3D Reconstruction in Scattering Media

Yuki Fujimura

Abstract

This dissertation discusses 3D reconstruction in scattering media. In the field of computer vision, tasks to obtain 3D information such as an object shape, surface normals, and scene depth are referred to as 3D reconstruction. Existing 3D reconstruction methods are basically designed for clear scenes. On the other hand, under bad weather conditions such as foggy scenes, or through murky water, the visibility of the scene is degraded. Such environments are referred to as scattering media. Light traveling through scattering media get scattered and attenuated by suspended particles, and thus the contrast of images captured in scattering media is reduced. Conventional 3D reconstruction methods are affected by the image degradation in scattering media. This dissertation presents image formation models for such degradation and proposes methods to enable 3D reconstruction in scattering media. 3D reconstruction methods can be divided into three categories on the basis of their principles, i.e., disparity-, shading-, and Time-of-Flight (ToF) -based method. We apply each method to scattering media with an appropriate physics-based scattering model.

We first formulate a physics model of light scattering and attenuation, which are typical phenomena in scattering media. This formulation leads to the single scattering model and atmospheric scattering model commonly used in image processing and computer vision. The difference between these two models is the requirement of active light sources. The atmospheric scattering model is used for disparity-based methods where the system consists of only cameras without active light sources. On the other hand, the single scattering model is used for shading- and ToF-based methods that require active light sources.

For a disparity-based method, we discuss multi-view stereo (MVS) in scattering media. MVS methods are used for reconstructing the 3D geometry of a scene from multiple images. They exploit the dense pixel correspondence between multiple images. We use a learning-based MVS method in scattering media, the input of which is a cost volume that is constructed by sweeping a fronto-parallel plane to a camera in a scene and evaluates the photometric consistency between multiple cameras under the assumptions that the scene lies on each plane. In scattering media, the ordinary cost volume however leads to undesirable results due to image degradation. To solve this problem, we propose a novel cost volume for scattering media, called the dehazing cost volume. Differing from the ordinary cost volume, our dehazing cost volume can compute the cost of photometric consistency considering image degradation by restoring input images with the depth of each swept plane under the atmospheric scattering model. We also propose a method of estimating scattering parameters, such as airlight, and a scattering coefficient, which are required for our dehazing cost volume. The output depth of a network with our dehazing cost volume can be regarded as a function of these parameters; thus, they are geometrically optimized with a sparse 3D point cloud obtained at a structure-from-motion step.

For a shading-based method, we discuss photometric stereo in scattering media. Photometric stereo reconstructs surface normals from images captured under different lighting conditions. Differing from the disparity-based method, photometric stereo requires active light sources such as spotlights, and thus the image degradation is modeled with the single scattering model. However, the analysis of the single scattering model is more difficult than that of the atmospheric scattering model. Specifically, the computation of shape-dependent forward scatter in highly turbid media is infeasible. For the efficient computation of forward scatter, we propose the analytical solution of forward scatter with lookup tables. The effect of forward scatter is then divided into a shape-dependent term and a global constant term. This formulates the image degradation as a sparse linear system, which can be solved efficiently. We develop an iterative algorithm where a forward scatter removal and 3D shape reconstruction are preformed alternately.

For a ToF-based method, we discuss depth measurement in scattering media with a continuous-wave ToF camera. Continuous-wave ToF cameras emit sinusoidal signals and observe amplitude of received signals and phase-shift between these signals. Since the phase shift depends on an optical path, we can reconstruct depth from the phase shift. Similar to common RGB cameras, however, the observed signal in scattering media includes a scattering component due to light scattering. The effect of scattering media is thus modeled in amplitude and phase space under the single scattering model. We assume that a target scene consists of an object region and a background that only contains a scattering component. We also introduce two priors to estimate the scattering component: first, the scattering component can be approximated using a quadratic function in a local image patch, and second, the scattering component has a symmetrical characteristic in an overall image. Scene segmentation of the object region and background is then formulated as robust estimation where the object region is regarded as outliers, and it enables the simultaneous object region estimation and depth recovery on the basis of an iteratively reweighted least squares optimization scheme.

Through extensive experiments with synthetic and real data that is captured in actual underwater or foggy scenes, we evaluate the performance of the proposed methods. Especially, the proposed methods is effective for highly turbid media or distant scenes where captured images are significantly affected by light scattering and attenuation.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Associate Professor Masaaki Iiyama for the continuous support during my academic study at Kyoto University. Since I started my research at Kyoto University, his immense knowledge and plentiful experience have encouraged me in all the time of academic life over almost six years. This research project and dissertation would not have been realized without his persistent help.

