High Quality Depth Refinement with Color Photometric Stereo

Songyou Peng

Supervised by:

Dr. Yvain Quéau Prof. Daniel Cremers

Computer Vision Group

Department of Computer Science

Technical University of Munich

A Thesis Submitted for the Degree of
MSc Erasmus Mundus in Vision and Robotics (VIBOT)

· 2017 ·

Abstract

The abstract will go here....

Research is what I'm doing when I don't know what I'm doing. . . .

Werner von Braun

Contents

List of Figures

List of Tables

Acknowledgments

Leave this part until I finish the whole thesis

Chapter 1

Introduction

1.1 Research Goal

- The use of RGB-D camera, the use of depth information, application: visual odometry on quadropter, human motion capture, 3D reconstruction.
- but the depth image from consumer RGB-D camera has bad accuracy and noisy and missing data. does not contain any fine details, for example the button on the shirt, some wrincles on your hand, etc.
- some method tried to recover these details by fusing the depth data from multiple views [6], however, still the recovered details is very limited.
- so here comes our work, we explore the intrinsic details of the object in an image, analyze the lights' positions and its corresponding influence of the shading on the object, the reflectance rate of the material on the objects.
- talk a bit about SFS and PS, their basic definition: obtain the shape from an image when the light is known.
- say that combine SFS or PS with observed depth can eliminate ambiguities and has been widely used for shape or depth refinement by and it is usually formulated as an inverse problem. and nonlinearity within the normal exists in such an inverse problem, some people tries to use some nonlinear optimization to solve this problems but makes the process super slow, some others freeze the nonlinear part with the thing in the last iteration, but this sometimes leads to divergence problem. So we have a good method.

- say that some state-of-the-art methods need to make some strict assumptions like constant albedo, but not really in line with real world objects. Some others try to estimate the albedo by imposing some piece-wise smoothness terms, but the estimation is not satisfying and the surface normal cannot be separated from the albedo. And how good is ours
- the idea of our two methods. use what as the input, first estimate waht, then waht then what. (we explicitly estimate the in- cident illumination in the scene based on the reconstructed shape, make an estimate of the albedo distribution on the surface, and then use this information together with the lighting equation to recover the fine-grained structure and orientation of points on the surface. We assume a Lam- bertian model of reflection where incident lighting is given by an environment map that is parameterized in the spher- ical harmonic domain, and where surface properties are given by a spatially-varying albedo map.)

contribution of our method

1. an more efficient and faster implementation of a state-of-the-art method
2. present a new RGB ratio model which resolve the nonlinearity in the inverse problem and achieve similar results to state-of-the-art
3. We proposed a robust multi-light depth refinement method which outperforms the state-of-the-art both quantitatively and qualitatively. Moreover, no regularization is imposed at all.
4. We extended our work to the depth super-resolution.

1.2 Outline

Chapter 2

Background

2.1 RGB-D Cameras

2.1.1 General

The RGB-D camera has been widely applied in many modern computer vision areas, for example 3D reconstruction [6], visual odometry and mapping on quadrocopter [7] and visual SLAM algorithms [8]. A RGB-D camera return a color image which is usually in RGB color space, and a depth map, every pixel of which reflects the real-world distance between the camera and the corresponding position of the pixel. Depending on the technologies used to measure the depth information, the RGB-D camera can be divided to passive and active [9].

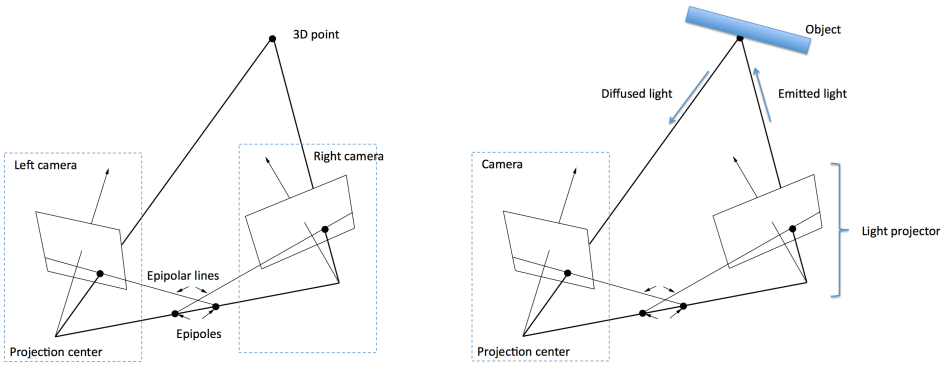
The so called passive RGBD-camera usually contains two RGB cameras with a known translation between them. After taking one picture for each, the features in two pictures are matched and then the triangulation is applied to obtain the depth. An illustration is shown in Fig. 2.1(a).

Active technologies usually emit lights to the environment so it has the capability of acquiring depth images in a totally dark indoor scenario. They can be furthered categorized as time of flight (ToF) or a structured light approach.

A ToF camera calculates the depth in each pixel by measuring the delay between the emission and the reflected time. ToF cameras typically emit either pulsed light or modulated light. The Microsoft Kinect 2.0 and IFM Efector are two examples of the ToF camera.

The RGB-D cameras with the structured light use a projector for a known pattern. Since the transformation between the camera the projector is pre-given, a camera observes the projected pattern and then triangulates to calculate the depth (Fig. 2.1(b)). The ASUS Xtion Pro Live, Intel RealSense R200 and Ensenso are several well-known cameras using structured lights. It

should be noted that many active stereo cameras project infrared (IR) lights. Due to the fact that sun is a source of infrared lights, these cameras are limited the usage only in the indoor environment.



(a) Passive stereo (b) Active stereo

Figure 2.1: Illustrations for the principle of passive and active stereo. Image courtesy of [1].

2.1.2 ASUS Xtion PRO LIVE

The ASUS Xtion Pro Live camera has two cameras and an IR projector as shown in Fig. 2.2(a). According to the official website [10], it provides the RGB image with the maximum resolution of 1280×1024, while the depth can be alternated either VGA resolution (640×480 with 30fps) or QVGA (320×240 with 60fps). Its depth is reported to range from 0.8 to 3.5m, but we found in the experiments that the blind area was less and merely around 0.5m. Xtion Pro Live has been used throughout our experiments and we choose different configurations depending on the applications. When the depth super-resolution is required, the RGB images have the resolution of 1280×1024, otherwise we keep the same RGB resolution as the depth (640×480).

2.2 Shape from Shading & Photometric Stereo

The well-known shape from shading (SFS) problem was first introduced by Horn [11] in 1970 and then a large amount of literatures flooded in to develop the field. The idea of SFS is, knowing the light source position, one can estimate the shape or the surface of an object from one single grayscale image. This inverse problem of SFS is highly ill-posed, which has been illustrated in Fig. 2.3.

(a) Camera structure [10]

(b) RGB image

(c) Depth image

Figure 2.2: The structure of ASUS Xtion Pro Live and the RGB and depth images of an indoor scene acquired by it.

From the mathematical perspective, a SFS problem can be formulated as follows:

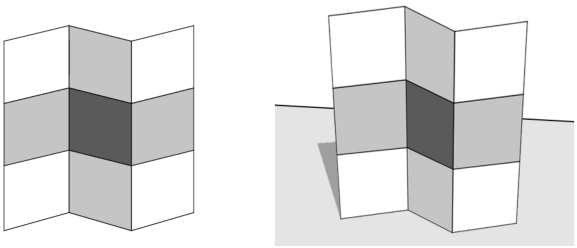
$$I = \rho S \tag{2.1}$$

where I is an intensity image, ρ is the reflectance (albedo) of the surface, and the S is the shading image. An example of such an image decomposition is shown in Fig. 2.4.

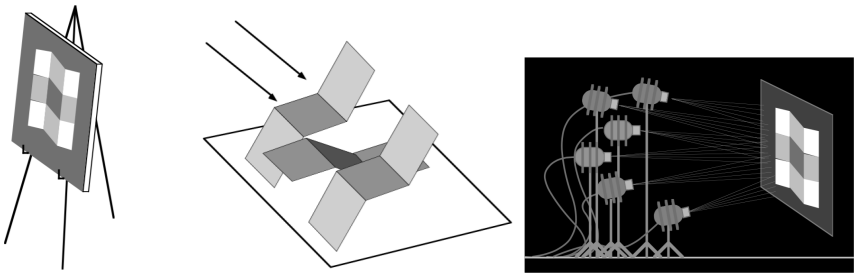
All SFS problem assumes the observed object follows the Lambert's cosine law, which says that the irradiance from a diffused object is proportional to the cosine between the point source light direction and the surface normal [13]. Based on the Lambert's cosine law, the Eq. 2.1 can be reformulated to the Lambertian reflectance model:

$$I = \rho \, \mathbf{l}^\top \mathbf{n} \tag{2.2}$$

which we can notice that the shading S is the inner product of the light direction and the surface normal. Thus, the task of SFS is to retrieve the shape (surface normal) from the shading based on the Lambertian reflectance model. Moreover, many state-of-the-art shape or depth refinement methods used an extension of Lambertian model called spherical harmonics



(a) An image (b) A possible explanation [12]



(c) painter's explanation (d) sculptor's explanation (e) Lighting designer's explanation

Figure 2.3: Various explanations for a twice-bent surface. Images are from [2]

I ρ S

Figure 2.4: The decomposition of a color toy panther. Images are from MIT intrinsic images dataset [3].

(SH) [14, 15] which can represent the illuminations more realistically. It has been shown that the first-order SH model (Eq. 2.3) can account for 87.5% of real world light so we applied it throughout the whole thesis.

$$I = \rho \left(\mathbf{l}^\top \mathbf{n} + \varphi\right) \tag{2.3}$$

where φ can be understood as the ambient light parameter.

The goal of SFS is to estimate the shape or the surface of object, which is represented by the surface normal $\mathbf{n}$. Two camera projection models are usually used for modelling the surface normal: orthographic and perspective projection. To derive $\mathbf{n}$ using an orthographic projection,

we consider projection along the z-axis [16]. Hence, A 3D point $P = (x, y, z)$ is mapped to the image point $p = (x, y)$. It should be noted that the depth z depends on the coordinate x and y. And we know the surface normal is orthogonal to the tangent plane in (x, y), which can be written as:

$$\mathbf{n}(P) \propto \partial_x P \times \partial_y P = \begin{pmatrix} 1 \\ 0 \\ z_x \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ z_y \end{pmatrix} = - \begin{pmatrix} z_x \\ z_y \\ -1 \end{pmatrix} \tag{2.4}$$

After normalizing and neglecting the negative sign, we acquire the unit-length surface normal with orthographic projection:

$$\mathbf{n}_{ortho} = \frac{1}{\sqrt{1 + \nabla z}} \begin{pmatrix} \nabla z \\ -1 \end{pmatrix} \tag{2.5}$$

For the more realistic perspective model, the 3D point P now becomes $\begin{pmatrix} (x - x_0)/f \\ (y - y_0)/f \\ z \end{pmatrix}$ and the corresponding normal is:

$$\mathbf{n}_{perspect} = \frac{1}{d} \begin{pmatrix} f\tilde{z}_x \\ f\tilde{z}_y \\ -1 - (x - x_0)\tilde{z}_x - (y - y_0)\tilde{z}_y \end{pmatrix} \tag{2.6}$$

where $\tilde{z} = \log z$, f the focal length, (x_0, y_0) the coordinates of principle points, and the normalizer $d = \sqrt{(fz_x)^2 + (fz_y)^2 + (-1 - (x - x_0)z_x - (y - y_0)z_y)^2}$.

From the definition of the surface normal, we can notice the SFS is an ambiguous problem. Even when the lighting and the albedo are known in Eq. 2.2, the inverse problem is ill-posed because the normal is 2 degrees of freedom. As we can see from Fig. **??**, the solution of SFS is ambiguous. Horn and Brooks [17] proposed the so-called integrability constraint $z_{xy} = z_{yx}$, which was the first constraint imposed on the filed of surface normal to make the SFS problem well-posed. Frankot and Chellapa [18] projected a non-integrable surface to the subspace spanning the valid smooth surface.

put the image from Pentland

Provided we have several images from the same view but with different illuminations, Eq. 2.1 can be modelled as:

$$\mathbf{I} = \mathbf{BL} \tag{2.7}$$

Assuming there are $n > 2$ (why = 2 when the albedo is also known?) images from various illumination conditions, $\mathbf{I}$ is the stack of all the intensity images, $\mathbf{B}$ corresponds to $\rho\mathbf{n}^\top$ and $\mathbf{L}$ represents n lighting. If the illuminations are known, we call this problem *calibrated* pho-

tometric stereo (PS), which was first introduced by Woodham [19] in 1980. The problem is over-constrained so the surface normals can be estimated using a simple least square. Some regions may sometimes suffer from the shadows for a certain illumination, so Forsythe and Ponce [20] formed a diagonal matrix to eliminate all those shadow points. Another interesting example of calibrated PS was proposed by Hernández *et al.* [21]. They controlled red, green, and blue three color lights from three directions and acquired the shape from only one color image, every channel of which was treated as a separate intensity image. This was one of the inspiration of our first proposed method RGB ratio model, which will be detailed in chapter 3.

However, the different lightings and the albedo are not always controlled or given, then we call this kind of problem *uncalibrated* photometric stereo. Hayakawa [22] found that the surface normals and the albedo could be recovered with a 3×3 linear transformation using singular value decomposition. Yuille and Snow [23] further reduced the number of ambiguities to three by adding the integrability constraint. The 3-parameter ambiguity is called gerenalized bas-relief (GBR) and now Eq. 2.7 can be represented as

$$\mathbf{I} = \mathbf{B}\mathbf{A}^{-1}\mathbf{A}\mathbf{L} \tag{2.8}$$

where A is the GBR matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mu & \nu & \lambda \end{pmatrix} \tag{2.9}$$

There are a large amount of research work trying to solve the GBR ambiguity in uncalibrated PS. Alldrin *et al.* [24] used the prior emphasizing that the albedo distribution should have a low entropy, to resolve the ambiguity. Favaro and Papadhimitri [25] eliminated the ambiguity by exploiting the spatial maximum points of the inner product between the normals and the lights. They also found out the solution is unique when the normal is constructed under the perspecitve projection instead of the orthographic projection [26]. Quéau *et al.* [27] estimated the GBR parameters by imposing the total variation norm.

In terms of depth, the GBR ambiguity is equivalent to the equation [27]:

$$z(x, y) = \lambda z(x, y) + \mu x + \nu y \tag{2.10}$$

Based on the equation, we find out that the GBR ambiguity still exists for the task of depth estimation. However, since the rough depth will be given as the input from RGB-D camera, the GBR ambiguity problem encountered by uncalibrated PS is resolved automatically. Now we will discuss some state-of-the-art depth refinement approaches.

