High Quality Depth Refinement with Color Photometric Stereo

Songyou Peng

Supervised by:

Dr. Yvain Quéau Prof. Daniel Cremers

figures/tum_logo.jpg

Computer Vision Group

Department of Computer Science

Technical University of Munich

figures/vibot_logo.jpg

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

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) [1]

Illustrate with the an image from MIT intrinsic dataset [2]

SH model is an extension of Lambertian model

`https://pdfs.semanticscholar.org/7b8d/fc5d6e276f8048bb53b4a5e0611019570f1b.pdf` [3]

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] [4].

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 [5]. (well, actually should check this one [6]) 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 [7]

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 [8] `https://classes.soe.ucsc.edu/cmps290b/Fall05/readings/Woodham80c.pdf`

uncalibrated light:

Hayakawa 94 [9] `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 [10]: `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 [11] use entropy `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.7264&rep=rep1&type=pdf`

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

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

[14] 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.* [15] 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 [16]. 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).

figures/methodology/inpainting_rgb.png

(a) RGB image

figures/methodology/inpainting_depth.pdf

(b) Input depth image

Figure 3.1: The input RGB and depth image of a vase. The brighter color on the image (b), the higher depth values.

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 [16] 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 [16] 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:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.4}$$

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.

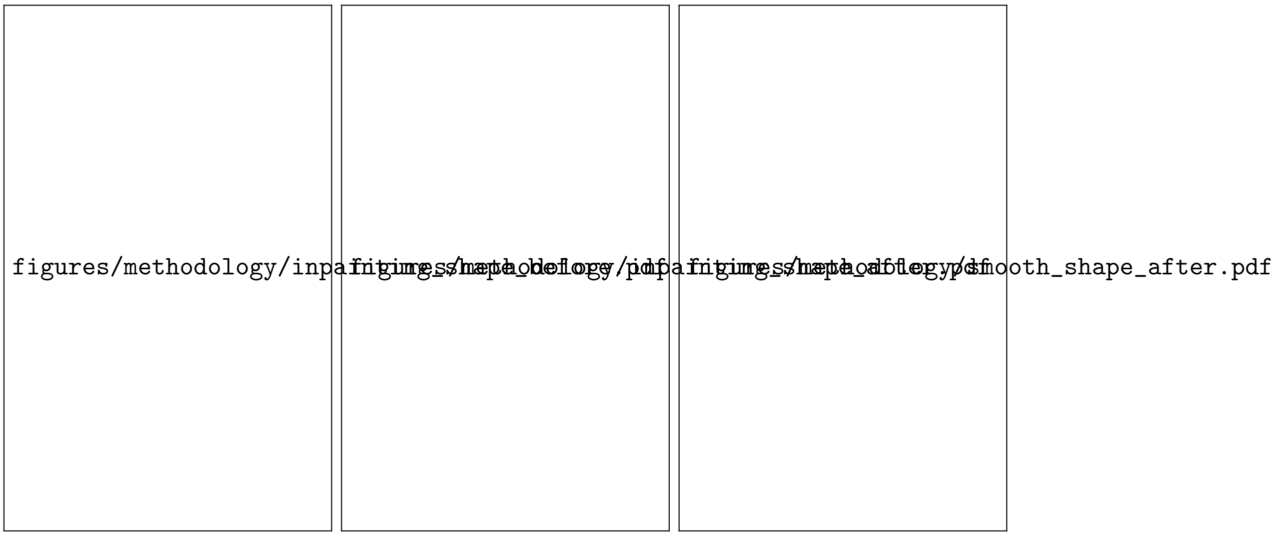
3.1.2 Depth denoising

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 [15, 17–21], 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.* [15] 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

In the following part of this section, we will explain the method and more our implementation details can be found in Appendix A.

3.2.1 Light estimation

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 [3] 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:

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

where I is the irradiance of the objects, which is represented as the intensity values, ρ 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. Here the surface normal $\mathbf{n}$ is formulated with orthographic projection, i.e.

$$\mathbf{n} = \frac{1}{\sqrt{1 + |\nabla z|^2}} \begin{pmatrix} \nabla z \\ -1 \end{pmatrix} \tag{3.7}$$

∇z 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 $\mathbf{l}^\top \mathbf{n} + \varphi$ in Eq. 3.6 as the shading $S(x, y)$. Therefore, we have:

$$S(x, y) = \mathbf{l}^\top \mathbf{n} + \varphi = \mathbf{s} \cdot \tilde{\mathbf{n}} \tag{3.8}$$

where

$$\mathbf{s} = \begin{pmatrix} \mathbf{l} \\ \varphi \end{pmatrix} \quad \tilde{\mathbf{n}} = \begin{pmatrix} \mathbf{n} \\ 1 \end{pmatrix} \tag{3.9}$$

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:

$$\min_{\mathbf{s}} \|I - \mathbf{s} \cdot \tilde{\mathbf{n}}\|_2^2 \tag{3.10}$$

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 colors, which we treat it 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. The overall regularized minimization problem for albedo estimation is:

$$\min_{\rho} \|I - \rho S\|_2^2 + \lambda_\rho \| \sum_{k \in \mathcal{N}} \omega_k (\rho - \rho_k)\|_2^2 \tag{3.11}$$

where k indicates the neighbouring index of a certain pixel, which 4-connectivity is chosen 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.

