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

1.2 Outline

Chapter 2

Background

Joint estimation of depth, reflectance and illumination for depth refinement

2.1 RGB-D Cameras

2.1.1 General

2.1.2 ASUS Xtion PRO LIVE

2.2 Shape from Shading & Photometric Stereo

there exist infinite solutions if only the SFS term is applied to refine the depth (Yvain's PhD thesis). Therefore, z-z0 data fidelity is albedo to applied to restrict to the unique solution

2.2.1 Lambertian reflectance model

We can show the Intrinsic image decomposition as the an example of Lambertian reflectance as an informal explanation. shading is the product of the a certain kind of illumination model and the shape (surface normal) [3]

Illustrate with the an image from MIT intrinsic dataset [4]

SH model is an extension of Lambertian model

https://pdfs.semanticscholar.org/7b8d/fc5d6e276f8048bb53b4a5e0611019570f1b.pdf [5]

cite Shape From Shading Emmanuel Prados, Olivier Faugeras https://en.wikipedia.org/wiki/Lambertian_reflectance

https://www.cs.cmu.edu/afs/cs/academic/class/15462-f09/www/lec/lec8.pdf http://www.cs.virginia.edu/~gfx/Courses/2011/ComputerVision/slides/lecture20_pstereo.pdf

we assume that surfaces in a scene are Lambertian, and we parameterize the incident lighting with spherical harmonics (SH) [Wu et al. 2011] [6].

In fact, we estimate incident irradiance as a function of the surface normal, that is the incident light, filtered by the cosine with the normal. For Lambertian reflectance, the incident irradiance function is known to be smooth, and can be represented with only little error using the first nine spherical harmonics basis functions up to 2nd order [7]. (well, actually should check this one [8]) As with previous approaches, we henceforth estimate lighting from a grayscale version of I, and thus assume gray lighting with equal values in each RGB channel. In some steps, full RGB images are used, which we denote Ic. Unlike offline multi-view methods, we employ a triangulated depth map as geometry parameterization. This means there is a fixed depth pixel to mesh vertex relation, and we can express the reflected irradiance B(i, j) of a depth pixel (i, j) with normal n(i, j) and albedo k(i, j)

This sentence is from [9]

2.2.2 Surface normal

http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

orthographic model perspective model

It is an ill-posed problem to estimate the normal, that's where SFS and PS are involved.

SFS: Horn

PS:

calibrated light: woodham [10] https://classes.soe.ucsc.edu/cmps290b/Fall05/readings/Woodham80c.pdf

uncalibrated light:

Hayakawa 94 [11] http://www.wisdom.weizmann.ac.il/~vision/courses/2010_2/papers/photometric_stereo.pdf start the I = albedo*light*normal 3×3 linear ambiguity

Yville 97 [12]: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.446.3648&rep=rep1&type=pdf use integrability (smoothness), reduce the ambiguity to 3-parameter ambiguity (GBR) $z(x, y) = \lambda z(x, y) + \mu x + \beta y$, which is GBR ambiguity. That's why our method works because we have initial depth z_0 and data fidelity term constrains the z to z_0, so the ambiguity equation is invalid. And in our case: PDE (Δz) -¿ integrability is implicity enforced

All the following PS method is trying to solve this ambiguity alldrin 07 [13] use entropy http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.7264&rep=rep1&type=pdf

[14] perspective http://www.cv-foundation.org/openaccess/content_cvpr_2013/papers/Papadhimitri_A_New_Perspective_2013_CVPR_paper.pdf

[15]https://pdfs.semanticscholar.org/2cf9/088e9faa81872b355a4ea0a9fae46d3c8a08.pdf

[16] use TV http://oatao.univ-toulouse.fr/15158/1/queau_15158.pdf

2.3 Depth Map Refinement

mention Super-resolution Imaging that it is also very interesting to extend our the state-of-the-art depth refinement to real refinement.

mentioned very latest research is also related to depth refinement, using several images from different views (Yvain's and Zuozuo's Arxiv paper)

Chapter 3

Methodology

Many computer vision applications such as 3D object 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 without changing to an expensive camera. Therefore, the depth refinement techniques play an essential role here.

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.* [2] which we have chosen to implement as a starting point. A proposed method based on a RGB ratio model is then followed and introduced to eliminate the nonlinearity in most of modern depth enhancement method. Finally, another proposed technique which does not require any regularization terms is presented. This method has also exhibited the ability of dealing with the objects with complicated albedos and extension to depth super-resolution.

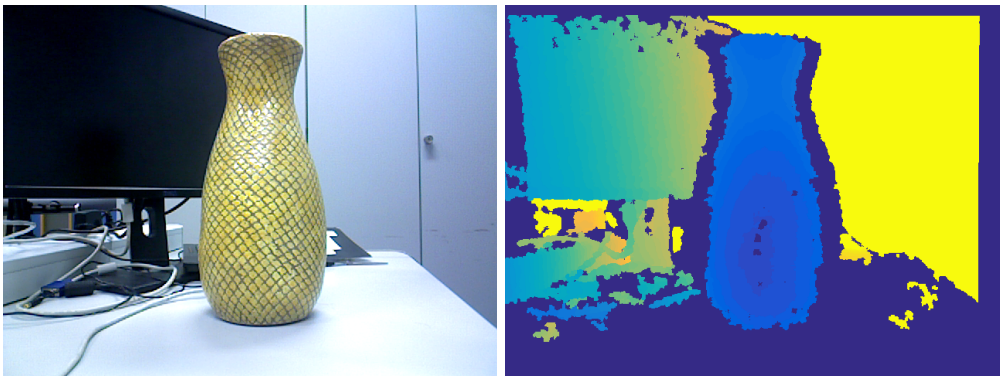
3.1 Pre-Processing

The first step for most of the image processing tasks is to pre-process the initial input image. Due to the hardware limitation of modern inexpensive RGBD 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 relative smooth surface.