2.3 Depth and Shape Refinement

In recent years, there are a large amount of literatures focusing on the depth or shape refinement based on either SFS (only use one single image) or PS (multiple images with different illuminations). They are collectively called shading-based methods. We will discuss these two streams respectively in the following.

2.3.1 SFS-based methods

Since SFS iteslf uses only one input image, it suffers from some intrinsic ambiguities that we have mentioned in the last section even when the light and the albedo are specified, so there will be more than one possible solution. Now, although a rough depth map is given, the illumination and the albedo are unknown, so some regularization terms have to be imposed in order to acquire an exact solution from the inverse problem.

Han *et al.* [4] presented a framework which combines a global lighting model using the given color and depth with the help of SH model, with a local lighting model which varies spatially. The surface orientations should obey the integrability constraint on the smooth surface so they enforced the constraint by penalizing the curl of local neighborings. However, the albedo in their method is assumed to be uniform. To handle the multi-albedo objects, they had to apply another intrinsic image decomposition algorithm [28] and k-mean clustering to group the albedos into some areas with constant values inside. Such framework is not only unrealistic but also very time-consuming so not able to be adapted to the real-world applications.

Yu *et al.* [29] iteratively update SH lighting and the albedo using the initial depth and the refined the shape with the estimated lighting and the relative albedo. They performed mean-shift clustering to segment the input RGB image into small regions with uniform albedo, and then obtained the relative albedos among various segmented regions. To fill in the missing depth information, a constrained texture synthesis and patch-based repairing scheme was applied. In constrast, we efficiently apply an basic image inpainting approach as the pre-processing and the results are also satisfying.

Wu *et al.* [30] extended their previous offline shading-based refinement work [31] to online shape refinement with highly parallel scheme and the help of GPU. They first calculated the 2nd-order SH parameters with the assumption of uniform albedo, and then estimated the albedo by simply dividing RGB image with the shading term. We will discuss in chapter 3 this process may leads to the severe albedo overfitting problem such that the albedo estimation is not correct. The shape is then refined in real-time by finding the surface that minimizes the difference between the shading and intensity image gradients. Thereafter, the coarse depth map was directly refined with smoothness and temporal constraint on the video by using a

Gauss-Newton solver on GPU.

Kim *et al.* [32] used a joint energy to estimate the depth, albedo and the light with smoothness regularization terms. An anisotropic Laplacian constraint on chromaticity was introduced for albedo and a local smoothness and bas-relief ambiguity similar to [33] constraints are imposed for depth. Based on our testing implementation, it has turned out that the Laplacian on chromaticity of the image cannot provide satisfying albedos for the small indoor environment. What's more, it is a very tedious process to tune all the parameters for the constraints.

RGBD-Fusion method from Or-El *et al.* [5] can also deal with natural illumination conditions and make the depth recovery task in real time under GPU. They imposed the constraints not only on the albedo and depth estimation but also pixel-wise ambient lighting. Their method does not really converge because of their way of handling the nonlinearity. This inspired us to propose a new RGB ratio model to eliminate the nonlinearity.

Or-El's following work [34] can deal with specular objects with the help of IR camera and a more complicated reflectance model than spherical harmonics. In constrast, our proposed multi-light method still uses a Lambertian diffused reflectance model but can handle the specularity.

From what we have discussed, SFS methods are often limited to the uniform or constant albedo. For the sake of handling multi-albedo cases, some SFS methods [4,29] adapted segmentation methods to divide the input image into some constant albedo part, but the real-world objects are usually with complex multi-albedo and small regions, which makes the segmentation not accurate or incorrect. Some other methods [5,30,32,34] tries to add some piecewise smooth constraints on the albedo but never really acquired satisfying outcome. Therefore, SFS-based methods using only one single image have the difficulty to separate the albedo from the surface normal, which may lead to the wrong depth estimation.

2.3.2 PS-based methods

Another category of shading-based depth refinement is PS-based methods. With the help of multiple images acquired from various illuminations, these approaches can resolve the ambiguities tolerated by SFS methods and have a better performance in the separation between the albedo and surface normal.

Haque *et al.* [35] proposed the a method to reconstruct the shape and refine the depth using an IR camera without the need of RGB camera. However, similar to many other multi-view photometric reconstruction approaches [36,37], they assumed the albedo is restricted to uniform and thus, it is feasible to use it for multi-albedo objects.

Their follow-up work from Chatterjee and Govindu [38] decomposed the input images under different illuminations with a standard photometric stereo manner. They used an iterative reweighted method to approximate the Rank 3 radiometric brightness matrix, then factorize

it into the corresponding lighting, albedo and surface normal. They can cope with the multi-albedo objects but still have to use the IR images instead of RGB images. In this case, at least one extra infrared light source is always required, while in our case only a cheap LED light or even just the flashlight on a phone is enough. Moreover, since the IR camera in ASUS Xtion Pro Live is limited to the resolution 640×480, while the RGB camera can reach 1280×1024, their approach can not perform depth super-resolution task like our multi-light method.

Wu *et al.* [39] uses a second-order spherical harmonics to model the general illuminations and use the shading constraint to help improve the object reconstruction. this method is extended to [31] whose results are furthered improved by integrating a weak temporal prior on lighting, albedo and the shape.

In this thesis, we proposed two shape refinement methods based on uncalibrated photometric stereo. Similar to [21], we firstly used RGB 3 LED lights for the active illuminations so we can treat the every channel of the obtained color image as an intensity image with lights from a different direction. Another proposed method needed only one white LED. With the the RGB-D camera's angle of view fixed, we manually moved the LED lights while the images are being taken. Moreover, the shading-based method have never been applied for the depth map super-resolution before. We successfully adapted our methods to depth super-resolution and achieved very pleasing outcomes.

Chapter 3

Methodology

Many computer vision applications such as 3D reconstruction or visual SLAM require the depth information from RGBD cameras. However, the results of these applications are often not unsatisfying because of the low quality of the depth acquisition from the cheap cameras. It would be gratifying if we can improve the depth quality from a consumer camera. Therefore, we discovered the field of depth refinement techniques and proposed our methods.

In this chapter, we first introduce some pre-processing techniques to fill the missing areas and reduce the noise of the input depth image. Then, we describe in detail one of the state-of-the-art depth refinement method from Or-El *et al.* [5] which we have chosen to implement as a starting point. A proposed method based on an RGB ratio model is then followed and introduced to eliminate the nonlinearity in most of the modern depth enhancement methods. Finally, another proposed technique which does not require any regularization terms is presented. This method has exhibited the ability to deal with the objects with complicated albedos and extension to the super-resolution for depth images.

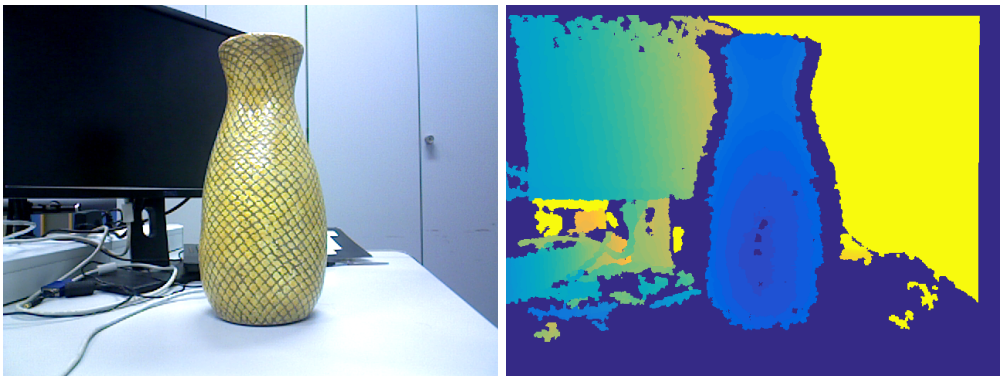
3.1 Pre-Processing

The first step for most of the image processing tasks is to pre-process the initial input image. Due to the hardware limitation of inexpensive RGB-D sensors, there usually exist holes with missing values on the depth images. Also, the depth data is often noisy so we need to do denoising and acquire a relatively smooth surface.

In this part, we will describe respectively the depth inpainting and denoising algorithm that we use for our pre-processing.

3.1.1 Depth inpainting

Image inpainting itself is a very mutual area and has been widely applied as a useful tool for many modern computer vision applications, e.g, restore the damaged parts of ancient paintings, and remove unwanted texts or objects in a photography [40]. Since the idea of image inpainting is to automatically replace the lost or undesired parts of an image with the neighbouring information by interpolating, we were inspired to apply it to fill in the missing depth information (Fig. 3.1).



(a) RGB image (b) Input depth image

Figure 3.1: The input RGB and depth image of a vase. The depth map in (b) is visualized using color from blue (near) to yellow (far).

It should be noted that, the depth inpainting is applied to the input rough depth map so there is no need to use some powerful and advanced algorithms. The only request is to fill the missing areas with inexpensive computational time.

The general mathematical form of a classic inpainting algorithm [40] can be written as follows

$$I^{t+1}(i,j) = I^t(i,j) + \mu U^t(i,j), \forall (i,j) \in \Omega \tag{3.1}$$

where $I(i,j)$ is the pixel value in image I, t is the artificial time step, μ is the updating rate, U is the update information and Ω is the area with missing information.

To build the update map U in each time step, there are two principles that [40] follow. One is the inpainted values inside Ω should be as smooth as possible. The other is the lines reaching the edge of Ω should be continued and cross the missing area, while the values in Ω should be propagated from the nearest neighbours of Ω along the lines.

Again, due to the fact that our input depth images have poor quality, the lines arriving at

the boundary $\delta\Omega$ may be incorrect or produced by the noises. Thus, it is reasonable that our initial depth inpainting problem focuses on the smooth propagation from the neighbours and fill in the holes.

In each pixel (x_0, y_0) inside Ω, U can be modelled as a discrete four-neighbour Laplacian operator:

$$U(x_0, y_0) = \Delta I = 4I(x_0, y_0) - I(x_0 + 1, y_0) - I(x_0 - 1, y_0) - I(x_0, y_0 + 1) - I(x_0, y_0 - 1) \tag{3.2}$$

Now the inpainting problem in Eq. 3.1 can be represented as a minimization problem:

$$\min \iint_{\Omega} |U(x, y)|^2 dxdy \tag{3.3}$$

This energy function can be reformulated to a typical linear equation in matrix form:

$$\mathbf{Ax} = \mathbf{b} \tag{3.4}$$

Assuming n is the number of pixel inside Ω and m is the sum of n and the number of neighbouring pixel around the boundary $\delta\Omega$, $\mathbf{A}$ is a $m \times n$ Laplacian matrix, $\mathbf{b}$ is a $m \times 1$ vector containing all the known boundary depth values and the 0 inside Ω. Solving the linear equation with simple least square method, we can acquire the inpainted values. With our this naive image inpainting algorithm, we can fill the holes on the depth image as shown in Fig. 3.2.

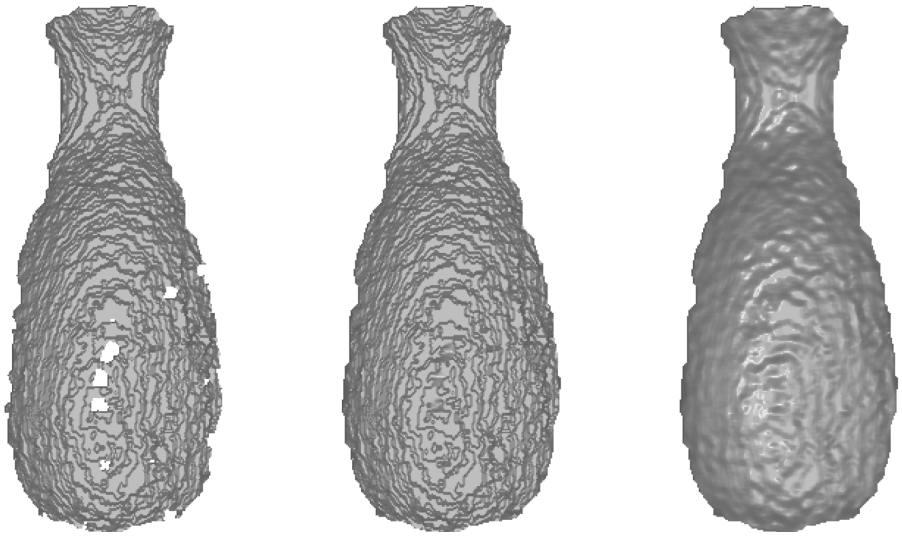
3.1.2 Depth denoising

the depth images acquired from the RGB-D cameras with moderate price usually contain various noises. As a standard pre-processing method, the image denoising technique is also applied to our input inpainted depth map. Similar to the state-of-the-art depth refinement methods [4, 5, 29, 34, 35, 41], bilateral fitering [42] is used as our depth pre-processing smoother.

The advantage of bilateral filter is reducing the noise while preserving the edge in the input image. More than a regular Gaussian smooth filter, which uses only the difference of the image values (depth in our case) between the center pixel the neighbours, the bilateral filter also utilizes the space difference as a reference to build up the weighting function. The filtered pixel value can be modelled as a weighted sum of neighbouring pixels:

$$\hat{I}(\mathbf{x}) = \frac{1}{W} \sum_{\mathbf{y} \in \mathcal{N}} I(\mathbf{y}) e^{-\left(\frac{\|I(\mathbf{x}) - I(\mathbf{y})\|^2}{2\sigma_r^2} + \frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma_d^2}\right)} \tag{3.5}$$

where $\hat{I}(\mathbf{x})$ is the filtered value at pixel $\mathbf{x} = (x, y)$, $\mathcal{N}$ represents the neighbouring pixels with $\mathbf{x}$ in the center, and W is the sum of the all the neighbouring weights centering in $\mathbf{x}$. The smoothed result on our input depth image is shown in Fig. 3.2.



(a) Input depth (b) After inpainting (c) After smoothing

Figure 3.2: Illustrations for the pre-processing on the depth of the vase.

After the pre-processing procedure, we have an initial smooth and inpainted depth image. It will be used as the input of all the depth refinement methods detailed in the following sections.

3.2 RGBD-Fusion Like method

RGBD-Fusion is a state-of-the-art depth recovery method proposed by Or-El *et al.* [5] in 2015. This novel method is adequate for natural scene illumination and able to enhance the depth map much faster than other methods. It is reasonable to gain a comprehensive understanding in the field of depth refinement by implementing this method.

It is worth mentioning that we didn't just follow the paper step by step without injecting any our own ideas. For example, instead of estimating the pixel-wise ambient light with a separate energy function, we jointly calculated all four first-order spherical harmonics parameters (3 for point-source light direction and 1 for ambient light) with a simple fast least square. In this case, we reduced the number of tuning parameters from 8 to 5 but the results have the only negligible difference. And throughout the whole estimation process of light, albedo and depth, we only used the information within the given mask which also speeded up the algorithm. This

is the reason we call our first method "RGB-Fusion Like" method.

