High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 1, IEEE.

[SSJ*10] SÝKORA D., SEDLACEK D., JINCHAO S., DINGLIANA J., COLLINS S.: Adding depth to cartoons using sparse depth (in) equalities. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 615–623.

[SSN07] SAXENA A., SCHULTE J., NG A. Y.: Depth estimation using monocular and stereo cues. In *IJCAI* (2007), vol. 7.

[SSN09] SAXENA A., SUN M., NG A. Y.: Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence 31*, 5 (2009), 824–840.

[STXJ15] SHI J., TAO X., XU L., JIA J.: Break ames room illusion: depth from general single images. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 225.

[SZP10] SUNKAVALLI K., ZICKLER T., PFISTER H.: Visibility subspaces: Uncalibrated photometric stereo with shadows. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 251–264.

[TK05] TANKUS A., KIRYATI N.: Photometric stereo under perspective projection. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 1, IEEE, pp. 611–616.

[TM78] TSUJI S., MATSUMOTO F.: Detection of ellipses by a modified hough transformation. *IEEE Transactions on Computers*, 8 (1978), 777–781.

[VP09] Virtual passepartouts. `https://www.flickr.com/photos/29412527@N04/3905723380`, 2009. Accessed: 2017-07-01.

[Wil87] WILLIS P.: Visual simulation of atmospheric haze. *Computer Graphics Forum 6*, 1 (1987), 35–41.

[WLF*11] WANG O., LANG M., FREI M., HORNUNG A., SMOLIC A., GROSS M.: Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling* (2011), ACM, pp. 47–54.

[WLGK16] WEINMANN M., LANGGUTH F., GOESELE M., KLEIN R.: Advances in Geometry and Reflectance Acquisition. In *EG 2016 - Tutorials* (2016), Sousa A., Bouatouch K., (Eds.), The Eurographics Association.

[WMTG05] WINNEMÖLLER H., MOHAN A., TUMBLIN J., GOOCH B.: Light waving: Estimating light positions from photographs alone. In *Computer Graphics Forum* (2005), vol. 24, Wiley Online Library, pp. 433–438.

[Woo80] WOODHAM R. J.: Photometric method for determining surface orientation from multiple images. *Optical engineering 19*, 1 (1980), 191139–191139.

[WWH05] WENGER P. D. A., WAESE C. T. A. G. J., HAWKINS T.: A lighting reproduction approach to live-action compositing. *Computer Graphics Proceedings, ACM SIGGRAPH* (2005).

[XDW15] XIE W., DAI C., WANG C. C.: Photometric stereo with near point lighting: A solution by mesh deformation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2015), IEEE.

[XJ02] XIE Y., JI Q.: A new efficient ellipse detection method. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (2002), vol. 2, IEEE, pp. 957–960.

[YSHSH13] YÜCER K., SORKINE-HORNUNG A., SORKINE-HORNUNG O.: Transfusive weights for content-aware image manipulation. In *VMV* (2013), pp. 57–64.

[YZ06] YING X., ZHA H.: *Interpreting Sphere Images Using the Double-Contact Theorem*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 724–733.

[ZCSM13] ZHU X., COHEN S., SCHILLER S., MILANFAR P.: Estimating spatially varying defocus blur from a single image. *IEEE Transactions on image processing 22*, 12 (2013), 4879–4891.

[ZCW*15] ZENG Q., CHEN W., WANG H., TU C., COHEN-OR D., LISCHINSKI D., CHEN B.: Hallucinating stereoscopy from a single image. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 1–12.

[ZECL12] ZAIDI Q., ENNIS R., CAO D., LEE B.: Neural locus of color afterimages. *Current Biology 22*, 3 (2012), 220–224.

[ZZS13] ZHENG Z., ZHANG Y., SUN Z.: Generating Pseudo-3D Painting Based on Visual Saliency and Composition Rules. In *Eurographics 2013 - Short Papers* (2013), Otaduy M.-A., Sorkine O., (Eds.), The Eurographics Association.