In this section, we will describe respectively the basic 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, or remove unwanted texts or objects in a photography [17]. 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 noisy image 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 [17] 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 are the area with missing information.

To build the update map U in each time step, there are two principles that [17] 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 problem 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.

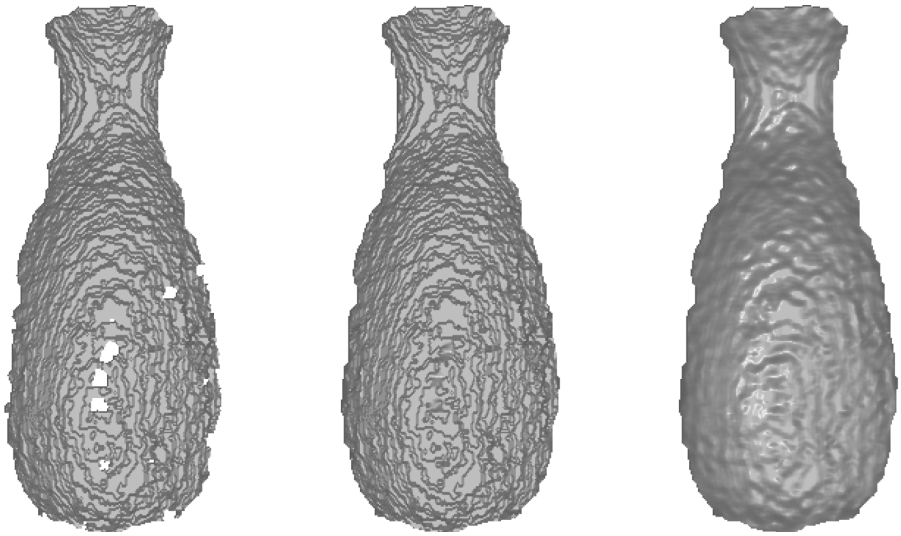
3.1.2 Depth denoising

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 [1, 2, 18–21], bilateral fitering [22] is used as our depth pre-processing smoother.

The advantages 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 $\mathbf{x}$, $\mathcal{N}$ represents the neighbouring pixels with $\mathbf{x}$ in the center, and W is the the sum of the all the weights. 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.* [2] 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 with our own idea inside.

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, and the results have 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 [5] has 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:

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

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 and ρ are vectorized to $\mathbb{R}^m$ within the mask, while $\tilde{\mathbf{n}} \in \mathbb{R}^{m\times 4}$. 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 a SFS term, an albedo anisotropic Laplacian term, a depth data fidelity term and a depth isotropic Laplacian term. Now we will go into the details step by

step.

3.2.1 Light estimation

Here the 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.

In the sense of intrinsic image decomposition, an image can be decomposed as the product of albedo and shading, so we can treat $\mathbf{s}^\top \tilde{\mathbf{n}}(x,y)$ in Eq. 3.7 as the shading.

To compute the spherical harmonics parameters, we assume the albedo ρ equals to 1 for each pixel. Since there are known intensity value 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 methods based on SFS or photometric stereo techniques assume constant or uniform albedo. Such assumption does not fit in with the real-world objects, and hence, 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, such albedo is prone to the overfitting, which make the acquired albedo contain all the undesired spatial layout details. This is due to the fact that both input image I and the surface normal $\mathbf{n}$ are noisy. To resolve the overfitting problem, we should impose some restrictions on the estimation of albedo. A large amount of our daily objects have piecewise smooth appearance, 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 to several pieces with different intensities, which can be treated as the image segmentation problem to some extend. 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 [23], an anisotropic Laplacian term is imposed to estimate the albedo. Now, the SH parameters $\mathbf{s}$ and the surface normal $\mathbf{n}$ are fixed, 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 to 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 non-linear problem, e.g. Levenberg-Marquardt algorithm or ADMM, but they are not suitable in 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 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:

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

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 in the overall energy (Eq. 3.10 was used as a core in all light, albedo and depth estimation. The whole process of RGBD-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 SH parameter, $\mathbf{s} = \arg\min_{\mathbf{s}} E(\rho = 1, z_0)$ {Eq. 3.12}
2: Estimate 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 many 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, but it 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 calibration without much effort. Hence, it is reasonable to formulate the surface normal with the perspective projection model.