The natural uncalibrated illumination condition means the light is no longer a point light source, thus a Lambertian model is not sufficient. Basri and Jocobs [14] have found that low order spherical harmonics (SH) model can well set out the irradiance of the diffused objects under the natural scene. More specifically, the first-order SH model can capture 87.5% of natural lighting, whose form is extended from the Lambertian model:

$$I(x,y) = \rho(x,y)(\mathbf{l}^\top \mathbf{n}(x,y) + \varphi) \tag{3.6}$$

where $I : \mathcal{M} \to \mathbb{R}^C$ is the irradiance of the objects, which is represented as the intensity values. $\rho : \mathcal{M} \to \mathbb{R}^C$ is the albedo, $\mathbf{l}^\top = \begin{pmatrix} l_x & l_y & l_z \end{pmatrix}$ describes the light direction and φ represents the ambient light. $\mathbf{n} : \mathcal{M} \to \mathbb{R}^3$ is the surface normal, which is dependent on the depth z. We define that $C = 1$ represents the grayscale image while $C = 3$ for color image. $(x, y) \in \mathcal{M}$ represents the pixel coordinate inside the given mask $\mathcal{M}$ of an object. Eq. 3.6 can be rewritten as:

$$I(x,y) = \rho(x,y)\ \mathbf{s}^\top \tilde{\mathbf{n}}(x,y) \tag{3.7}$$

where

$$\mathbf{s} = \begin{pmatrix} \mathbf{l} \\ \varphi \end{pmatrix} \quad \tilde{\mathbf{n}}(x,y) = \begin{pmatrix} \mathbf{n}(x,y) \\ 1 \end{pmatrix} \tag{3.8}$$

$\mathbf{s}$ is the first-order SH parameters. It should be mentioned that the 1st-order SH model is used as the fundamental model throughout the whole methodology part.

After introducing the preliminary knowledge, the overall energy function for the RGBD-Fusion like method which can jointly estimate lights, albedo and depth is described below:

$$\begin{aligned} E(\rho, z, \mathbf{s}) = \sum_{(x,y)\in\mathcal{M}} \sum |I(x,y) - \rho(x,y)\mathbf{s}^\top \tilde{\mathbf{n}}(x,y)|^2 + \lambda_\rho \sum_{(x,y)\in\mathcal{M}} \sum_{k\in\mathcal{N}} |\omega_k(x,y)(\rho(x,y) - \rho_k)|^2 \\ + \lambda_z \sum_{(x,y)\in\mathcal{M}} \sum |z(x,y) - z_0(x,y)|^2 + \lambda_l \sum_{(x,y)\in\mathcal{M}} \sum |\Delta z(x,y)|^2 \end{aligned} \tag{3.9}$$

For the sake of simplicity, we will use $\|\cdot\|_2^2 = \sum_{(x,y)\in\mathcal{M}} \sum (\cdot)^2$ to reshape the equation, and then I, z and ρ are vectorized to $\mathbb{R}^m$ within the mask, while $\tilde{\mathbf{n}} \in \mathbb{R}^{m\times 4}$ with each row as $\begin{bmatrix} \mathbf{n}(x,y)^\top & 1 \end{bmatrix}$. m is the number of pixel inside the mask $\mathcal{M}$. So Eq. 3.9 can be reformulated as:

$$E(\rho, z, \mathbf{s}) = \|I - \rho \cdot\ \tilde{\mathbf{n}}(z)\mathbf{s}\|_2^2 + \lambda_\rho \| \sum_{k\in\mathcal{N}} \omega_k(\rho - \rho_k)\|_2^2 + \lambda_z \|z - z_0\|_2^2 + \lambda_l \|\Delta z\|_2^2 \tag{3.10}$$

The function consists of an SFS term, an albedo anisotropic Laplacian term, a depth data fidelity term and a depth isotropic Laplacian term. Now we will go through the details step by step.

3.2.1 Light estimation

Here the unit length surface normal $\mathbf{n}$ is formulated with orthographic projection, i.e.

$$\mathbf{n}(x,y) = \frac{1}{\sqrt{1+|\nabla z(x,y)|^2}} \begin{pmatrix} \nabla z(x,y) \\ -1 \end{pmatrix} \tag{3.11}$$

$\nabla z(x,y)$ represents the gradient of depth image $z(x,y)$ in x and y directions. Since we have the input depth from pre-processing, initial $\mathbf{n}_0$ is known.

To compute the spherical harmonics parameters, we assume the albedo ρ equals to 1 for each pixel. Since there are known intensity values and surface normal in each pixel within the mask, we will have an overdetermined least square problem from the energy in Eq. 3.10:

$$\min_{\mathbf{s}} \|\tilde{\mathbf{n}}\mathbf{s} - I\|_2^2 \tag{3.12}$$

This process only need to be applied once at the beginning of the process since the least squares is not sensitive to the details on the surface, thus the estimation from the smooth surface is enough.

3.2.2 Albedo estimation

As mentioned in chapter 2, many depth recovery techniques based on SFS or PS assume constant or uniform albedo. Such assumption does not fit in with the real-world objects so they perform poorly on the shape estimation for multi-albedo cases. In order to acquire a satisfying shape outcome, an effective multi-albedo estimation process is a matter of importance.

We know from Eq. 3.6 that, assuming we have the knowledge of input intensity and estimated shading, the albedo image can be directly obtained from I/S. However, due to the fact that both input image I and the surface normal $\mathbf{n}$ are noisy, such albedo is prone to the overfitting, which makes the acquired albedo contain all the undesired spatial layout details. To resolve the overfitting problem, we should impose some restrictions on the estimation of albedo. Plenty of real-world objects has piecewise smooth appearance as shown in Fig. 3.3, which means most pieces of a layout are dominated by certain colors. Therefore, a prior that emphasizes the piecewise smoothness on the albedo should be defined.

The albedo of an object can be roughly divided into several pieces with different intensities,

which can be treated as the image segmentation problem to some extent. Thus, we should refer to some classic variational segmentation methods and adapt the edge preserving smoothness term to our problem. Similar to the idea in [43], an anisotropic Laplacian term is imposed to estimate the albedo. Now, the SH parameters $\mathbf{s}$ and the surface normal $\mathbf{n}$ are freezed and the overall regularized minimization problem in Eq. 3.10 is:

$$\min_{\rho} \|\rho \cdot \tilde{\mathbf{n}}\mathbf{s} - I\|_2^2 + \lambda_\rho \| \sum_{k \in \mathcal{N}} \omega_k (\rho - \rho_k)\|_2^2 \tag{3.13}$$

where k indicates the neighbouring index of a certain pixel, which 4-connected set is chosen for $\mathcal{N}$ in our case. The weight ω_k is defined as below, and it is dependent on two parameters σ_I and σ_z which accounts for the discontinuity in both intensity and depth.

$$\omega_k = \exp\left(- \frac{\|I - I_k\|_2^2}{2\sigma_I^2} - \frac{\|z - z_k\|_2^2}{2\sigma_z^2} \right) \tag{3.14}$$

3.2.3 Depth enhancement

After acquiring the first-order spherical lighting parameters $\mathbf{s}$ and the albedo ρ, we can refine our depth with the help of Eq. 3.6 and Eq. 3.11. Now our minimization problem with respect to the depth z in Eq. 3.10 can be written as below. The data fidelity term is applied to resolve the SFS ambiguities and enables our refined surface close to the input. The Laplacian smoothness term makes sure that there is no strong discontinuity in the output.

$$\min_{z} \|\rho \cdot \tilde{\mathbf{n}}(z)\mathbf{s} - I\|_2^2 + \lambda_z \|z - z_0\|_2^2 + \lambda_l \|\Delta z\|_2^2 \tag{3.15}$$

where z_0 is the input depth and Δ represents the Laplacian operator. It can be easily noticed that this introduced function is non-linear because the normal in our SFS term contains a denominator related to the depth gradient. Many optimization methods can be applied to solve the nonlinear problem, e.g. Levenberg-Marquardt algorithm or ADMM, but they are not suitable for our application due to expensive computational time. Here a "fixed point" method which is similar to iteratively reweighted least square (IRLS) has been introduced to deal with our problem efficiently.

The idea of the fixed-point approach is in each iteration, the normalizer in the surface normal can be treated as a weighting term and determined by the depth from the last iteration. With the help of this trick, the normalizer is known and Eq. 3.15 is linear again. We can solve the linear system using any fast linear optimization method. In each iteration t, this process can

be represented element-wise as follows:

$$\begin{aligned} \mathbf{n}(z^{(t)}, z^{(t-1)}) &= w(z^{(t-1)}) \begin{pmatrix} \nabla z^{(t)} \\ -1 \end{pmatrix} \\ w(z^{(t-1)}) &= \frac{1}{\sqrt{1 + |\nabla z^{(t-1)}|^2}} \end{aligned} \tag{3.16}$$

And now the depth refinement problem in Eq. 3.15 is reformulated as below in each iteration:

$$\min_{z^{(t)}} \|\rho \cdot \tilde{\mathbf{n}}(z^{(t)}, z^{(t-1)})\mathbf{s} - I\|_2^2 + \lambda_z \|z^{(t)} - z_0\|_2^2 + \lambda_l \|\Delta z^{(t)}\|_2^2 \tag{3.17}$$

As long as the energy decreases in each iteration, the process is repeated.

To sum up the approach in this section, it should be noted that the SFS term was used as a core in all light, albedo and depth estimation in the overall energy. The whole process of the RGBD Fusion-Like method has been described in Alg. 1 and some real-world results are shown in Fig. 3.3.

Algorithm 1 RGBD-Fusion Like Depth Refinement

Input: Initial depth image z_0, RGB image I

1: Estimate the SH parameter, $\mathbf{s} = \arg\min_{\mathbf{s}} E(\rho = 1, z_0)$ {Eq. 3.12}
2: Estimate the albedo, $\rho = \arg\min_{\rho} E(z_0, \mathbf{s})$ {Eq. 3.13}
3: t = 1, $z^{(t-1)} = z_0$
4: **while** $E(\rho, z^{(t)}, \mathbf{s}) - E(\rho, z^{(t-1)}, \mathbf{s}) < 0$ **do**
5: $\quad z^{(t)} = \arg\min_{z} E(\rho, z, \mathbf{s})$ {Eq. 3.17}
6: $\quad t := t + 1$
7: **end while**

Output: Refined depth image $z^{(t)}$

3.2.4 Limitations

Although our RGBD-Fusion like method works moderately well in some real cases, it is not difficult to find the limitations and improve correspondingly.

- the surface normal modelled by the orthographic projection is merely an ideal case which is not really in line with the real world camera model. And the intrinsic parameters such as the focal length and the coordinate of the principle point are either usually given as a preliminary knowledge, or obtained from camera calibration without much effort. Hence,

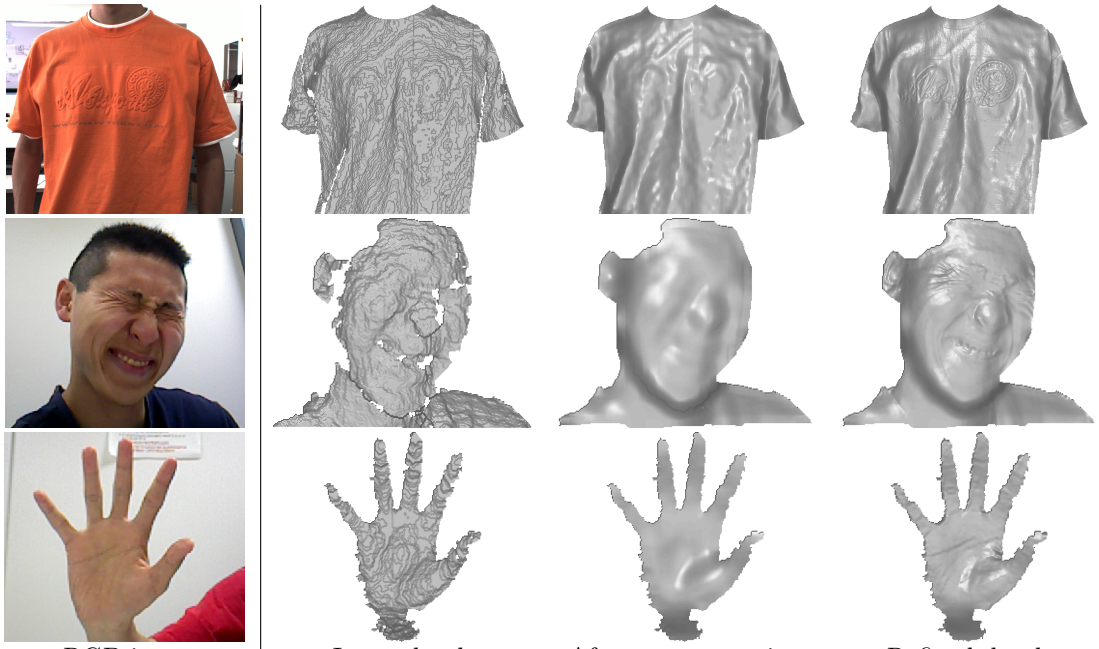


Figure 3.3: Illustrations for our implementation of RGBD-Fusion Like method. Top row is a T-shirt from [4]. Middle and bottom row are author's face and palm.

it is reasonable to formulate the surface normal with the perspective projection model.

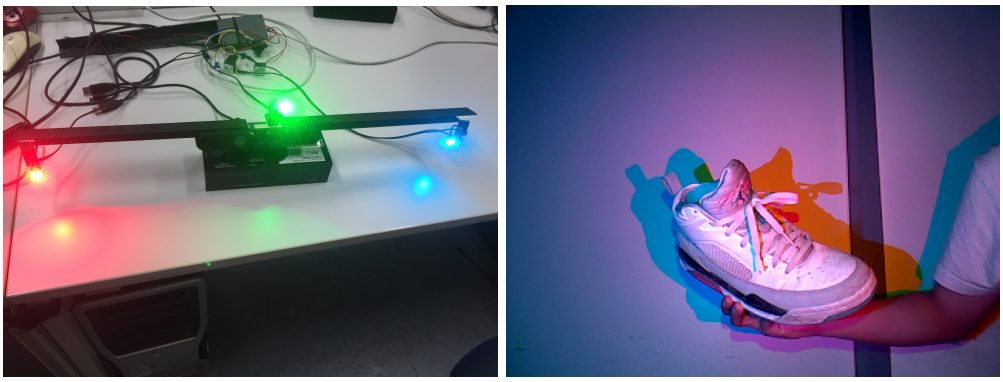
- In our RGBD-Fusion like method, only the intensity is applied because the values in RGB channels are more or less the same under the natural scene illumination. When we estimated the SH lighting parameters and the albedo in 3 channels separately, the results are quite similar to each other. So using all three channels rather than just the grayscale image will not provide much extra information and improve the depth enhancement. Instead, it will just decelerate the whole algorithm. We ought to find a way to take better advantages of all three channels in a photometric stereo manner.