ACKNOWLEDGMENTS

I appreciate a lot for what I have experienced during the past four years. I could not finish my PhD without the help from too many people. Hereby, I would like to thank them.

Dear Elmar, I couldn't thank him enough for his excellent supervision, kind help and warmhearted care in the whole period of my PhD. He is such a fantastic person with great charisma. He always supports and encourages me to be a better researcher and person. I like to talk with him since there are always some takeaway messages from which I can benefit. I couldn't imagine what I could become without his guidance.

I would like to thank my collaborators. I started my PhD by doing one project together with Timothy, I thank him a lot for helping me catch up with some fundamental knowledge that I lacked. He always gives me valuable advices for my questions. At that time, I was also lucky to have Matthias around. Thank him for introducing MxEngine to me and helping me initialize my framework. Later, Jean-Marc and Bert stepped in. I would like to thank them for their supervision in my early PhD period. I learned a lot and really enjoyed the time we worked together. Dear Jean-Marc, his positive personality has a big impact on me. I really like and respect him. My other collaborators Pablo and Martin, thanks for the valuable discussions and helping me out when I had troubles.

I would like to thank the CGV people. Dear Noeska, thanks for her support and always being available when I feel down. Dear Changgong, my big Chinese brother, thank him for everything. Dear Christopher, when I have questions concerning maths, his name is the first one that pops in my head. Thank him for the discussions and gym time. I would also like to own my gratitude to other people in the group: Klaus, Rafael, Anna, Nestor, Victor, Ben, Sergio, Ricardo, Leo, Peiteng, Chaoran, Sungkil, Jerry, Niels (thanks for the coffee breaks together), Thomas Kroes, Thomas Höllt, Nicola, Renata. It's a great thing to get to know you and I would never forget the wonderful memories.

Ruud, Bart, Stefanie, Sandra and Marloes are deserved to mention separately for their technique and administrative support through these years.

I would also like to thank my committee members for accepting the invitation and providing valuable feedbacks.

My Chinese friends in Delft: Kaihua, Yunlong, Chong, Peiyao, Anqi, Xinyuan, Lei, Xinchao, Guangliang and others. So many thanks to them for making me not feel lonely when I am overseas alone.

In the end, I would also like to point out my parents and younger brother. Thank them for the trust and love.

CURRICULUM VITÆ

Jingtang LIAO

16-08-1988 Born in Sichuan, China.

EDUCATION

2006–2010 Bachelor of Science in Mechanic Engineering
Beihang University
Beijing, China

2010–2013 Master of Science in Aerospace Engineering
Beihang University
Beijing, China

2013 – 2017 PhD. Computer Science
Delft University of Technology
Delft, The Netherlands
Thesis: Techniques for Depth Acquisition and Enhancement of Depth Perception
Promotor: Prof. Dr. Elmar Eisemann

List of Publications

5. **Jingtang Liao**, Shuheng Shen, Elmar Eisemann, *Depth Annotations: Designing Depth of a Single Image for Depth-based Effects*, Computers & Graphics,(submitted).

4. **Jingtang Liao**, Martin Eisemann, Elmar Eisemann, *Split-Depth Image Generation and Optimization*, Computer Graphics Forum, vol.36, no.7, pages 175–182, 2017.

3. **Jingtang Liao**, Shuheng Shen, Elmar Eisemann, *Depth Map Design and Depth-based Effects With a Single Image*, Graphics Interface, pages 57–64, 2017.

2. **Jingtang Liao**, Bert Buchholz, Jean-Marc Thiery, Pablo Bauszat and Elmar Eisemann, *Indoor Scene Reconstruction Using Near-light Photometric Stereo*, IEEE Transactions on Image Processing, vol.26, no. 3, pages 1089–1101, 2017.

1. Timothy Kol, **Jingtang Liao**, Elmar Eisemann, *Real-time Canonical-angle Views in 3D Virtual Cities*, Vision, Modeling & Visualization, pages 55–62, 2014.