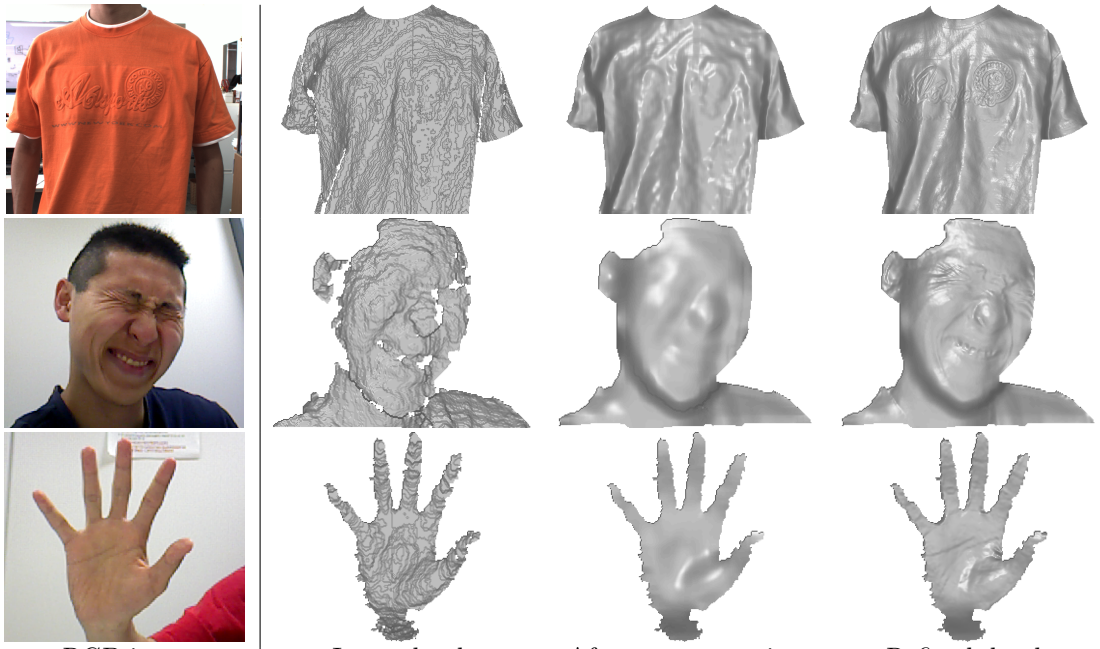


Figure 3.3: Illustrations for our implementation of RGBD-Fusion Like method. Top row is a T-shirt from [1]. Middle and bottom row are author's face and palm.

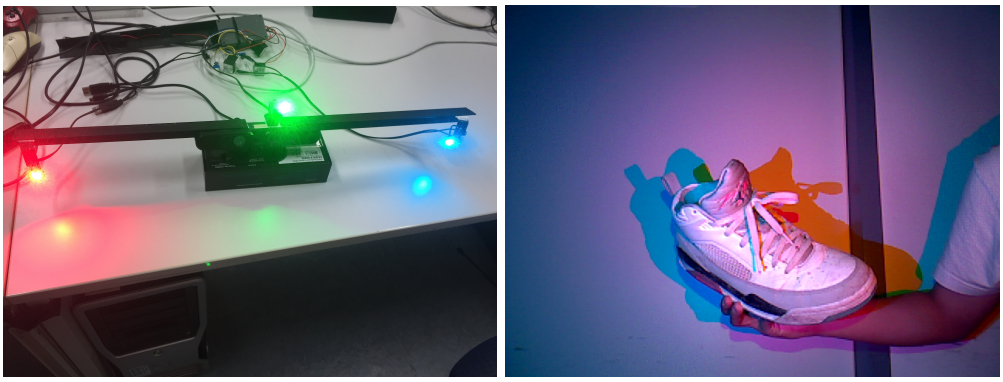
- In our RGBD-Fusion like method, only the intensity is applied because the values in RGB channels are more or less the same under the natural scene illumination. When we estimated the SH lighting parameters and the albedo in 3 channels separately, the results are quite similar to each other. So using all three channel rather than just the intensity value 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.

- The most important inspiration for us to propose RGB ratio model in the next section is, the RGBD-Fusion like method was not convergent in terms of depth enhancement part because of the fix-point method. In the 4th line of the Alg. 1, we make the iteration 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 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 an 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:

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

Similarly, we can acquire another two ratio models which are between green and blue, and blue and red channels respectively. We are able to notice 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 to 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 iteratively 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 about 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 is or close to 0. This is the reason why we need to set up 3 LED lights with a distance to each other, which will provide us enough difference on the light directions.

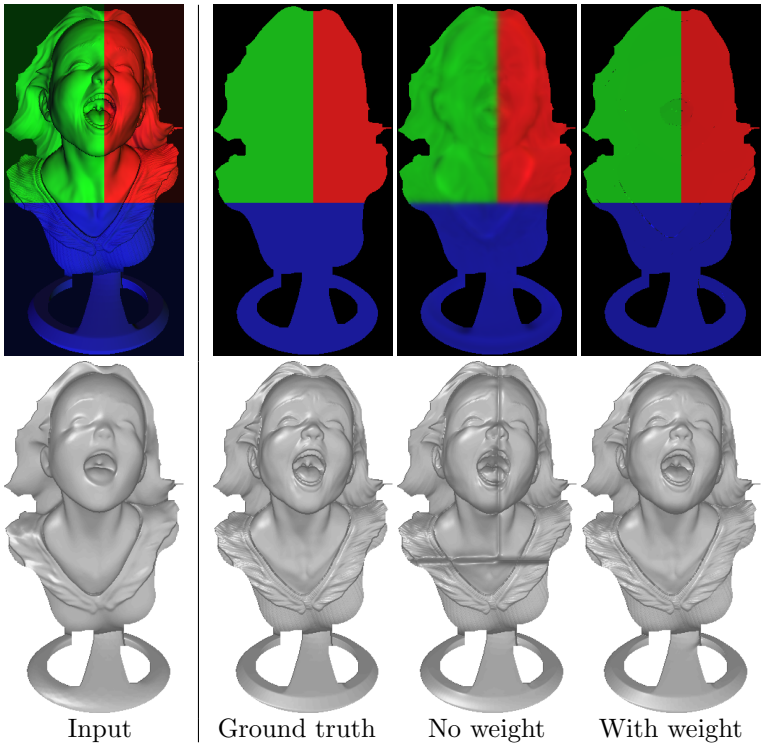


Figure 3.5: Illustrations for the importance of the weight ω inside regularization term in Eq. 3.24 when estimating the albedo. Top: first one is the input color image and the rest three are the albedos. Bottom: 3D shape from depth. Noted that the light parameter is given.