- The most important inspiration for us to propose the new RGB ratio model in the next section is, the RGBD-Fusion like method was not always convergent in terms of depth enhancement part because of the fix-point optimization method. In the 4th line of the Alg. 1, we force the iteration to stop when the energy for the depth refinement starts increasing. This is due to the reason that the fixed-point method actually tries to solve the non-linearity in a tricky way, which is mathematically not totally correct. Therefore, we thought of the idea of RGB ratio model, which can eliminate the denominator inside the normal and promise a real linear problem.

3.3 Proposed method I: RGB Ratio Model

According to the limitations in the last section, we thought of the idea of RGB ratio model. First of all, we replace the orthographic projection model with the perspective one. And then, to fully use the information of the RGB three channels while eliminating the non-linearity in the objective function in the depth refinement, we use the ratio model between every two channels among the three.

It should be noted that we need to add active R, G and B 3 LED lights for the sake of emphasizing the difference among RGB channels. The green LED is installed in the middle with the red and blue ones on the two sides of ASUS Xtion Pro Live camera (both are around 30 cm to the green LED). The hardware setup of our system and a color image taken with such setup are illustrated in Fig. 3.4.



(a) LED lights setup (b) A color image with our lighting setup

Figure 3.4: Illustrations for the RGB LED setup and the corresponding image.

Now to derive our new ratio model, we treat each channel of the color image I as a single intensity image, denoted by I_R, I_G, I_B. Therefore, 3 equations can be obtained from Eq. 3.6.

$$\begin{aligned} I_R &= \rho_R(\mathbf{l}_R^\top \mathbf{n} + \varphi_R) \\ I_G &= \rho_G(\mathbf{l}_G^\top \mathbf{n} + \varphi_G) \\ I_B &= \rho_B(\mathbf{l}_B^\top \mathbf{n} + \varphi_B) \end{aligned} \tag{3.18}$$

Using R and G channel as an example, we acquire the ratio model between R and G:

$$\frac{I_R - \rho_R\varphi_R}{I_G - \rho_G\varphi_G} = \frac{\rho_R \mathbf{l}_R^\top \mathbf{n}}{\rho_G \mathbf{l}_G^\top \mathbf{n}} \tag{3.19}$$

Similarly, we can acquire another two ratio models which are between green and blue, and blue and red channels respectively. It can be noticed from Eq 3.19 that, the non-linearity problem mentioned before has been solved because the denominator in the surface normal $\mathbf{n}$ is cancelled out. Also, our normal is derived from perspective camera model and can be represented as a function of $\log z$. For the sake of simplicity we directly represent $z = \log z$ and redefine $\mathbf{n}$ without the normalizer in the following part.

$$\mathbf{n}(x,y) = \begin{pmatrix} fz_x(x,y) \\ fz_y(x,y) \\ -1 - \tilde{x}z_x(x,y) - \tilde{y}z_y(x,y) \end{pmatrix} \tag{3.20}$$

where f is the focal length, $(\tilde{x}, \tilde{y}) = (x - x_0, y - y_0)$, with (x_0, y_0) the coordinates of principle points and (x, y) the coordinate of a pixel inside the given mask, $[z_x, z_y]$ is the gradient of depth z.

The overall energy for the proposed RGB ratio model method is:

$$\begin{aligned} E(\mathcal{P}^{(t)}, z^{(t)}, \mathbf{s}^{(t)}) = \|Ratio(\mathcal{P}^{(t)}, z^{(t)})\|_2^2 + \lambda_z||z^{(t)} - z_0||^2 \\ + \lambda_\rho^1\|\omega\nabla\mathcal{P}^{(t)}\|^2 + \lambda_\rho^2\|\mathcal{P} - \mathcal{P}^{(t-1)}\|^2 + \sum_c \|\rho_c\, \mathbf{s}_c^\top \tilde{\mathbf{n}} - I_c\|_2^2,\ c \in \{R,G,B\} \end{aligned} \tag{3.21}$$

where $\mathcal{P}$ is the stack of RGB albedo. This energy is composed of a proposed ratio SFS term, a depth fidelity term, an albedo smoothness term, an albedo fidelity term and a SH estimation term. Now we will explain our proposed algorithm based on the new ratio model.

3.3.1 Algorithm details

Similar to the RGBD-Fusion Like method, the algorithm is separated into 3 parts: light estimation, albedo estimation and depth enhancement. However, our new method requires an initial estimation of the color albedo as the input of our iterative method, and hence, we calculate the initial SH parameters $\mathbf{l}^0$ with Eq. 3.6 and the color albedo $\mathcal{P}^0$ with Eq. 3.13 using the old model. Noted that the initial estimation is performed with respect to all RGB three channels.

Albedo refinement: with the acquired ρ^0 and $\mathbf{l}^0$, we can start the iterative refinement process. In order to refine the color albedo with our ratio model, in each iteration, we need to

reshape the ratio model described in Eq. 3.19 as follows:

$$\begin{aligned} I_G\mathbf{l}_R^\top\mathbf{n}\rho_R - I_R\mathbf{l}_G^\top\mathbf{n}\rho_G &= \rho_R\rho_G(\varphi_G\mathbf{l}_R^\top\mathbf{n} - \varphi_R\mathbf{l}_G^\top\mathbf{n}) \\ I_B\mathbf{l}_G^\top\mathbf{n}\rho_G - I_G\mathbf{l}_B^\top\mathbf{n}\rho_B &= \rho_G\rho_B(\varphi_B\mathbf{l}_G^\top\mathbf{n} - \varphi_G\mathbf{l}_B^\top\mathbf{n}) \\ I_R\mathbf{l}_B^\top\mathbf{n}\rho_B - I_B\mathbf{l}_R^\top\mathbf{n}\rho_R &= \rho_B\rho_R(\varphi_R\mathbf{l}_B^\top\mathbf{n} - \varphi_B\mathbf{l}_R^\top\mathbf{n}) \end{aligned} \tag{3.22}$$

For each pixel, we can reformulate the Eq. 3.22 to a matrix form:

$$\begin{pmatrix} I_G\mathbf{l}_R^\top\mathbf{n} & -I_R\mathbf{l}_G^\top\mathbf{n} & 0 \\ 0 & I_B\mathbf{l}_G^\top\mathbf{n} & -I_G\mathbf{l}_B^\top\mathbf{n} \\ -I_B\mathbf{l}_R^\top\mathbf{n} & 0 & I_R\mathbf{l}_B^\top\mathbf{n} \end{pmatrix} \begin{pmatrix} \rho_R(x,y) \\ \rho_G(x,y) \\ \rho_B(x,y) \end{pmatrix}_{3\times 1} = \begin{pmatrix} \rho_R\rho_G(\varphi_G\mathbf{l}_R^\top\mathbf{n} - \varphi_R\mathbf{l}_G^\top\mathbf{n}) \\ \rho_G\rho_B(\varphi_B\mathbf{l}_G^\top\mathbf{n} - \varphi_G\mathbf{l}_B^\top\mathbf{n}) \\ \rho_B\rho_R(\varphi_R\mathbf{l}_B^\top\mathbf{n} - \varphi_B\mathbf{l}_R^\top\mathbf{n}) \end{pmatrix} \tag{3.23}$$

This small linear system can be generalized to a big sparse linear system denoted as $\mathbf{A}_\rho \cdot \mathcal{P} = \mathbf{b}_\rho$. The structure of this equation can be found in Appendix A. Here, $\mathcal{P}$ represents the stack of RGB three albedos.

To acquire the RGB albedos, some regularization terms are required similar to Eq. 3.13. Now in each iteration, we fix the normal and the SH parameters, the minimization problem of color albedo in Eq. 3.21 in each iteration now becomes:

$$\mathcal{P}^{(t)} = \arg\min_{\mathcal{P}} \|\mathbf{A}_\rho^{(t-1)}\mathcal{P} - \mathbf{b}_\rho^{(t-1)}\|^2 + \lambda_\rho^1\|\omega\nabla\mathcal{P}\|^2 + \lambda_\rho^2\|\mathcal{P} - \mathcal{P}^{(t-1)}\|^2 \tag{3.24}$$

where the weight $\omega = \begin{pmatrix} \omega_R \\ \omega_G \\ \omega_B \end{pmatrix}$, which can be denoted as:

$$\omega_c = \exp(-\frac{\sigma_c||\nabla I_c||^2}{\max||\nabla I_c||^2}), \quad c \in \{R, G, B\} \tag{3.25}$$

σ_c is a tuning parameter for each channel c. We can notice from Fig. 3.5 the importance of imposing the weight ω. Without the weight, the isotropic smoothness regularization will not take care of the boundary of the albedo, which leads to the bad depth enhancement.

There are three interesting aspects of the albedo estimation which worth having a few more words:

1. One observation about Eq. 3.23 is that, if the SH parameters are the same among the three channels, the right side of the equal sign becomes 0. This is one of the reasons why we really need to set up 3 LED lights with a distance to each other, which will provide us with enough difference on the light directions.

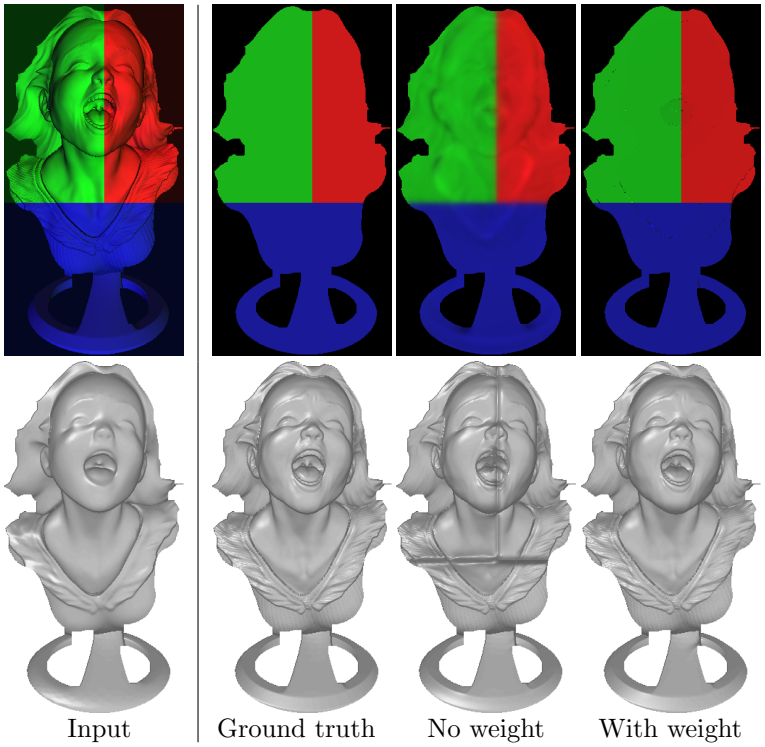


Figure 3.5: Illustrations for the importance of the weight ω inside regularization term in Eq. 3.24 when estimating the albedo. Top: first one is the input color image and the rest three are the albedos. Bottom: 3D shape from depth. Noted that here the light parameter is given.

2. Instead of using anisotropic Laplacian regularization in the RGBD Fusion Like method, the smoothness term in Eq. 3.24 only takes the use of the gradient of ρ with a weight only depending on the RGB image's gradient. It takes fewer efforts to build such a smoothness term than the anisotropic term but turns out the acquired albedo is still satisfying.

3. If we don't use a data fidelity term $\|\mathcal{P} - \mathcal{P}^{(t-1)}\|^2$, the albedo will get increasingly dark after several iterations. This is due to the fact that there also exist the RGB albedos in $\mathbf{b}_\rho$, so $\rho = 0$ will become the solution of our ratio model term. Therefore, adding the data term can not only avoid such problem but help refine the albedo iteratively.

Depth refinement: After acquiring the color albedo in time step t, we are going to refine the depth with the help of the ratio model. First we reshape Eq. 3.19 with the surface normal

$\mathbf{n}$ as the argument:

$$\begin{aligned}\rho_G(I_R - \rho_R\varphi_R)\mathbf{l}_G^T\mathbf{n} - \rho_R(I_G - \rho_G\varphi_G)\mathbf{l}_R^T\mathbf{n} = 0\\ \rho_B(I_G - \rho_G\varphi_G)\mathbf{l}_B^T\mathbf{n} - \rho_G(I_B - \rho_B\varphi_B)\mathbf{l}_G^T\mathbf{n} = 0\\ \rho_R(I_B - \rho_B\varphi_B)\mathbf{l}_R^T\mathbf{n} - \rho_B(I_R - \rho_R\varphi_R)\mathbf{l}_B^T\mathbf{n} = 0\end{aligned} \tag{3.26}$$

since the normal $\mathbf{n}$ now is a function of z, Eq. 3.26 can be actually simplified as below (the derivation details can be found in Appendix A):

$$\Psi z = 0 \tag{3.27}$$

When the estimated color albedo and light are fixed, the depth refinement problem in Eq. 3.21 is:

$$z^{(t)} = \arg\min_z ||\Psi z||^2 + \lambda_z||z - z_0||^2 \tag{3.28}$$

Light estimation Since estimating light with the proposed ratio model is an ill-posed problem, we decided to use Eq. 3.12 to calculate the SH parameters for each channel when the albedo and the surface normal are freezed. The minimization problem from Eq. 3.21 can then be written as:

$$\mathbf{s}^{(t)} = \arg\min_{\mathbf{s}=(\mathbf{s}_R,\mathbf{s}_G,\mathbf{s}_B)} \sum_c \|\rho_c\, \mathbf{s}_c^\top \tilde{\mathbf{n}} - I_c\|_2^2,\ c \in \{R, G, B\} \tag{3.29}$$

Alg. 2 outlines the process of RGB ratio model and Fig. 3.6 illustrates the effectiveness of this approach.

3.3.2 Limitations

Our method can estimate the albedo and the depth better than our RGBD-Fusion like method in some cases because the non-linearity optimization problem for the RGBD-Fusion like method has been solved. Still, there exist some defects:

- Three LED lights have to be set up far away from each other. As already mentioned about Eq. 3.22, the albedo refinement may fail if the lights are set too close. This can lead to some inconvenience, such as the requirement of relatively larger space to set up the system than the RGBD-Fusion like method.