$$\omega_k = \exp\left(- \frac{\|I - I_k\|_2^2}{2\sigma_I^2} - \frac{\|z - z_k\|_2^2}{2\sigma_z^2} \right) \tag{3.12}$$

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. Now our minimization problem with respect to the depth z can be written as below. Except for the SFS term, we add a data fidelity term to enable our refined surface close to the input and a smoothness term to make sure that there is no strong discontinuity in the output.

$$\min_z \|I - \rho(\mathbf{l}^\top \frac{1}{\sqrt{1+|\nabla z|^2}} \begin{pmatrix} \nabla z \\ -1 \end{pmatrix} + \varphi)\|_2^2 + \lambda_z \|z - z_0\|_2^2 + \lambda_l \|\Delta z\|_2^2 \tag{3.13}$$

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. 3.13 is linear again. We can solve the linear system using any fast linear optimization method. In each iteration t, this process can be represented like:

$$\begin{aligned} \mathbf{n}^{(t)} &= w^{(t)} \begin{pmatrix} \nabla z^{(t)} \\ -1 \end{pmatrix} \\ w^{(t)} &= \frac{1}{\sqrt{1+|\nabla z^{(t-1)}|^2}} \end{aligned} \tag{3.14}$$

And now the depth refinement problem in Eq. 3.13 is reformulated as below in each iteration:

$$\min_z \|I - \rho(\mathbf{l}^\top \frac{1}{\sqrt{1+|\nabla z^{(t-1)}|^2}} \begin{pmatrix} \nabla z \\ -1 \end{pmatrix} + \varphi)\|_2^2 + \lambda_z \|z - z_0\|_2^2 + \lambda_l \|\Delta z\|_2^2 \tag{3.15}$$

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. Therefore, we can write an overall energy function for this RGBD-Fusion like method:

$$E(\rho, z, \mathbf{s}) = \|I - \rho \mathbf{s}^\top \tilde{\mathbf{n}}(z)\|_2^2 + \lambda_\rho \|\sum_{k \in \mathcal{N}} \omega_k (\rho - \rho_k)\|_2^2 + \lambda_z \|z - z_0\|_2^2 + \lambda_l \|\Delta z\|_2^2 \tag{3.16}$$

Finally, the whole process of RGBD-Like method has been described in Alg. 1.

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. 3.10}
2: Estimate albedo, $\rho = \arg\min_{\rho} \; E(z_0, \mathbf{s})$ {Eq. 3.11}
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. 3.13}
6: $\quad t := t + 1$
7: **end while**

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

3.3 Proposed method I: RGB Ratio Model

3.3.1 Motivation

From what we described in the previous section, it is not difficult to find the limitations for the RGBD-Like method 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.

- 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 can find a way to take better advantages of all three channels.

- The most important inspiration for us to propose RGB ratio model 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 solves 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.

According to the discussion above, 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. 3.3.

figures/methodology/ratio_setup.JPG

(a) LED lights setup

figures/methodology/ratio_setup_view.png

(b) A color image with our lighting setup

Figure 3.3: Illustrations for the pre-processing on the depth of the vase.

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.

$$\begin{aligned} I_R &= \rho_R(\mathbf{l}_R^\top \mathbf{n} + \varphi_R) \\ I_G &= \rho_G(\mathbf{l}_G^\top \mathbf{n} + \varphi_G) \\ I_B &= \rho_B(\mathbf{l}_B^\top \mathbf{n} + \varphi_B) \end{aligned} \tag{3.17}$$

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

$$\frac{I_R - \rho_R\varphi_R}{I_G - \rho_G\varphi_G} = \frac{\rho_R \mathbf{l}_R^\top \mathbf{n}}{\rho_G \mathbf{l}_G^\top \mathbf{n}} \tag{3.18}$$

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 3.18 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.

$$\mathbf{n} = \begin{pmatrix} fz_x \\ fz_y \\ -1 - \tilde{x}z_x - \tilde{y}z_y \end{pmatrix} \tag{3.19}$$

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.

Now we will explain our proposed algorithm based on the new ratio model.

3.3.2 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 then the albedo ρ^0 with Eq. 3.11 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. 3.18 as follows:

$$\begin{aligned} I_G\mathbf{l}_R^\top\mathbf{n}\rho_R - I_R\mathbf{l}_G^\top\mathbf{n}\rho_G &= \rho_R\rho_G(\varphi_G\mathbf{l}_R^\top\mathbf{n} - \varphi_R\mathbf{l}_G^\top\mathbf{n}) \\ I_B\mathbf{l}_G^\top\mathbf{n}\rho_G - I_G\mathbf{l}_B^\top\mathbf{n}\rho_B &= \rho_G\rho_B(\varphi_B\mathbf{l}_G^\top\mathbf{n} - \varphi_G\mathbf{l}_B^\top\mathbf{n}) \\ I_R\mathbf{l}_B^\top\mathbf{n}\rho_B - I_B\mathbf{l}_R^\top\mathbf{n}\rho_R &= \rho_B\rho_R(\varphi_R\mathbf{l}_B^\top\mathbf{n} - \varphi_B\mathbf{l}_R^\top\mathbf{n}) \end{aligned} \tag{3.20}$$