2. Instead of using anisotropic Laplacian regularization in RGBD-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 less efforts to build such a smoothness term than the anisotropic term, but 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

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 **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 a 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:

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

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: $\quad\mathcal{P}^{(t)} = \arg\min_{\mathcal{P}} E(\mathcal{P}^{(t-1)}, z^{(t-1)}, \mathbf{s}^{(t-1)})$ {Eq. 3.24}
6: $\quad z^{(t)} = \arg\min_{z} E(\mathcal{P}^{(t)}, \mathbf{s}^{(t-1)})$ {Eq. 3.28}
7: $\quad\mathbf{s}^{(t)} = \arg\min_{\mathbf{s}} E(\mathcal{P}^{(t)}, z^{(t)})$ {Eq. 3.29}
8: $\quad t := t + 1$
9: **end while**

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

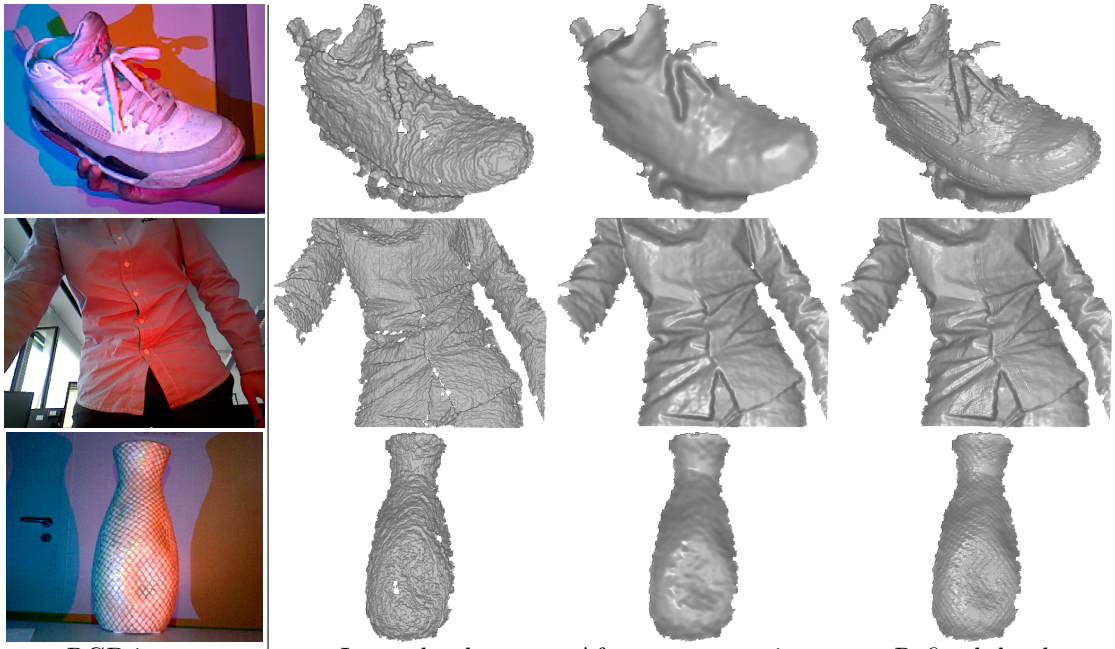


Figure 3.6: Illustrations for the depth refinement of our proposed RGB ratio model. It should be mentioned that the middle row was under the natural scene illumination and our method still works well.

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 for our new RGB ratio model.

- 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 too close and thus, we need to put 3 lights too near. This can lead to some inconvenience, such as the requirement of enough space to put the system.

- RGB three lights is likely to bring extra specularity. If we want to refine the depth of an specular objects, 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 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 in 3 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 corrrect. What is more, The parameters for the regularization terms are often needed to be different for various object, 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 shading 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) 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.

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 flash lamp on any phone is okay). An example of a vase from different lighting directions are shown in Fig. 3.7.

put a real-world albedo estimate case that RGBD-fusion fails but our method works

3.4.2 Algorithm details

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

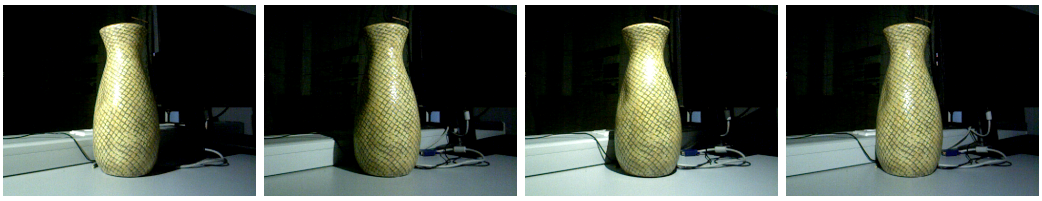


Figure 3.7: Illustrations for the obtained color images of a vase from various light directions with a white LED light.