I would like to express my appreciation to Professor Emeritus Michihiko Minoh. He was my first supervisor at Kyoto University. He has given me continuous support and valuable guidance throughout my studies even after he moved to RIKEN. I would like to thank the rest of my dissertation committee, Professor Yuichi Nakamura and Professor Ko Nishino, for their time and valuable feedback on a preliminary version of this dissertation.

I would also like to thank Dr. Hidekazu Kasahara and all the current and former colleagues of Minoh laboratory and Iiyama laboratory for their helpful discussion on my research and all the great times that we have shared.

Finally, I would like to thank my parents, Ichiro and Reiko, for their understanding and support during my long term studies. I could not complete this dissertation without their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For intelligent systems, it is important to understand their surroundings. When they interact or avoid collisions with objects, three-dimensional (3D) information plays an important role. In the field of computer vision, tasks to obtain 3D information such as an object shape, surface normals, and scene depth are referred to as 3D reconstruction methods. The typical input of the 3D reconstruction methods is a single or multiple two-dimensional (2D) images captured by RGB cameras. The applications of such techniques include autonomous mobile robots and self-driving vehicles.

Existing 3D reconstruction methods are basically designed for clear scenes. On the other hand, under bad weather conditions such as foggy scenes, or through murky water, the visibility of the scene is degraded. Figure 1.1 shows an example of an image captured under a foggy scene. Such environments are referred to as scattering media. Light traveling through scattering media get scattered and attenuated by suspended particles, and thus the contrast of images captured in scattering media is reduced.

This dissertation discusses 3D reconstruction in scattering media. This enables many applications in difficult scenes, for exmaple, drones and self-driving vehicles under bad weather, or autonomous underwater vehicles. However, conventional 3D reconstruction methods are affected by the image degradation in scattering media. In this dissertation, we present image formation models with such degradation and propose methods to enable 3D reconstruction in scattering media.

1.1 3D reconstruction

First of all, we overview existing 3D reconstruction methods. As mentioned above, their typical input is a single or multiple 2D images captured by RGB cameras. On the other hand, some methods use special sensors or active light sources in addition to cameras. The output representation of each method also varies, e.g., a point cloud, a surface mesh, surface normals, and a depth map.

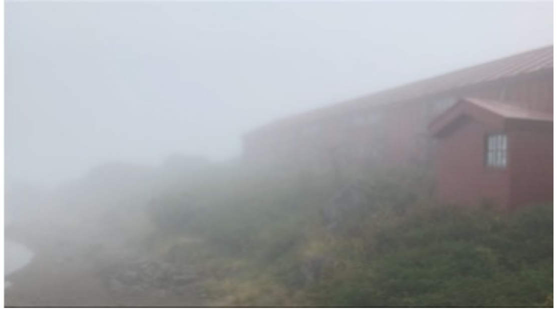


Figure 1.1: Image captured under bad weather

We divide the 3D reconstruction methods into three categories on the basis of their principles (Fig. 1.2). In the following, we summarize the advantages and disadvantages of each category. An appropriate 3D reconstuction method depends on target scenes and applications.

Methods based on disparity

Methods in this category use multiple cameras to capture multiple 2D images, which are taken as input for the methods. If a system consists of two or more than three cameras, it is called stereo or multi-view stereo (MVS) [1], respectively. The principle of 3D reconstruction of these methods is triangulation.

Traditional methods first extract feature points from input images. These featrue points often correspond to edges in the images, e.g., textures or object boundaries. A typical featrue extraction algorithm used in computer vision is the scale invariant feature transform (SIFT) [2], where the extracted features are robust to a scale, rotation, and a lighting condition in a scene. The extracted feature points are then matched between the images to get correspondences. If the positional relationship between the camreas is known, 3D points corersponding to the feature points in the images can be computed on the basis of triangulation.

Instead of using multiple cameras, we can move a single camera to get an image sequence. Structure-from-motion (SfM) [3] takes such an image sequence as input and estimates the camera motion and a scene structure simultaneously. Traditional SfM methods estimate only a sparse 3D point cloud, which corresponds to extracted 2D feature points. Therefore, after SfM estimates the camrea motion, MVS methods are often followed to get dense 3D shapes of objects.