- RGB three lights are likely to bring extra specularity. If we want to refine the depth of a specular object, the specular reflection will be from not only the natural scene illumination, but also RGB lights from 3 directions, which will make the refined results even

Algorithm 2 RGB Ratio Model method

Input: Initial depth image z_0, RGB image I, mask, focal length, principle point

1: $\mathbf{s}^{(0)} = \arg\min_{\mathbf{s}} E(\mathcal{P} = 1, z_0)$ {Eq. 3.29}
2: Estimate initial color albedo and build $\mathcal{P}^{(0)}$ {Eq. 3.13}
3: t = 1, $z^{(0)} = z_0$
4: **while** $\frac{\|E(\mathcal{P}^{(t)}, z^{(t)}, \mathbf{s}^{(t)}) - E(\mathcal{P}^{(t-1)}, z^{(t-1)}, \mathbf{s}^{(t-1)})\|}{E(\mathcal{P}^{(t-1)}, z^{(t-1)}, \mathbf{s}^{(t-1)})} > \epsilon$ **do**
5: $\mathcal{P}^{(t)} = \arg\min_{\mathcal{P}} E(\mathcal{P}^{(t-1)}, z^{(t-1)}, \mathbf{s}^{(t-1)})$ {Eq. 3.24}
6: $z^{(t)} = \arg\min_{z} E(\mathcal{P}^{(t)}, \mathbf{s}^{(t-1)})$ {Eq. 3.28}
7: $\mathbf{s}^{(t)} = \arg\min_{\mathbf{s}} E(\mathcal{P}^{(t)}, z^{(t)})$ {Eq. 3.29}
8: $t := t + 1$
9: **end while**

Output: Refined depth image $z^{(t)}$ and stacked color albedo $\mathcal{P}^{(t)}$

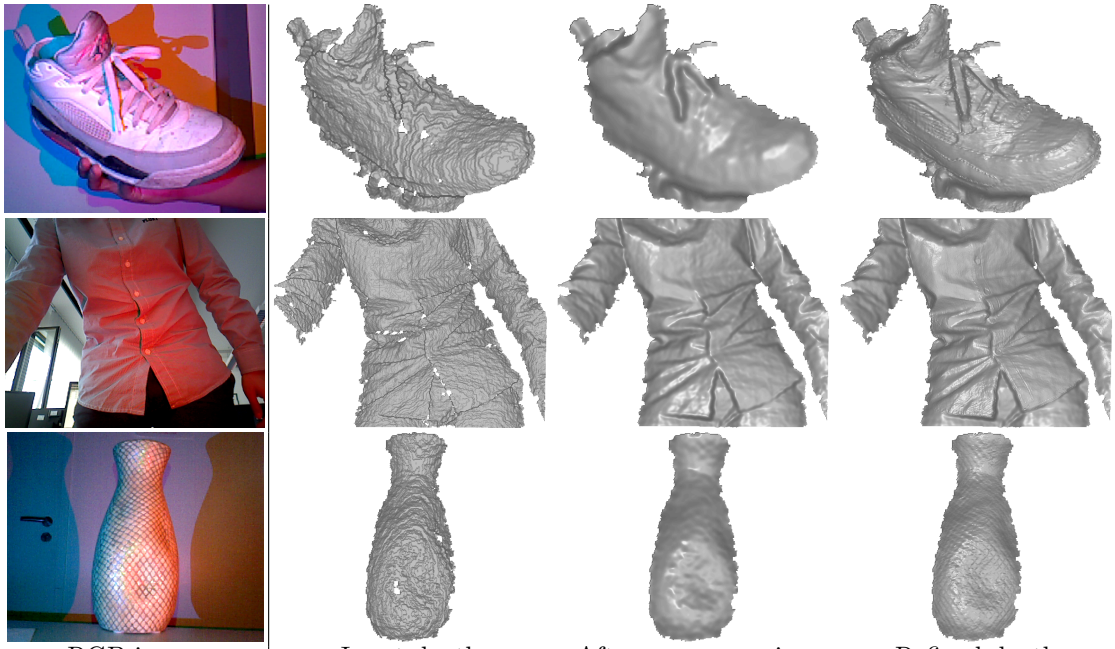


Figure 3.6: Illustrations for the depth refinement of our proposed RGB ratio model. It should be mentioned that the middle row was under the natural scene illumination and our method still works well.

worse.

- Auto white balance (AWB) has a big impact on the refined results. This is due to the fact that the success of our model highly relies on the difference among 3 channels in a color image. And AWB will mix up the information among the channels so it is very necessary to turn it off. This impedes the generalization of our model because AWB has been set as a default in many modern inexpensive cameras.

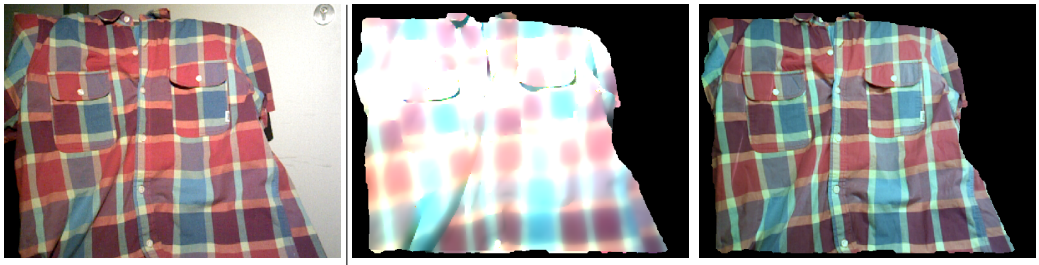
3.4 Proposed method II: Robust Multi-Light Model

3.4.1 Inspiration

We can notice that the albedo estimation of both our RGBD-Like method and RGB ratio model is highly dependent on the regularization terms which emphasize the piece-wise smoothness. This is a standard approach for almost all the state-of-the-art depth refinement method to estimate the albedo. They work fine when the albedo itself is very simple with big patches of patterns and only several dominant colors. However, there are more real-world objects containing complicated layout colors and patterns which all these methods with such a process of albedo estimation will fail. Were the albedo estimation not working, the outcome of final depth refinement has no chance to be correct. What is more, The parameters for the regularization terms are often needed to be different for various objects, so the parameters tuning for the regularization is tedious and time-consuming.

As a consequence, it is reasonable to propose a new method which is able to eliminate all the regularization terms and estimate complicated albedo. The necessity of using regularizations for calculating albedo have been mentioned in section 3.2.2, which in short is about avoiding the overfitting problem if only the SFS term is applied. Provided we have several color images for a still object with light coming from various directions, the shading term in Eq. 3.13 without the need of regularizations is sufficient for estimating the albedo. (not sure if it is true) Assuming the n lights directions are estimated while the rough surface normal and n color images are given, in this case, computing the albedo with a least square can resolve the overfitting problem. Fig. 3.7 has shown the robustness of our proposed method for the albedo estimation.

In order to simulate the scenario that a direct light comes from different directions, we simply sway a white LED light in different directions and take several images (even just the flashlight on any phone is okay). An example of a vase from different lighting directions is shown in Fig. 3.8.



RGB image | RGBD-Fusion [5] | Proposed multi-light method

Figure 3.7: Comparison for the albedo estimation between our proposed robust multi-light method and RGBD-Fusion method. RGBD-Fusion was under our implementation since their source code did not provide the albedo estimation.

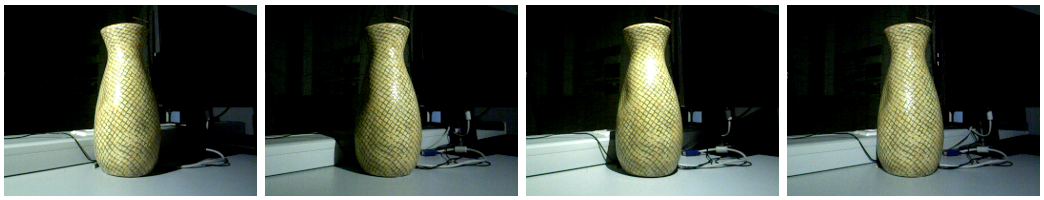


Figure 3.8: Illustrations for the obtained color images of a vase from various light directions with a white LED light.

3.4.2 Algorithm details

Since we didn't control the lighting with RGB LED lights but one white light, the ratio model in Eq. 3.19 is not applicable here. We decided to use the standard 1st-order SH model described in Eq. 3.6 again to construct the input color images.

Similar to those two methods mentioned before, the proposed algorithm consists of three parts: light estimation, albedo estimation and depth enhancement. We need to iteratively update all of them but unlike the RGB ratio model method, we don't need to have an initial estimated light and albedo beforehand. Instead, the albedo can be assumed to be 1 everywhere and the illuminations can be set to frontal directions at the beginning. Then we start refining everything iteratively in the loop, as shown in Alg. 3.

First of all, the SFS model we use for the proposed method is still based on Eq. 3.6, so the corresponding photometric stereo minimization problem now becomes:

$$\sum_{i}\sum_{c}\sum_{(x,y)\in\mathcal{M}}\sum |\rho_c(x,y)\mathbf{s}_{i,c}^{\top}\tilde{\mathbf{n}}(x,y) - I_{i,c}(x,y)|^2 \tag{3.30}$$

$c \in \{R,G,B\},\ i \in \{1,\cdots,n\}$, where n stands for the total number of varying light directions.

A simplified version of this shading energy is:

$$\sum_i \sum_c \|\rho_c \cdot \tilde{\mathbf{n}}\mathbf{s}_{i,c} - I_{i,c}\|_2^2 \tag{3.31}$$

Now the overall energy for our proposed robust multi-light method is characterized as:

$$E(\rho, z, \mathbf{s}) = \sum_i \sum_c \|\rho_c \cdot \tilde{\mathbf{n}}(z)\mathbf{s}_{i,c} - I_{i,c}\|_2^2 + \lambda_z \|z - z_0\|_2^2 \tag{3.32}$$

As we can notice, the new overall energy is extremely simple with only one shading term and one depth fidelity term, without any regularization terms for the albedo or depth estimation. And we have found out that $\lambda_z = 0.01$ works well for all cases, which means our system can be used by anybody easily without problems.

Light estimation In each iteration, we first freeze the albedo and the surface normal and then refine the SH parameters for all input images from the overall energy in Eq. 3.32. To estimate the light with the simple least squares, we need to reshape the SFS term to a linear problem with the SH light as the argument.

First of all, to further simplify the energy, we define $\mathbb{I}_c$ and $\mathbf{S}_c$ as:

$$\mathbb{I}_c = \begin{pmatrix} I_{1,c} \\ \vdots \\ I_{n,c} \end{pmatrix} \quad \mathbf{S}_c = \begin{pmatrix} \mathbf{s}_{1,c} \\ \vdots \\ \mathbf{s}_{n,c} \end{pmatrix} \tag{3.33}$$

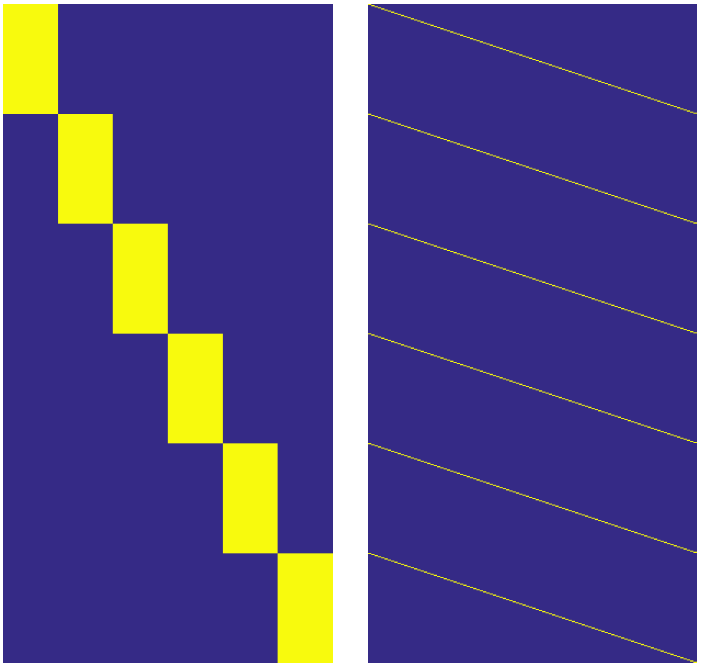
$\mathbb{I}_c \in \mathbb{R}^{mn}, \mathbf{S}_c \in \mathbb{R}^{4n}$. And then we define a multiplication operator $\odot$ between any matrix $\mathbf{A}$ and vector $\mathbf{b}$ with the same number of rows, $\mathbf{C} = \mathbf{A} \odot \mathbf{b}$. $\mathbf{C}$ is the result of the element-wise multiplication between each column of $\mathbf{A}$ and $\mathbf{b}$. Now we have a vector $\rho_c \in \mathbb{R}^m$ and a matrix $\tilde{\mathbf{n}} \in \mathbb{R}^{m\times 4}$, where m represents the number of pixel inside the mask $\mathcal{M}$. So we repeat the resulted matrix $\tilde{\mathbf{n}} \odot \rho_c$ on the diagonal of a big sparse matrix $\mathbf{A}_{\mathbb{S}_c} \in \mathbb{R}^{mn\times 4n}$, whose structure is illustrated in Fig. 4.1(a). The Eq. 3.32 now is reformulated as:

$$\min_{\mathbf{S}_c} \sum_c \|\mathbf{A}_{\mathbf{s}_c}\mathbf{S}_c - \mathbb{I}_c\|_2^2 \tag{3.34}$$

Albedo estimation Similar to the light estimation, we need to reshape the SFS term in the overall energy in order to solve it with least squares. The energy for the albedo estimation from the overall energy looks like this:

$$\min_{\rho_c} \sum_c \|\mathbf{A}_{\rho_c}\rho_c - \mathbb{I}_c\|_2^2 \tag{3.35}$$

$\mathbf{A}_{\rho_c} \in \mathbb{R}^{mn \times m}$ is the stack of n diagonal matrices $\text{diag}(\tilde{\mathbf{n}} \cdot \mathbf{s}_{i,c})$, where $\text{diag} : \mathbb{R}^m \to \mathbb{R}^{m \times m}, i \in \{1, \cdots, n\}$. A toy example of the structure of $\mathbf{A}_{\rho_c}$ with $n = 6$ is illustrated in Fig. 4.1(b).



(a) An example of $\mathbf{A}_{\mathbf{s}_c}$ with $n = 6$. The yellow part represents the repetition of matrix $\tilde{\mathbf{n}} \odot \rho_c$

(b) An example of $\mathbf{A}_{\rho_c}$ with $n = 6$. Every yellow line represents $\text{diag}(\tilde{\mathbf{n}} \cdot \mathbf{s}_{i,c})$

Figure 3.9: Illustrations for the structures of the matrices $\mathbf{A}_{\mathbf{s}_c}$ and $\mathbf{A}_{\rho_c}$. The number of different light conditions is $n = 6$.