Again, the proposed algorithm consists of three parts: light estimation, albedo estimation and depth enhancement. We need to iteratively update the light directions, color albedo and the depth, but unlike the RGB ratio model method, we don't need to have an initial estimated light and albedo beforehand. Instead, we can just assume the albedo to be 1 everywhere at the beginning and then start refining everything reiteratively in the loop, as shown in Alg. 3.

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

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

$c\in\{R,G,B\},\ i\in\{1,\cdots,n\}$, where n stands for the total number of varying light directions. A simplified version of the SFS 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 SFS term and one depth fidelity term, no any regularization terms for the albedo or depth estimation. And we have found out that $\lambda_z=0.01$ works really 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 with least squares. The energy for the albedo estimation from the overall energy looks like this:

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

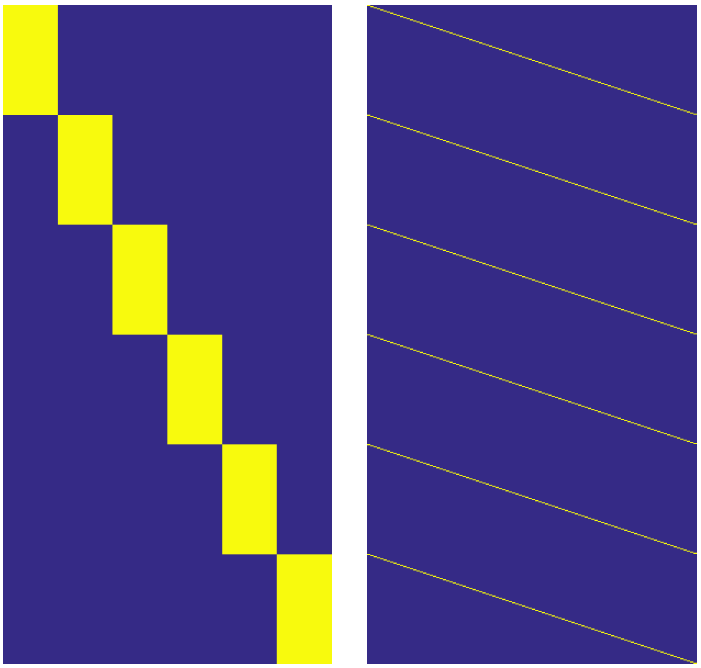
$\mathbf{A}_{\rho_c} \in \mathbb{R}^{mn\times m}$ is the stack of n diagonal matrices $\text{diag}(\tilde{\mathbf{n}} \cdot \mathbf{s}_{i,c})$, where $\text{diag} : \mathbb{R}^m \to \mathbb{R}^{m\times m}, i \in \{1, \cdots, n\}$. A toy example of $\mathbf{A}_{\rho_c}$ with $n = 6$ is illustrated in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

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 from 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 normalizer. After rearranging, we have:

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



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

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

Figure 3.8: Illustrations for the structures of the matrices $\mathbb{A}_{\mathbb{S}_c}$ and $\mathbb{A}_{\rho_c}$. The number of different light conditions is $n = 6$.

If we extend this to the whole pixel in the mask $\mathcal{M}$, the equation becomes:

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

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:

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

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

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

It worths mentioning that in each iteration, we freeze d with z from last iteration, so the non-linearity is solved. 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. And the structure of the proposed algorithm is described in Alg. 3 and one refined depth result is shown in Fig. 3.9. More examples can be found in chapter 4.

3.4.3 When super-resolution meets depth refinement

For most of the well-known comsumer 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 of the combination of the photometric stereo

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 $\rho^{(t)}$

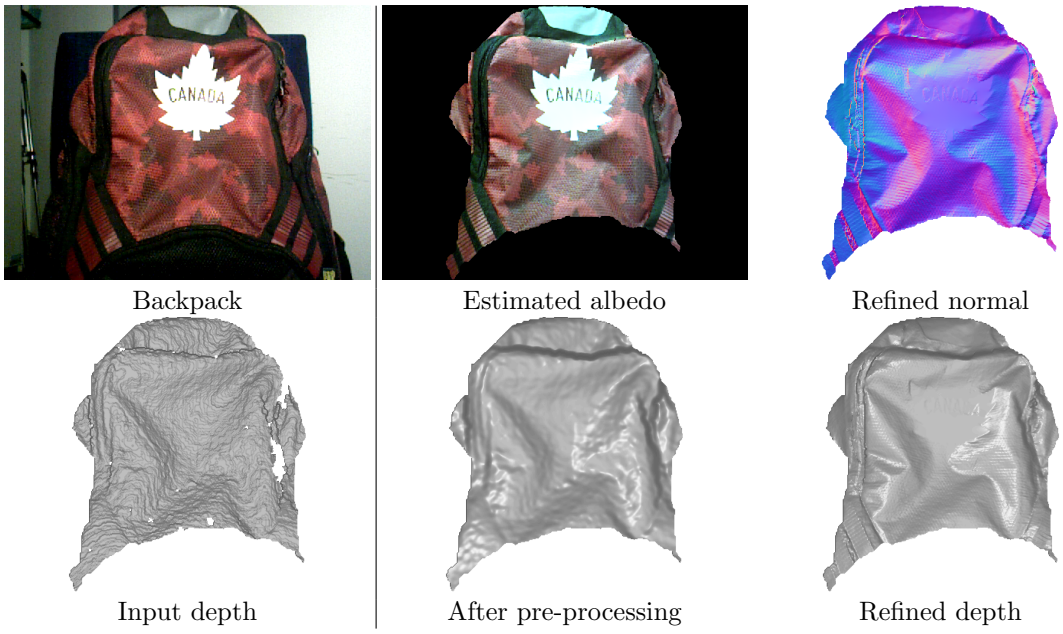


Figure 3.9: Illustrations for our proposed robust multi-light method. Here $n = 10$ images with various lighting conditions have been used, one of which is the top left RGB image.