Projector-camera systems [4] can also be used for 3D reconstruction on the same principle. These methods consist of a single camera and a single projector. The projector emits a stripes-like structured pattern onto an object. This pattern is matched with an image captured by the camera to make correspondence. For the camera-only methods, it is difficult to reconstruct objects with textureless surfaces because feature extraction based on edges would fail. On

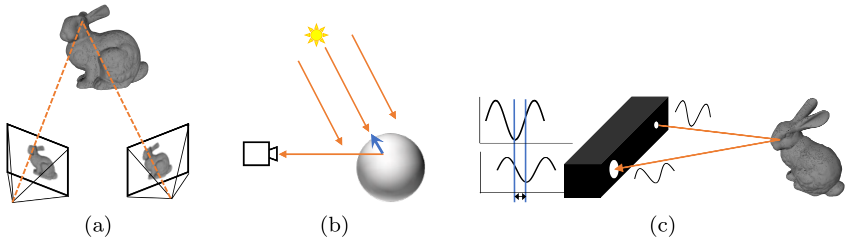


Figure 1.2: Categories of 3D reconstruction. (a) methods based on disparity. (b) methods based on shading. (c) methods based on ToF.

the other hand, the projector-camera methods can deal with such surfaces by projecting patterns onto the surfaces.

Methods based on shading

Methods in this category leverage brightness on object surfaces to infer the 3D shape. As well as the disparity-based methods, these methods also take 2D images as input, while they directly use the pixel intensity of the input images.

These methods are based on the fact that light intensity reflected on the object surface depends on its surface normal and lighting direction. The simplest reflectance model is referred to as the Lambertian model. Intuitively, the observed image is brightest when the surface normal is parallel to the lighting direction. On the other hand, the observed image gets darker as the angle between the surface normal and lighting direction becomes larger.

The most basic approach is called shape from shading [5], which takes a image captured by a single camera as input, while the reconstructed shape has uncertainty whether color gradation stems from shading or original object color. To eliminate this uncertainty, photometric stereo [6] takes multiple images captured under different lighting conditions as input.

The typical output representation of the shading-based methods is surface normals. Differing from the disparity-based methods, these methods can be applied to textureless surfaces because feature matching is not required. In addition, dense reconstruction is easily achieved because the surface normals are computed in a pixel-wise manner. On the other hand, there are several disadvantages of these methods. First, the traditional methods require active light sources like spotlights, and their light directions must be computed beforehand. This limits the application within a controlled environment. Moreover, it is difficult to reconstruct surfaces with special reflectance properties such as view-dependent specular reflection or subsurface scattering because the Lambertian model does not hold in such cases. Interreflection and shadow also reduce the accuracy of these methods. Another limitation is that these methods estimate only gradient information, and thus the integration of the surface normals is required to get the object shape. This limits the possible class of shape reconstuction, e.g., depth discontinuity cannot be reconstructed with these shading-based methods. Some prior knowledge is required if we want to reconstruct absolute depth.

Methods based on time-of-flight (ToF)

Instead of using RGB cameras, we can use special sensors that are designed to measure scene depth. Typical sensors are based on time-of-flight (ToF), that is, a light source within the sensor emit a signal into a target scene and receives the reflected light. The scene depth can be computed using the time difference between the emitted and received signals. Although sensors with infrared light are sensitive to sunlight in outdoor scenes, ToF-based methods are robust to ambient light in a scene and enable real-time depth measurement.

Recently, off-the-shelf ToF cameras such as the Microsoft Kinect for Windows v2 (Kinect v2) are available at a low cost. ToF sensors with single photon avalanche diode (SPAD), which are much more expensive sensors, can be used for more detailed scene analysis on the spatio-temporal dimension. Such analysis is referred to as transient imaging, where the sensor can observe the number of photons arrived at each time stamp, the temporal resolution of which is typically from pico- to femto-seconds. The applications of such sensors include non-line-of-sight imaging [7], which enables to recognize an object behind walls or reconstruct its 3D shape.

1.2 Appropriate scattering model

This dissertation discusses 3D reconstruction in scattering media. As mentioned above, image degradation in scattering media reduces the accuracy of 3D reconstruction. For example, feature extraction and feature matching in the disparity-based methods become difficult due to the decrease of image contrast. For the shading- and ToF-based methods, light attenuation and undesirable scattered light observed at the camera has a significant influence because they directly use pixel intensity for 3D reconstruction.

It is necessary to use an appropriate image formation model in scattering media depending on an applied 3D reconstruction method. In Section 1.1, we divide 3D reconstruction methods into three groups, i.e., disparity-, shading-, and ToF-based methods. These groups can be further divided into two types by their characteristics: they require active light sources or not. For example, the systems of the disparity-based method typically consist of only cameras. On the other hand, the shading-based methods such as photometric stereo require multiple light sources. As described in Chapter 2, the requirement of active light sources is a major factor to determine the scattering model, and thus we introduce two models for the image formation in scattering media. Figure 1.3 shows the overview of this dissertation. We discuss three methods, each of which is based on disparity, shading, and ToF, respectively. For each method, the appropriate scattering model is selected based on the requirement of active light sources.