Depth enhancement After having the estimated light and the color albedo, we can continue refining the depth. Again, we need to rearrange the energy function with the depth z as the argument. First, let us start with the simplest case and consider one pixel in one of the input images. If we expand Eq. 3.6 with the the perspective projection normal in Eq. 3.20, we have:

$$I(x,y) = \rho(x,y) \cdot \begin{pmatrix} l^1 & l^2 & l^3 \end{pmatrix} \begin{pmatrix} fz_x(x,y) \\ fz_y(x,y) \\ -1 - (x - x_0)z_x(x,y) - (y - y_0)z_y(x,y) \end{pmatrix} / d(x,y) + \rho(x,y) \cdot \varphi \tag{3.36}$$

where $d = \sqrt{(fz_x(x,y))^2 + (fz_y(x,y))^2 + (-1 - (x - x_0)z_x(x,y) - (y - y_0)z_y(x,y))^2}$ is a nor-

malizer. After rearranging, we have:

$$\frac{l^1 f - l^3(x - x_0)}{d(x,y)} z_x(x,y) + \frac{l^2 f - l^3(y - y_0)}{d(x,y)} z_y(x,y) = I(x,y) + \frac{l^3}{d(x,y)} - \rho(x,y)\varphi \tag{3.37}$$

we extend this equation to all the pixels in the mask $\mathcal{M}$ and acquire:

$$\frac{l^1 f - l^3 \tilde{x}}{d} \cdot z_x + \frac{l^2 f - l^3 \tilde{y}}{d} \cdot z_y = I + \frac{l^3}{d} - \varphi \cdot \rho \tag{3.38}$$

Provided we have the gradient matrices in x and y directions denoted roughly as:

$$D_x = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}_{m \times m} \quad D_y = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 1 & -1 & \cdots & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \\ 0 & 0 & \cdots & 1 \end{pmatrix}_{m \times m} \tag{3.39}$$

Then Eq. 3.38 becomes:

$$\left(\mathrm{diag}(\frac{l^1 f - l^3 \tilde{x}}{d}) D_x + \mathrm{diag}(\frac{l^2 f - l^3 \tilde{y}}{d}) D_y \right) z = I + \frac{l^3}{d} - \varphi \cdot \rho \tag{3.40}$$

This is the linear equation for one image. Now we define the $\mathbf{A}_z$ and $\mathbf{b}_z$ for our system:

$$\begin{aligned} \mathbf{A}_{z_c} &= \begin{pmatrix} \mathrm{diag}(\frac{l^1_{1,c} f - l^3_{1,c} \tilde{x}}{d_1}) D_x + \mathrm{diag}(\frac{l^2_{1,c} f - l^3_{1,c} \tilde{y}}{d_1}) D_y \\ \vdots \\ \mathrm{diag}(\frac{l^1_{n,c} f - l^3_{n,c} \tilde{x}}{d_n}) D_x + \mathrm{diag}(\frac{l^2_{n,c} f - l^3_{n,c} \tilde{y}}{d_n}) D_y \end{pmatrix}_{mn \times m} \\ \mathbf{b}_{z_c} &= \begin{pmatrix} I_{1,c} + \frac{l^3_{1,c}}{d_1} - \varphi_{1,c} \cdot \rho_c \\ \vdots \\ I_{n,c} + \frac{l^3_{n,c}}{d_n} - \varphi_{n,c} \cdot \rho_c \end{pmatrix}_{mn \times 1} \end{aligned} \tag{3.41}$$

It worths mentioning that in each iteration, we freeze d with z from last iteration to solve the non-linearity. Finally, we have stack $\mathbf{A}_{z_c}$ and $\mathbf{b}_{z_c}$ for each channel $c \in \{R, G, B\}$:

$$\mathbf{A}_z = \begin{pmatrix} \mathbf{A}_{z_R} \\ \mathbf{A}_{z_G} \\ \mathbf{A}_{z_B} \end{pmatrix}, \quad \mathbf{b}_z = \begin{pmatrix} \mathbf{b}_{z_R} \\ \mathbf{b}_{z_G} \\ \mathbf{b}_{z_B} \end{pmatrix} \tag{3.42}$$

After all the derivations, we finally model our energy for the depth enhancement as:

$$\min_{z} \|\mathbf{A}_z z - \mathbf{b}_z\|_2^2 + \lambda_z \|z - z_0\|_2^2 \tag{3.43}$$

The conjugate gradient (CG) method has been applied to optimize all three sub energy. The structure of the proposed algorithm is described in Alg. 3 and one refined depth example is shown in Fig. 3.10. More examples can be found in chapter 4.

Algorithm 3 Robust Multi-Light Model Method

Input: Initial depth image z_0, RGB image I, mask, focal length, principle point

1: t = 1, $z^{(t-1)} = z_0$, $\rho_R^{(0)}, \rho_G^{(0)}, \rho_B^{(0)} = 1$

2: **while** $\frac{\|E(\rho^{(t)}, z^{(t)}, \mathbf{s}^{(t)}) - E(\rho^{(t-1)}, z^{(t-1)}, \mathbf{s}^{(t)})\|}{E(\rho^{(t-1)}, z^{(t-1)}), \mathbf{s}^{(t-1)}} > \epsilon$ **do**

3: $\mathbf{s}^{(t)} = \arg\min_{\rho} E(\rho^{(t-1)}, z^{(t-1)})$ {Eq. 3.34}

4: $\rho^{(t)} = \arg\min_{\rho} E(z^{(t-1)}, \mathbf{s}^{(t)})$ {Eq. 3.35}

5: $z^{(t)} = \arg\min_{z} E(\rho^{(t)}, z^{(t-1)}, \mathbf{s}^{(t)})$ {Eq. 3.45}

6: $t := t + 1$

7: **end while**

Output: Refined depth image $z^{(t)}$ and albedo $\rho^{(t)}$

3.4.3 When super-resolution meets depth refinement

We can notice that there exist over-smoothed problems in many 3D scanning applications, which is due to the fact that the depth map given by the consumer RGB-D camera usually has a really low resolution. The scanning or reconstruction results will be then improved if the depth resolution is increased. For most of the well-known consumer RGB-D cameras, the depth resolution is far smaller than the RGB resolution. For instance, ASUS Xtion Pro Live can acquire 1280×1024 RGB images and 640×480 depth images. Microsoft Kinect 2.0 owns 1920×1080 RGB resolution but only 512×424 depth one, and Intel RealSense R200 has a 1920×1080 RGB camera while the depth reoslution is 640×480. It would be very useful if we can not only refine the depth map in its original scale but close to the RGB resolution.

In this section, we will present our approach to the combination of the photometric stereo and super-resolution, with the help of which we will provide satisfying high-quality and high-resolution depth maps.

The scale factor between the RGB and depth image is around 2 for ASUS XtionPro Live, so we can at least enlarge our map two times larger than the original size, which means the

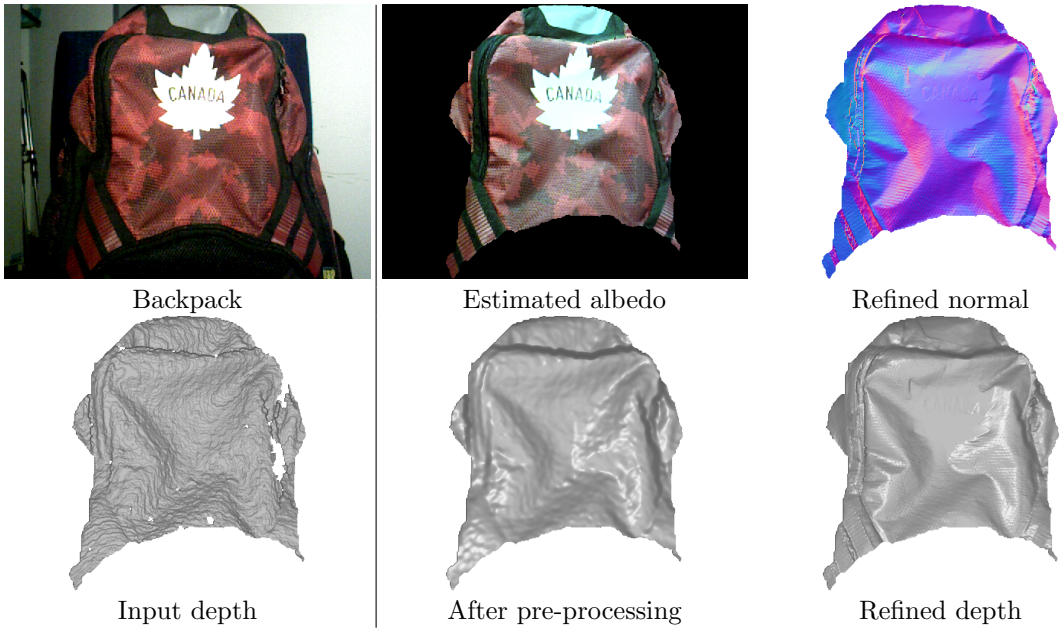


Figure 3.10: Illustrations for our proposed robust multi-light method. Here $n = 10$ images with various lighting conditions have been used, one of which is the top left RGB image.

refined depth resolution will be 1280×960. And we assume that the input depth image has been registered well (done easily with OpenNI) such that the upsampled depth Z is aligned with the large RGB image after simple interpolation.

Provided the acquired small depth and the super-resolution depth are denoted as z and Z respectively, a standard single depth image super-resolution problem can be represented like in [44]:

$$z = KFZ \tag{3.44}$$

where K represents a downsampling operator and F a blurring filter. For the sake of simplicity, we don't consider the blurring filter F but only the simple downsampling operator K. If z and Z are vectorized, the K turns to a downsampling matrix which a native 4-connected isotropic downsampling matrix is chosen.

In the light and albedo estimation part for the depth super-resolution, the energy in Eq. 3.34 and Eq. 3.35 are directly used. When we build $\mathbf{A}_{\mathbf{s}_c}$ and $\mathbf{A}_{\rho_c}$, the surface normal $\mathbf{n}(z)$ is replaced with $\mathbf{N}(Z)$ which is the normal of the large depth. However, the depth enhancement part with Eq. 3.45 should be adapted to the super-resolution framework. As we know, calculating Z from the super-resolution equation in Eq. 3.44 is ill-posed so our shading term can be treated as the regularization term. Now, Z is again used to replace z during the construction of $\mathbf{A}_z$ and $\mathbf{b}_z$,

which are now denoted by $\mathbf{A}_Z$ and $\mathbf{b}_Z$.

With the input small depth $z_0 \in \mathbb{R}^m$ and a downsampling kernel $K \in \mathbb{R}^{m \times M}$ where M and m represent the number of the pixel within the big mask and the small mask respectively, the super-resolution depth refinement energy now is changed to:

$$\min_Z \|\mathbf{A}_Z Z - \mathbf{b}_Z\|_2^2 + \lambda_z \|K \cdot Z - z_0\|_2^2 \tag{3.45}$$

After optimizing this energy, we will acquire the super-resolution version refined depth, which has been illustrated in Fig. 3.11.

(a) One input image (960 × 1280) (b) Estimated albedo (960 × 1280) (c) Estimated normal(960 × 1280)

(d) Input depth image (480 × 640) (e) Smoothed depth image (480 × 640) (f) Refined super-resolution depth (960 × 1280)

Figure 3.11: Results of the super-resolution depth of a paper bag. Input depth size is 480×640, and the refined depth's is 960×1280.

Chapter 4

Results and Evaluation

Parameter setup First we need to specify the parameters we used throughout the whole evaluation part. The default parameters are applied for the RGBD-Fusion method, which has 8 in total. It should be mentioned that, since both proposed methods don't have smoothness term for the depth enhancement, the λ_z^2 in RGBD-Fusion and λ_l in our implementation RGBD-Fusion Like method are set to 0 for the sake of fairness comparison. Only during the quantitative evalution, we want to illustrate the importance of this smoothness term so the λ_z^2 and λ_l are set to the values in table 4.1.

Table 4.1: Parameters of all the methods throughout all the experiments.

Method	Total number	Parameters
RGBD-Fusion [5]	8	$\lambda_\rho = 0.1, \lambda_\beta^1 = 0.1, \lambda_\beta^2 = 0.1, \tau = 0.05, \sigma_c = \sqrt{0.05}, \sigma_d = \sqrt{50}, \lambda_z^1 = 0.004, \lambda_z^2 = 0.0075$
RGBD-Fusion Like method (Eq. 3.10)	5	$\lambda_\rho = 10, \sigma_I = \sqrt{0.05}, \sigma_z = \sqrt{50}, \lambda_z = 500, \lambda_l = 2$
Proposed I: RGB Ratio model (Eq. 3.21)	4	$\lambda_\rho^1 = 10^{15}, \lambda_\rho^2 = 10^{13}, \sigma_c = 100, \lambda_z = 100$
Proposed II: Robust Multi-Light (Eq. 3.32)	1	$\lambda_z = 100$

4.1 Quantitative Evaluation

4.1.1 Data generation

In order to quantitatively validate the performance of our proposed methods and our implementation of the RGBD-Fusion, we use the well-known "The Joyful Yell" dataset with 3 point

light sources and ambient lights. Three various albedo scenarios are considered:

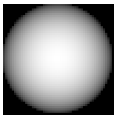
- Red, green and blue piece-wise constant areas

- Colorful patterns with a few small details inside[1]

- Colorful patterns with complicated details[2]

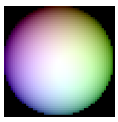
To simulate the natural scene illumination, we assume the RGB lighting as frontal directions, so the first-order SH parameters are modelled as:

$$\mathbf{s}_R = \mathbf{s}_G = \mathbf{s}_B = \begin{bmatrix} 0 & 0 & -1 & 0.2 \end{bmatrix}^\top$$



And then, in order to reproduce the LED configuration for the proposed RGB ratio model, we define the 3 lighting directions as:

$$\mathbf{s}_R = \begin{bmatrix} 0 & 0 & -1 & 0.15 \end{bmatrix}^\top$$
$$\mathbf{s}_G = \begin{bmatrix} 0.3 & 0.2 & -1 & 0.25 \end{bmatrix}^\top$$
$$\mathbf{s}_B = \begin{bmatrix} -0.2 & 0.3 & -1 & 0.2 \end{bmatrix}^\top$$

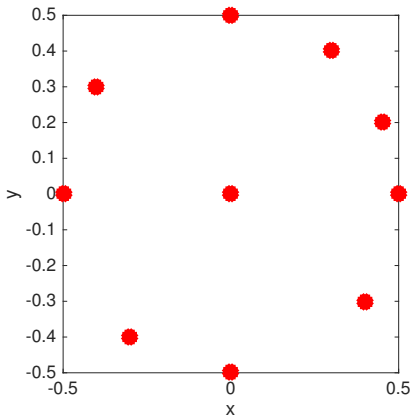


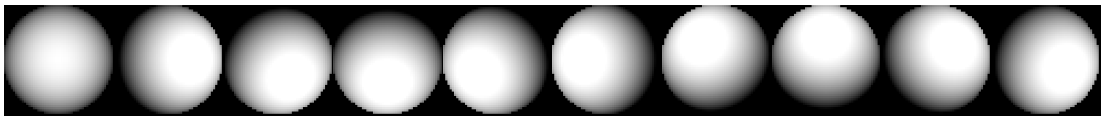
Finally, we need to produce a sequence of same images with various directional lights for our robust multi-light model. A "lighting" matrix L and the corresponding positions of 10 point

[1] EBSD map. Image Courtesy of https://mtex-toolbox.github.io/files/doc/EBSDSpatialPlots.html
[2] 1000 Visual Mashups. Image Courtsesy of https://www.flickr.com/photos/qthomasbower/3470650293

light sources can be illustrated as below, where red points represent the light positions.

$$L = \begin{pmatrix} 0.5 & 0 & -1 & 0.2 \\ 0.3 & 0.4 & -1 & 0.2 \\ 0 & 0.5 & -1 & 0.2 \\ -0.4 & 0.3 & -1 & 0.2 \\ -0.5 & 0 & -1 & 0.2 \\ -0.3 & -0.4 & -1 & 0.2 \\ 0 & -0.5 & -1 & 0.2 \\ 0.4 & -0.3 & -1 & 0.2 \\ 0 & 0 & -1 & 0.2 \\ 0.45 & 0.2 & -1 & 0.2 \end{pmatrix}^\top$$





With the 3 different albedos and all the pre-defined lights, we can create the synthetic color images like the first row in Fig. 4.2.