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 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.

Assuming 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 [24]:

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

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

In the light, albedo estimation part, 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, the super-resolution equation in Eq. 3.44 is ill-posed so our SFS 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 $z_0 \in \mathcal{R}^m$ and a downsampling kernal $K \in \mathbb{R}^{m \times M}$ where M and m represent the number of 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 acquired super-resolution version refined depth, as illustrated in Fig. 3.10.

(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 3.10: 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

Give some introduction, the structure of this chapter.

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 [2]	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	5	$\lambda_\rho = 10, \sigma_I = \sqrt{0.05}, \sigma_z = \sqrt{50}, \lambda_z = 500, \lambda_l = 2$
Proposed I: RGB Ratio	4	$\lambda_\rho^1 = 10^{15}, \lambda_\rho^2 = 10^{13}, \sigma_c = 100, \lambda_z = 100$
Proposed II: Multi-Light	1	$\lambda_z = 100$

4.1 Quantitative Evaluation

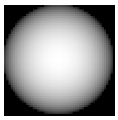
Data generation In order to quantitatively validate the performance of our proposed methods and our implementation of the RGBD-Fusion, we use the well-known "The Joyful Yell" dataset with 3 point light sources and ambient lights. Three various albedo scenarios are considered:

- Red, green and blue piece-wise constant areas

- Colorful patterns with a few small details inside[1]
- Colorful patterns with complicated details[2]

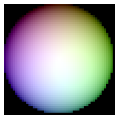
To simulate the natural scene illumination, we assume the RGB lighting as frontal directions, so the first-order SH parameters are modelled as:

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



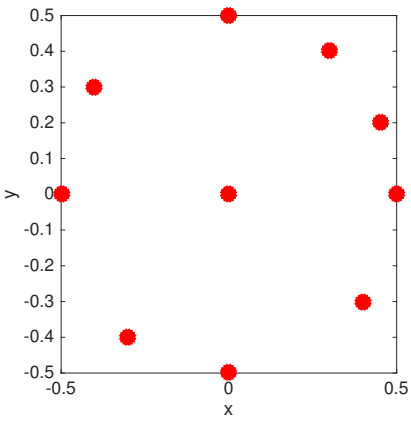
And then, in order to reproduce the LED configuration for the proposed RGB ratio model, we define the 3 lighting directions as:

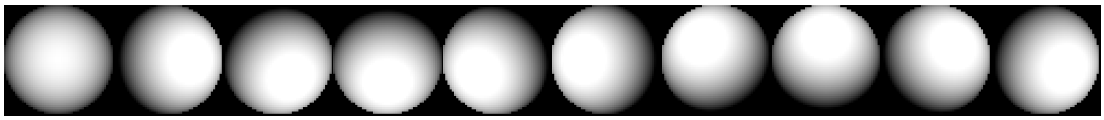
$$\mathbf{s}_R = \begin{bmatrix} 0 & 0 & -1 & 0.15 \end{bmatrix}^\top$$
$$\mathbf{s}_G = \begin{bmatrix} 0.3 & 0.2 & -1 & 0.25 \end{bmatrix}^\top$$
$$\mathbf{s}_B = \begin{bmatrix} -0.2 & 0.3 & -1 & 0.2 \end{bmatrix}^\top$$



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

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





[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

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.

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 [2] (no smooth)	3.3418	18.9115	3.3872	27.0026	3.3411	25.6574
RGBD-Fusion [2]	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 [2] (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).

(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.

4.2 Real Data Evaluations

try to add noise on some of the images in the synthetic dataset and check the effectiveness of our method for robust normal recovery