For each pixel, we can reformulate the Eq. 3.20 to a matrix form:

$$\begin{pmatrix} I_G\mathbf{l}_R^\top\mathbf{n} & -I_R\mathbf{l}_G^\top\mathbf{n} & 0 \\ 0 & I_B\mathbf{l}_G^\top\mathbf{n} & -I_G\mathbf{l}_B^\top\mathbf{n} \\ -I_B\mathbf{l}_R^\top\mathbf{n} & 0 & I_R\mathbf{l}_B^\top\mathbf{n} \end{pmatrix} \begin{pmatrix} \rho_R \\ \rho_G \\ \rho_B \end{pmatrix}_{3\times 1} = \begin{pmatrix} \rho_R\rho_G(\varphi_G\mathbf{l}_R^\top\mathbf{n} - \varphi_R\mathbf{l}_G^\top\mathbf{n}) \\ \rho_G\rho_B(\varphi_B\mathbf{l}_G^\top\mathbf{n} - \varphi_G\mathbf{l}_B^\top\mathbf{n}) \\ \rho_B\rho_R(\varphi_R\mathbf{l}_B^\top\mathbf{n} - \varphi_B\mathbf{l}_R^\top\mathbf{n}) \end{pmatrix} \tag{3.21}$$

which can be denoted as $\mathbf{A}_\rho \cdot \rho = \mathbf{b}_\rho$ and for the sake of simplicity, ρ in this section represents the stack of RGB three albedos.

To acquire the RGB albedos, some regularization terms are required similar to Eq. 3.11. The minimization problem of color albedo in each iteration now becomes:

$$\rho^{(t)} = \arg\min_{\rho} \|\mathbf{A}_{\rho}^{(t-1)}\rho - \mathbf{b}_{\rho}^{(t-1)}\|^2 + \lambda_{\rho}^{1}\|\omega\nabla\rho\|^2 + \lambda_{\rho}^{2}\|\rho - \rho^{(t-1)}\|^2 \tag{3.22}$$

where the weight $\omega = [\omega_R, \omega_G, \omega_B]$, which can be denoted as:

$$\omega_i = \exp(-\frac{\sigma_i||\nabla I_i||^2}{\max||\nabla I_i||^2}), \quad i \in \{R, G, B\} \tag{3.23}$$

σ is a tuning parameter for each channel.

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

1. One observation about Eq. 3.21 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.

2. Instead of using anisotropic Laplacian regularization in RGBD-Like method, the smoothness term in Eq. 3.22 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 $\|\rho - \rho^{(t-1)}\|^2$, 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.

need to add the albedo estimation 1, ground truth, 2, without regularization, 3, without weighting, 4 with weighting

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. 3.18 with the surface normal $\mathbf{n}$ as the argument:

$$\begin{aligned} \rho_G(I_R - \rho_R\varphi_R)\mathbf{l}_G^T\mathbf{n} - \rho_R(I_G - \rho_G\varphi_G)\mathbf{l}_R^T\mathbf{n} = 0 \\ \rho_B(I_G - \rho_G\varphi_G)\mathbf{l}_B^T\mathbf{n} - \rho_G(I_B - \rho_B\varphi_B)\mathbf{l}_G^T\mathbf{n} = 0 \\ \rho_R(I_B - \rho_B\varphi_B)\mathbf{l}_R^T\mathbf{n} - \rho_B(I_R - \rho_R\varphi_R)\mathbf{l}_B^T\mathbf{n} = 0 \end{aligned} \tag{3.24}$$

since the normal $\mathbf{n}$ now is a function of z, Eq. 3.24 can be actually simplified as below (the derivation details can be found in Appendix A):

$$\Psi z = 0 \tag{3.25}$$

Correspondingly, the minimization for perspective model becomes:

$$z^{(t)} = \arg\min_{z} ||z - \mathbf{z}_0^2||^2 + \lambda||\Psi\tilde{\mathbf{z}}||^2 \tag{3.26}$$

3.4 Proposed method II: Robust Lighting Variation Model without Regularization

3.4.1 Limitations for previous methods

- *LEDs have to be set up far away from each other.*
- *Natural illumination is a problem.*
- *Always need to turn off the auto white balance.*
- *Only feasible for the simple albedo cases! Biggest problem because the depth is not really correct with the wrong albedo.*
- *Non specular objects*

3.4.2 Depth super-resolution

Chapter 4

Results and Evaluation

4.1 RGB-D Cameras

4.1.1 General

4.1.2 ASUS Xtion PRO LIVE

4.2 Shape from Shading & Photometric Stereo

4.3 Intrinsic Image Decomposition

4.4 Super-resolution Imaging

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

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

Bibliography

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[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] 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.

[25] 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.