4.1.2 Results Accuracy

Metrics Two metrics have been defined to quantitatively evaluate the performance of depth refinement: root mean square error (RMSE) and mean angular error (MAE). Since we have already had the input rough depth and the ground truth depth as shown in Fig. 4.1, we can define two metrics as follows.

Assuming z_g, N_g and z, N are the ground truth and the refined depth and normal respectively, m the total number of pixels inside the given mask $\mathcal{M}$ and i the index inside the mask, a loosely definition is:

$$e_{RMSE} = \sqrt{\frac{\sum_i^m (z(i) - z_g(i))^2}{m}} \tag{4.1}$$

$$e_{MAE} = \frac{\sum_i^m \arccos(N(i) \cdot N_g(i))}{m} \tag{4.2}$$

The RMSE reflects refined depth quality, while the MAE illustrates if the refined object's shape is similar to the real one. It should be mentioned that e_{MAE} gives values in radians but we

convert it to degrees.

Table 4.2: Quantitative evaluations among 4 methods. RMSE and MAE are in pixels and degrees respectively. "No smooth" means no laplacian smoothness term in depth enhancement.

	Simple RGB		Pattern		Complicated Pattern	
Method	RMSE	MAE	RMSE	MAE	RMSE	MAE
Input reference	3.3305	16.3096	3.3305	16.3096	3.3305	16.3096
RGBD-Fusion [5] (no smooth)	3.3418	18.9115	3.3872	27.0026	3.3411	25.6574
RGBD-Fusion [5]	3.1751	17.2197	3.1890	18.4722	3.1708	18.0850
Fusion-Like (no smooth)	3.3475	17.5911	3.3459	23.4808	3.3898	35.2610
Fusion-Like	2.8700	17.1776	2.8749	17.7302	2.8848	19.6452
RGB ratio model	**1.9437**	**5.0574**	2.9116	17.5238	3.1006	21.2286
Robust multi-light model	3.4025	6.6640	**1.5794**	**1.7368**	**1.8424**	**2.6815**

According to table 4.1, 4.2 and Fig. 4.2, there are some interesting observations:

- Our RGBD-Fusion Like method uses less parameters than RGBD-Fusion [5] (5 against 8) but achieves almost the same results as the original paper.
- The Laplacian smoothness term in the depth enhancement energy of RGBD-Fusion method has a huge impact on the refined results. In contrast, both our proposed methods have no smoothness term but gives equal or better results.
- Single depth image refinement methods (RGBD-Fusion and RGB ratio model) have a chance to acquire satisfying results only when the albedo is elementary with several big color patches. However, they will fail and give even worse in terms of RMSE and MAE when the albedos get complex. Most of the small details on the albedo of "Pattern" and "Complicate Pattern" cannot be acquired, which leads to the wrong depth estimation. This is due to the fact that the albedo estimation in these methods highly relies on the regularization terms which prefers piecewise smooth, but this does not meet the condition of most real-world objects.
- It can be effortlessly noticed that our robust multi-light method has a strong ability to handle the cases with extremely complicated albedo. Instead of using any regularization for calculating the albedo, extra images with various light directions solve the overfitting problems of albedo and enable the albedo estimation with only the SFS term (Eq. 3.35).

4.1.3 Runtime

First of all, all the tests were performed in MATLAB R2015b under Mac OSX 10.10.5, Intel Core i5 2.7GHz, 2 cores, 8GB memory. The resolution of our synthetic image was 540×960.

(a) Input depth (b) Ground truth depth

Figure 4.1: The 3D shape of input rough depth and the ground truth depth for the quantitative evalution.

We first compare the runtime for 4 methods, which is shown in table 4.3. It is noticeable that our implementation of RGBD-Fusion like method is much faster than the original approach while the accuracy from our implementaion is quite similar to the original one. On the one hand, we only consider the pixel inside the mask while the official implementation uses the whole image. One the other hand, instead of estimating the ambient light for each pixel with an extra energy, we directly treat the ambient light as one parameter inside the first-order spherical harmonics so the the ambient light could be obtained along with lighting directions.

It should be mentioned that both of our proposed approaches are indeed about twice slower than RGBD-Fusion method. However, the RGBD-Fusion method stops when the overall energy starts increasing, which happens merely in $1-3$ iterations based on our experiment. In contrast, the minimization for our methods are both convergent so the runtime highly relies on the threshold for the relative change of the energy values.

Table 4.3: The comparison of Runtime among RGBD-Fusion method, our implementation RGBD-Fusion Like method, proposed RGB ratio model and robust multi-light method.

Method	Runtime (s)
RGBD-Fusion [5]	21.64
RGBD-Fusion Like	7.75
Proposed I: RGB Ratio	49.33
Proposed II: Multi-Light	52.82

Now for the proposed robust multi-light method, we are interested in understanding how the

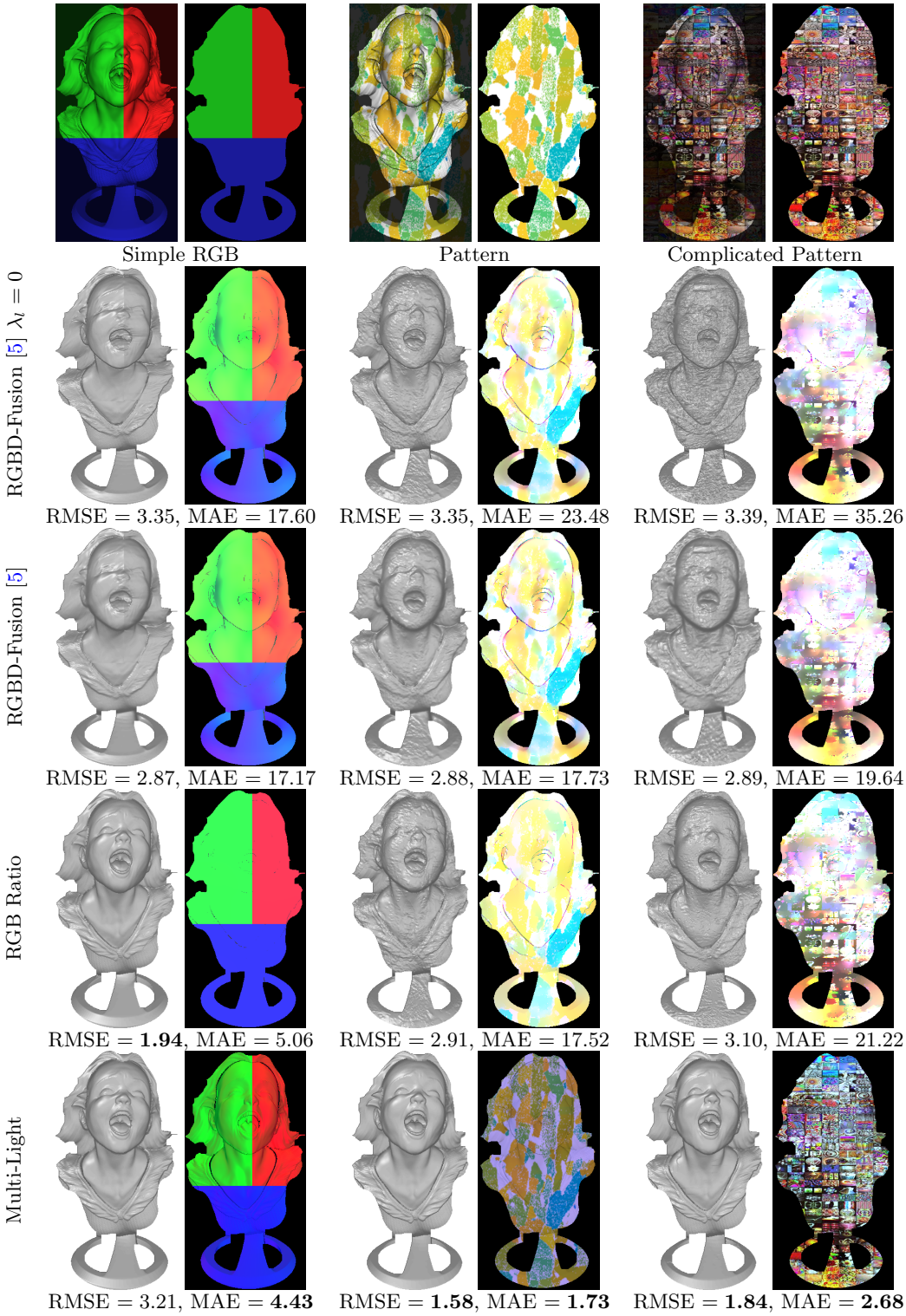


Figure 4.2: Evaluation of our two proposed methods RGB ratio and Robust Multi-Light method against our implementaion of RGBD-Fusion [5], in three different albedos from simple to complicated. Our proposed methods outperform RGBD-Fusion in all tests with respect to both RMSE and MAE. The reference errors of input are 3.35 for RMSE and 16.75 for MAE.

number of images makes the difference on the runtime as well as RMSE and MAE. To perform the experiment, we first randomly pre-defined 100 illumination directions and constructed the corresponding synthetic color images. And then we picked $4, 10, 15, 20, 30, 40, 50, 60, 80, 100$ images from this new dataset and recorded the runtime in each iteration as well as the final errors when the stopping criteria in Alg. 3 set to $\epsilon = 0.01$.

It has been shown in Fig. 4.3 that the runtime for each iteration has quadratic ascent when the number of images increases. But the impressive observation is, when the image amount increases, two errors first decreases and then go up again. Therefore, if we take the comprehensive consideration, $10 - 20$ is a suitable number for different number of lightings.

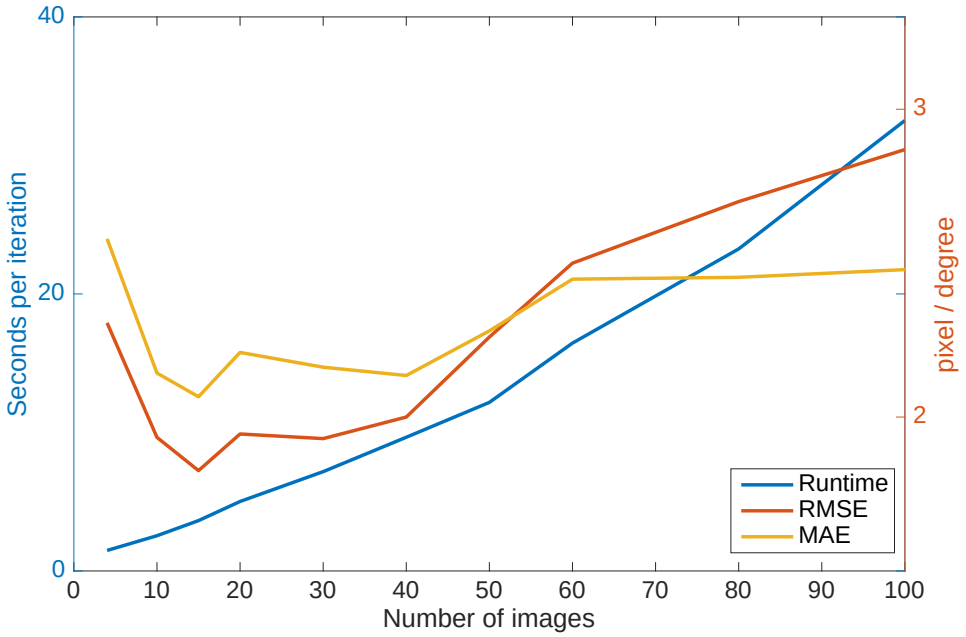


Figure 4.3: Illustrations for the runtime, RMSE and MAE in various number of images for the proposed robust multi-light method.

4.2 Real Data Evaluations

Except for the quantitative evaluation, we have found out that our robust multi-light method also exceeds the state-of-the-art methods qualitatively in many aspects. In this part, we will first show the robustness of depth estimation of our method on the objects with complex albedo. Then, we will demonstrate the performance on the non-Lambertian (specular) objects.

Moreover, as mentioned in chapter 2, uncalibrated photometric stereo suffers from the GBR ambiguity. So we will finally exemplify that our method also outperforms other photometric stereo algorithms because of the input depth cues.

4.2.1 Complicated albedo objects

4.2.2 Specular (non-Lambertian) objects

We also compare our multi-light method with the RGBD-Fusion [5] method. Same as last section, we turned off the Laplacian smoothness term in their method for the sake of fairness.

The Lambertian reflectance model is built up based on the Lambert's cosine law which the diffused object is the prerequisite. It should not work with the non-Lambertian objects theoretically. Indeed, the shapes recovered from RGBD-Fusion appear very obvious artefacts in the specular areas, as we can notice in Fig. 4.5. This is because the specular area is "light polluted" and we cannot retrieve any color or shape details from it. In contrast, although the Lambertian model is also applied in our robust multi-light method, it can still recover the real shapes with the presence of specularity.