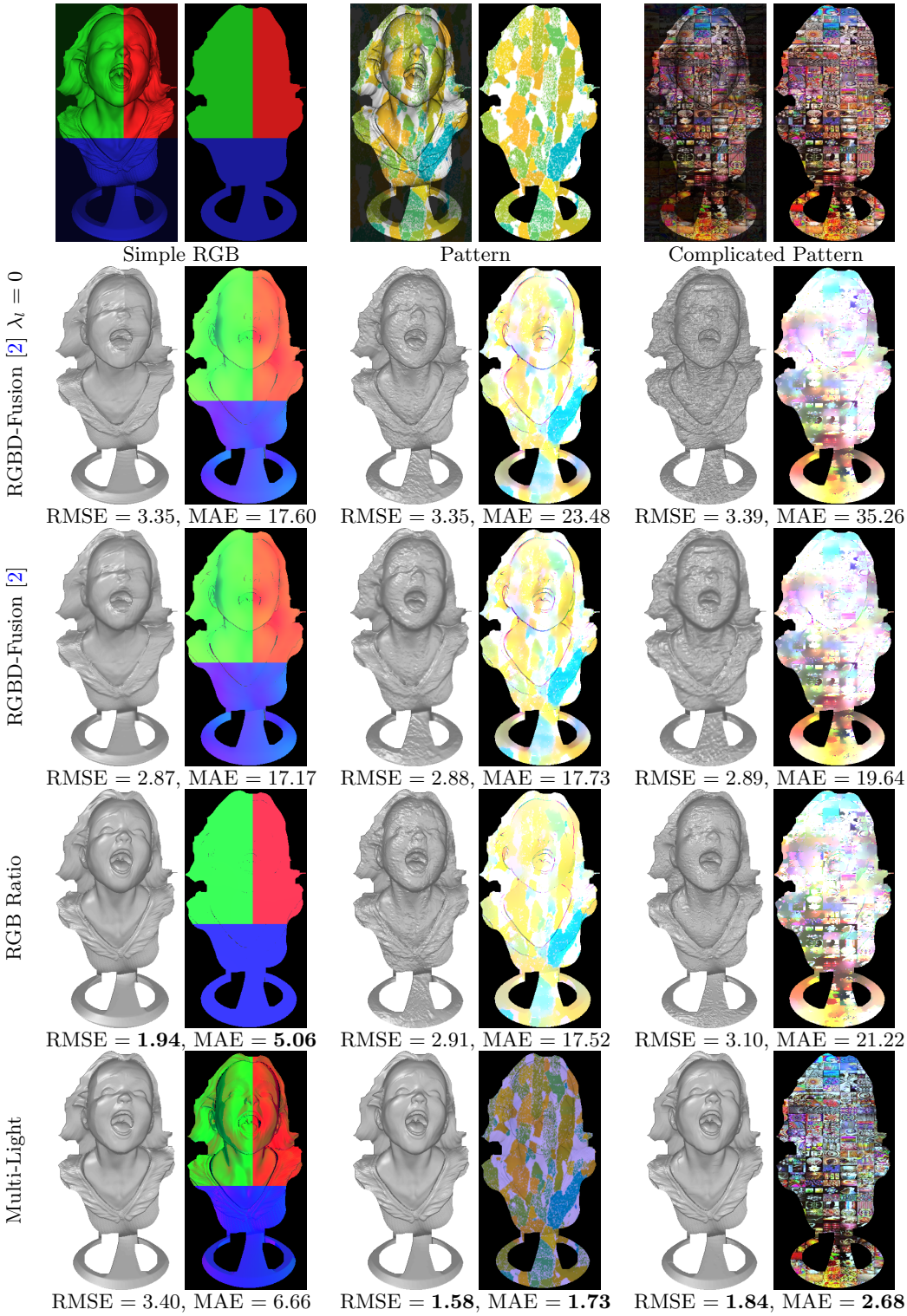


Figure 4.2: Evaluation of our two proposed methods RGB ratio and Robust Multi-Light method against our implementaion of RGBD-Fusion [2], in three different albedos from simple to complicated. Our proposed methods outperform RGBD-Fusion in all tests with respect to both RMSE and MAE. The reference errors of input are 3.35 for RMSE and 16.75 for MAE.

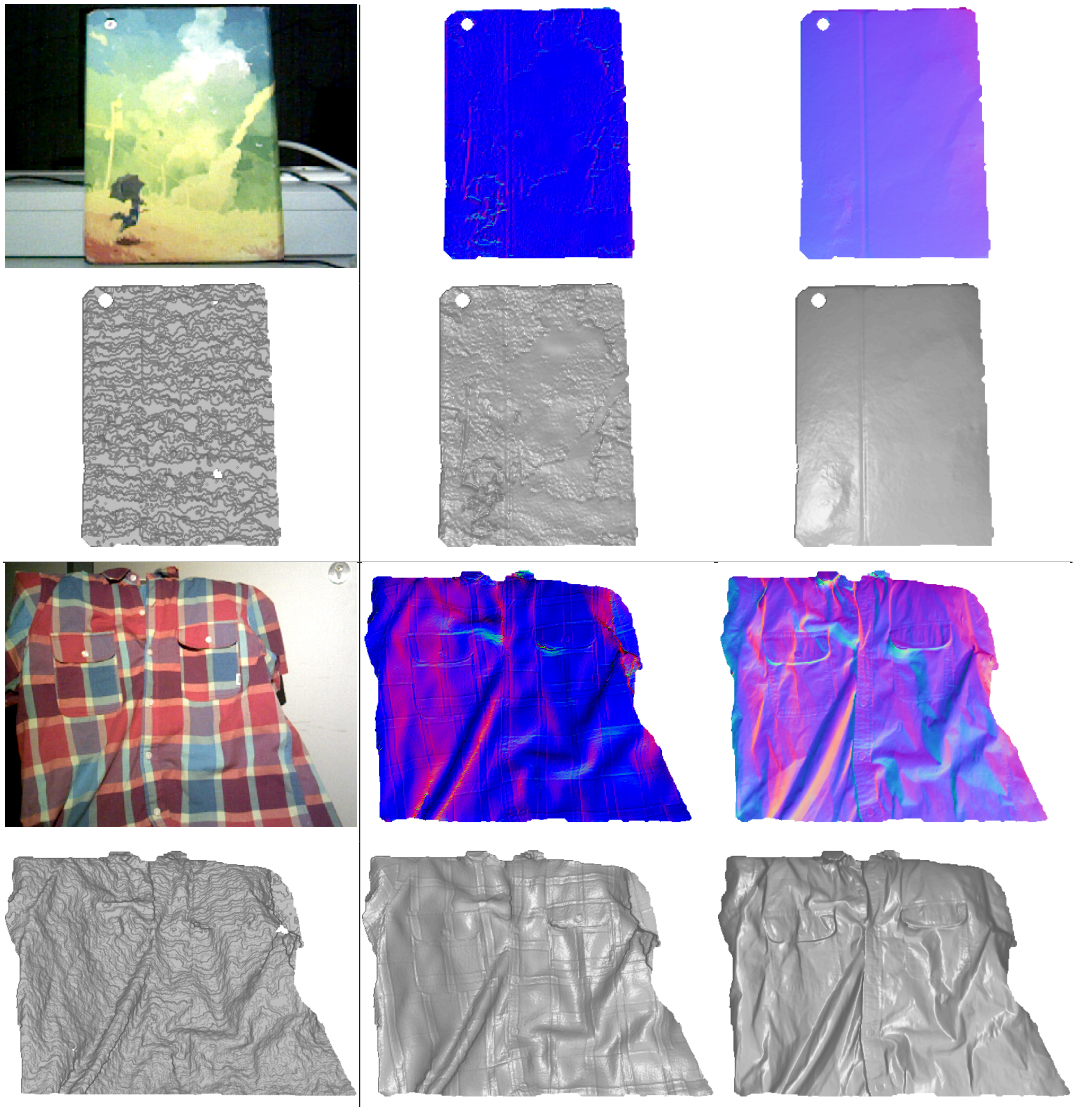
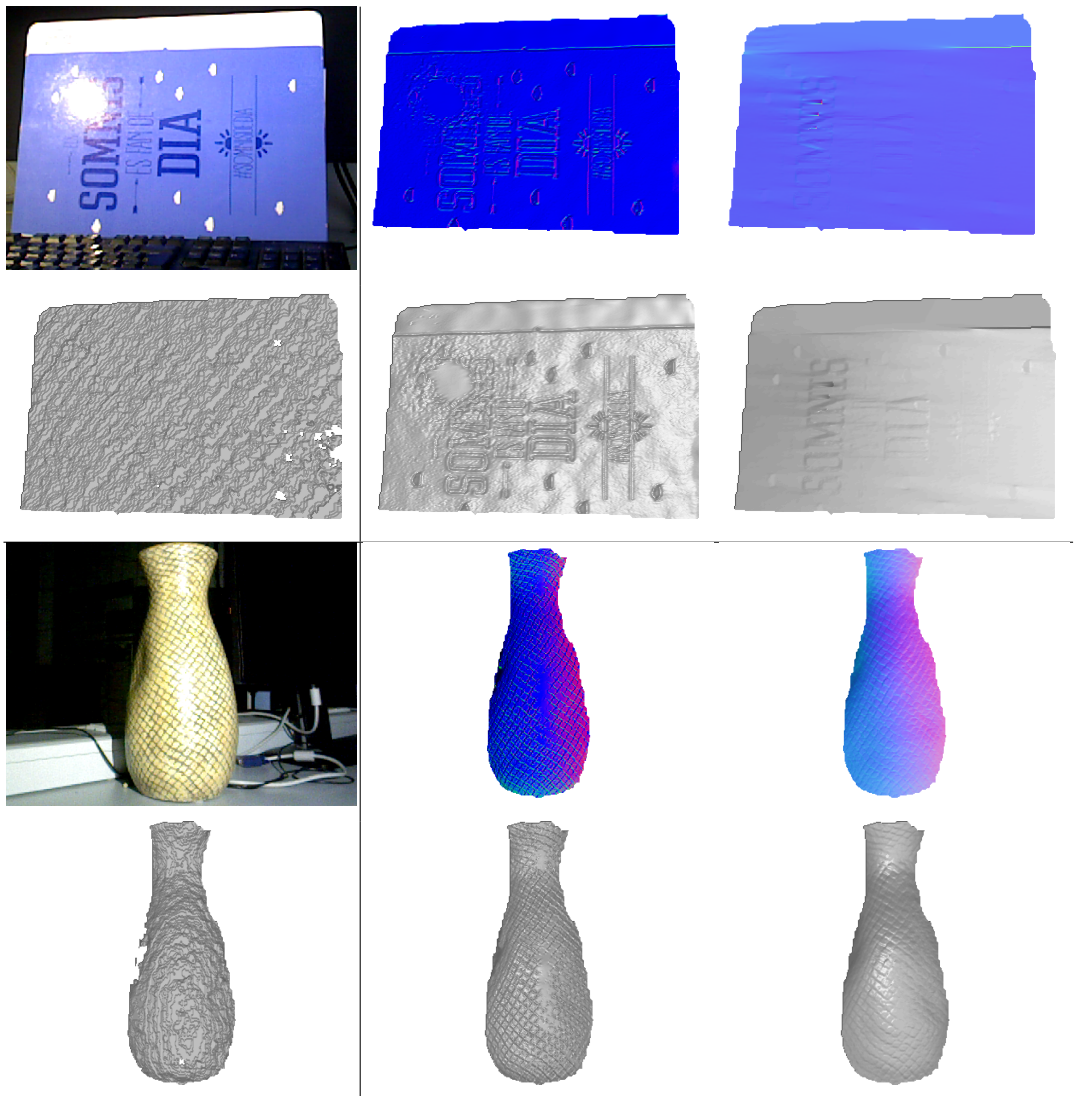


Figure 4.3: Comparison our multi-light model with RGBD-Fusion in two specular objects. On the first column, the RGB images of the folder and the vase are ones of the 10 various illuminations. First and third rows correspond to the surface normal from the refined depth, while second and fourth are the refined depth.

4.2.1 Complicated albedo objects

4.2.2 Specular objects (non-Lambertian object)



Input RGBD-Fusion [2] Proposed Multi-Light Model

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.

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 [25] or a deep denoising neural network [26].

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

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] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.