To explain the reason that our method is working, we first assume that 10 input color images are given. We know the specularity in each image differs from all others owing to the fact that we move the active LED light location. This means, the specularity appearing in a certain image has a high probability not to be specular again in the rest 9 images. And in the albedo and depth refinement of our algorithm, we estimate the albedo and the depth with least square using all 10 images instead of one, so the specular area in that image will be treated as the Therefore,

4.2.3 Comparison with Photometric stereo method

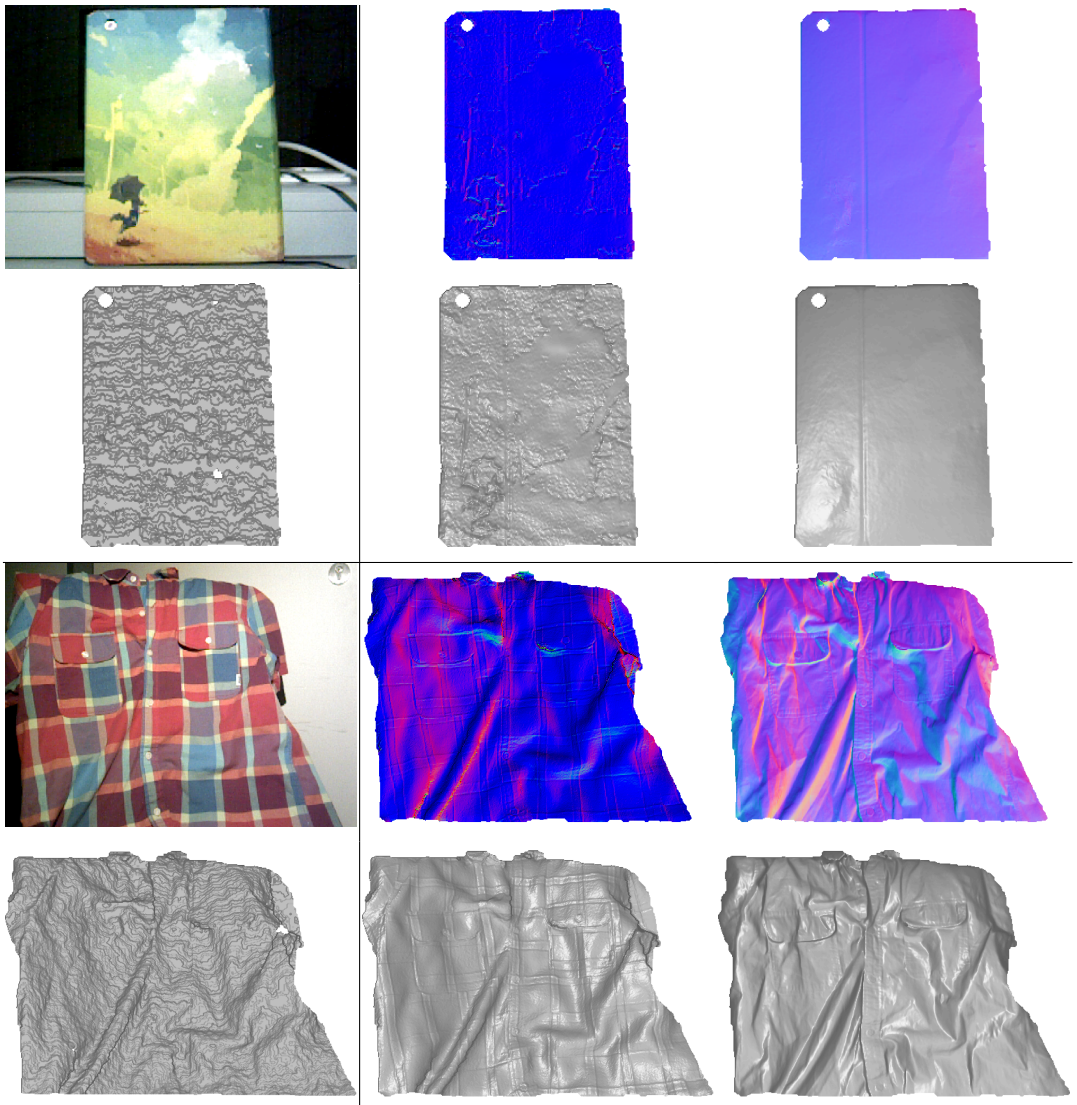


Figure 4.4: Comparison our multi-light model with RGBD-Fusion in two specular objects. On the first column, the RGB images of the folder and the vase are ones of the 10 various illuminations. First and third rows correspond to the surface normal from the refined depth, while second and fourth are the refined depth.

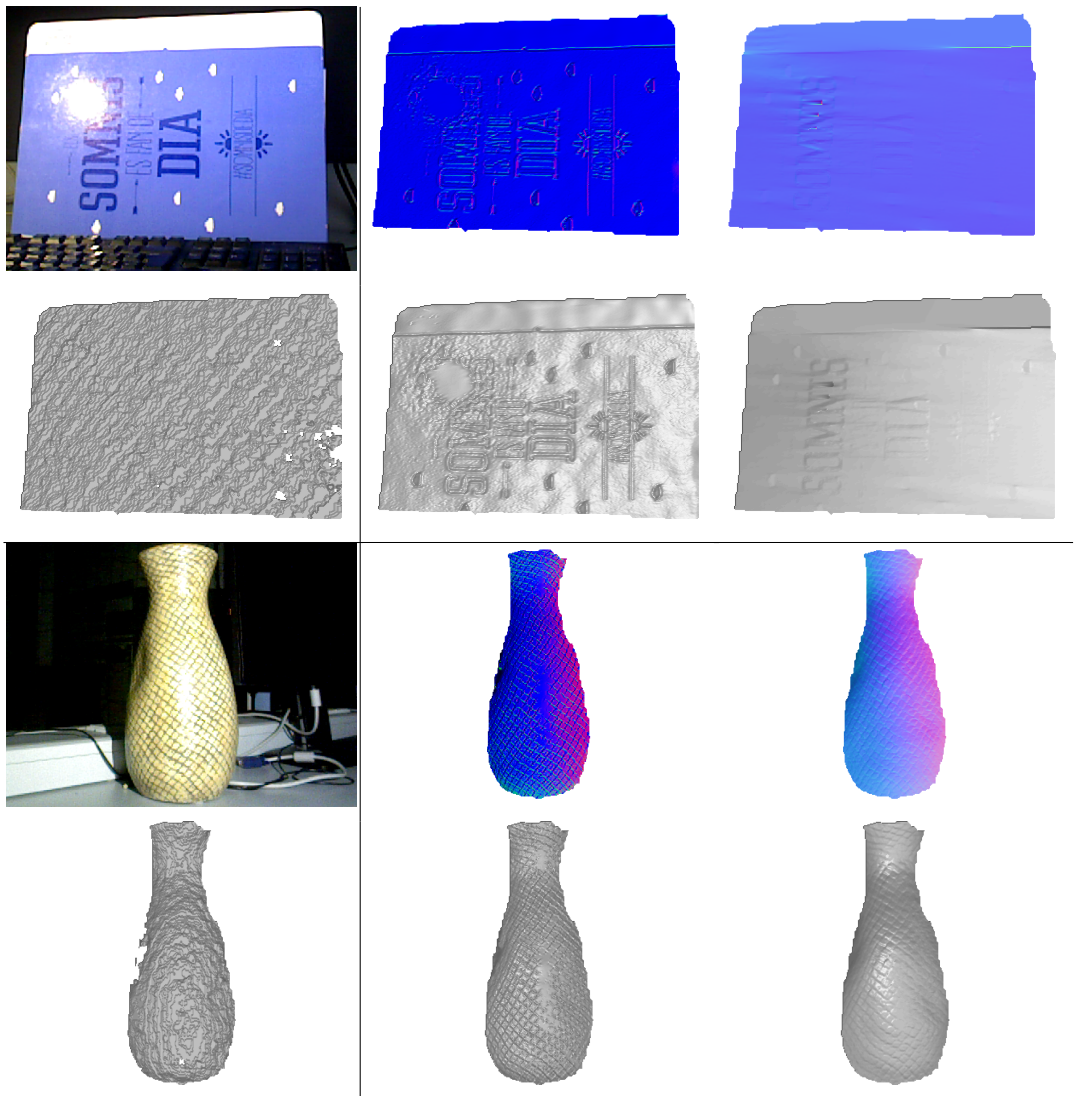


Figure 4.5: Comparison our multi-light model with RGBD-Fusion in two specular objects. On the first column, the RGB images of the folder and the vase are ones of the 10 various illuminations. First and third rows correspond to the surface normal from the refined depth, while second and fourth are the refined depth.

Chapter 5

Conclusion and Future Work

talk about that almost all the state-of-the-art method in single depth image estimation is not really theoretically correct. Their results looks good but actually not really correct because of the albedo estimation is not satisfying with all those regularizers. Recently some researchers have proposed a general framework to solve deblurring and demosaiking problems without knowing what the regularizer itself is. Instead, they separate the classic $\|Ax-b\|^2+R(x)$ using methods like Primal-Dual, ADMM or forward backward. To solve the proximal operator of the $R(x)$ in these optimization method, they just solve it with a BM3D denoiser [45] or a deep denoising neural network [46].

Therefore, it would be very interesting if we can use such a method to calculate the albedo.

Similar to the p36 book from Forsythe and Ponce, we can times a matrix to deal with the shadow problem in images.

Appendix A

Implementation details

1. Detail about how to build Laplacian efficiently inside the mask
2. the derivation of $\Psi z = 0$ in RGB ratio model part
3. List of mathmetical symbol like paper "Shading-based Refinement on Volumetric Signed Distance Functions"

Bibliography

[1] Radu Horaud. A short tutorial on three-dimensional cameras, 2013. http://perception.inrialpes.fr/~Horaud/Courses/pdf/Horaud_3Dcameras_tutorial.pdf.

[2] Edward H Adelson and Alex P Pentland. The perception of shading and reflectance. *Perception as Bayesian inference*, pages 409–423, 1996.

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

[4] Yudeog Han, Joon-Young Lee, and In So Kweon. High quality shape from a single rgb-d image under uncalibrated natural illumination. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1624, 2013.

[5] Roy Or-El, Guy Rosman, Aaron Wetzler, Ron Kimmel, and Alfred M Bruckstein. Rgbd-fusion: Real-time high precision depth recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416, 2015.

[6] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[7] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research*, pages 235–252. Springer, 2017.

[8] Nikolas Engelhard, Felix Endres, Jürgen Hess, Jürgen Sturm, and Wolfram Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, volume 180, pages 1–15, 2011.

[9] C. Kerl. Odometry from rgb-d cameras for autonomous quadrocopters. Master's thesis, Technical University Munich, Germany, Nov. 2012.

[10] ASUS. Xtion pro live. https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/. Accessed: 2017-05-15.

[11] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970.

[12] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015.

[13] Klett, Eberhard Witwe, Detleffsen, Christoph Peter, et al. *IH Lambert... Photometria sive de mensura et gradibus luminis, colorum et umbrae*. sumptibus viduae Eberhardi Klett, 1760.

[14] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003.

[15] Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001.

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] Berthold KP Horn and Michael J Brooks. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33(2):174–208, 1986.

[18] Robert T. Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):439–451, 1988.

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

[20] David A Forsyth and Jean Ponce. A modern approach. *Computer vision: a modern approach*, pages 88–101, 2003.

[21] Carlos Hernández, George Vogiatzis, and Roberto Cipolla. Overcoming shadows in 3-source photometric stereo. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):419–426, 2011.

[22] Hideki Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA A*, 11(11):3079–3089, 1994.

[23] Alan Yuille and Daniel Snow. Shape and albedo from multiple images using integrability. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 158–164. IEEE, 1997.

[24] Neil G Alldrin, Satya P Mallick, and David J Kriegman. Resolving the generalized bas-relief ambiguity by entropy minimization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[25] Thoma Papadhimitri and Paolo Favaro. A closed-form, consistent and robust solution to uncalibrated photometric stereo via local diffuse reflectance maxima. *International journal of computer vision*, 107(2):139–154, 2014.

[26] Thoma Papadhimitri and Paolo Favaro. A new perspective on uncalibrated photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1474–1481, 2013.

[27] Yvain Quéau, François Lauze, and Jean-Denis Durou. Solving uncalibrated photometric stereo using total variation. *Journal of Mathematical Imaging and Vision*, 52(1):87–107, 2015.

[28] Jonathan T Barron and Jitendra Malik. High-frequency shape and albedo from shading using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2521–2528. IEEE, 2011.

[29] Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, and Stephen Lin. Shading-based shape refinement of rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1422, 2013.

[30] Chenglei Wu, Michael Zollhöfer, Matthias Nießner, Marc Stamminger, Shahram Izadi, and Christian Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33(6):200, 2014.

[31] Chenglei Wu, Kiran Varanasi, Yebin Liu, Hans-Peter Seidel, and Christian Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1108–1115. IEEE, 2011.

[32] Kichang Kim, Akihiko Torii, and Masatoshi Okutomi. Joint estimation of depth, reflectance and illumination for depth refinement. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–9, 2015.

[33] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013.

[34] Roy Or-El, Rom Hershkovitz, Aaron Wetzler, Guy Rosman, Alfred M Bruckstein, and Ron Kimmel. Real-time depth refinement for specular objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4378–4386, 2016.

[35] Mohammadul Haque, Avishek Chatterjee, Venu Madhav Govindu, et al. High quality photometric reconstruction using a depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2275–2282, 2014.

[36] Jaesik Park, Sudipta N Sinha, Yasuyuki Matsushita, Yu-Wing Tai, and In So Kweon. Multiview photometric stereo using planar mesh parameterization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1161–1168, 2013.

[37] Yvain Quéau, Jean Mélou, Jean-Denis Durou, and Daniel Cremers. Dense multi-view 3d-reconstruction without dense correspondences. *arXiv preprint arXiv:1704.00337*, 2017.

[38] Avishek Chatterjee and Venu Madhav Govindu. Photometric refinement of depth maps for multi-albedo objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 933–941, 2015.

[39] Chenglei Wu, Bennett Wilburn, Yasuyuki Matsushita, and Christian Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 969–976. IEEE, 2011.

[40] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[41] Qing Zhang, Mao Ye, Ruigang Yang, Yasuyuki Matsushita, Bennett Wilburn, and Huimin Yu. Edge-preserving photometric stereo via depth fusion. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2472–2479. IEEE, 2012.

[42] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[43] Wallace Casaca, Luis Gustavo Nonato, and Gabriel Taubin. Laplacian coordinates for seeded image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–391, 2014.

[44] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[45] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pajak, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, et al. FlexISP: A flexible camera image processing framework. *ACM Transactions on Graphics (TOG)*, 33(6):231, 2014.

[46] Tim Meinhardt, Michael Möller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. *arXiv preprint arXiv:1704.03488*, 2017.