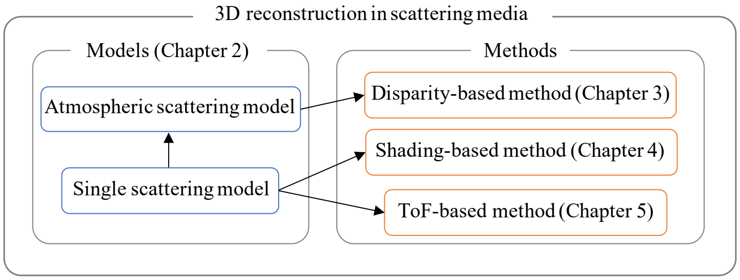

3D reconstruction in scattering media
Models (Chapter 2)
Atmospheric scattering model
Single scattering model
Methods
Disparity-based method (Chapter 3)
Shading-based method (Chapter 4)
ToF-based method (Chapter 5)

Figure 1.3: Overview of this dissertation

1.3 Scope of application

This dissertation proposes three methods for 3D reconstruction in scattering media, while they do not cover all possible situations. In this section, targe scenes are parameterized by a scene scale and medium density, and we discuss the scope of application with actual example scenes as a unified framework as shown in Fig. 1.4.

First, we discuss scattering components that should be considered for each method. Through this dissertation, light scattering is modeled with only single scattering. This means that multiple scattering is assumed to be negligibly small as described in Chapter 2. Although multiple scattering has a significant effect as medium density gets higher, we focus on environment where multiple scattering can be ignored. In Chapter 4, single scattering can also be categorized into backscatter and forward scatter [1]. Forward scatter components are defined as light that reflects on an object surface after single scattering or scattered light after reflecting on an object surface. These forward scatter components heavily depend on an object shape, the analysis of which is more difficult than that of the backscatter. The effect of the forward scatter can be ignored if medium density is low, while it should be modeled in highly turbid media.

Second, we discuss availability on different scales and medium density. Images and signals observed in scattering media are more degraded as a scene scale and medium density become larger. In foggy scenes, for example, visibility is reduced as fog is thicker or scene depth becomes larger. Large degradation obscures original signals and makes 3D reconstruction more difficult. As shown in Fig. 1.4, the proposed shading-based method focus on highly turbid media, where scattering components are dominant because original signals are heavily attenuated and forward scatter also has a significant effect. In such environment, sensitive capturing systems such as high dynamic range cameras are required, e.g., an 18-bit camera that we used in the proposed method. Regarding avail-

[1] This definition is not strictly correct because a scattering direction depends on the positional relationship between a camera and a light source. For simplicity, however, we define a backscatter component as scattered light that is observed by a camera without reaching an object surface, and a forward scatter component as the other scattered light as described in Chapter 4.

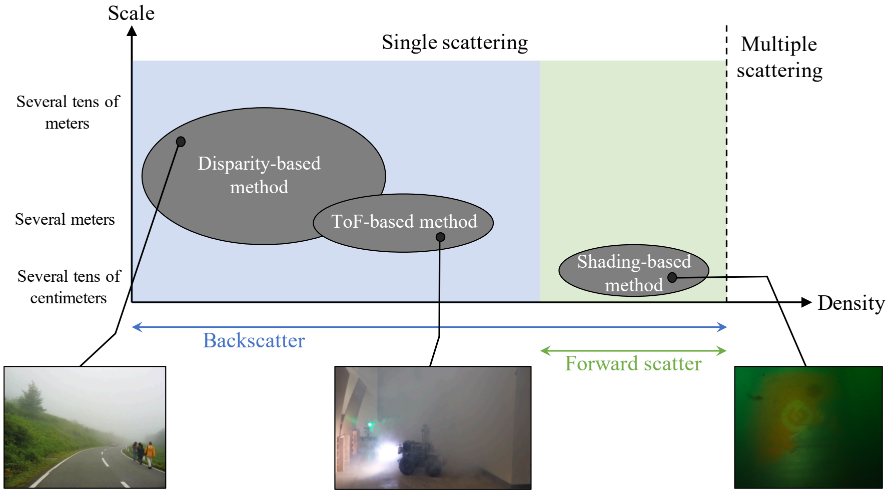


Figure 1.4: Scope of application. Targe scenes are parameterized by scene scale and medium density. Each image of example scene is cited from [8], [9], and [10].

ability, a scene scale and medium density are inversely proportional. Thus, the upper right area with a large scale and high density in Fig. 1.4 corresponds to very difficult environment. On the other hand, the lower left area with a small scale and low density corresponds to environment where degradation due to light scattering is very small and special design for 3D reconstruction methods will not be required.

Finally, we discuss the scope of the application of each method. This dissertation aims to develop disparity-, shading-, and ToF-based methods for scattering media. When medium density is relatively small and the visibility of the scene is from several meters to several tens of meters, spatial features such as edges in a captured image are easily extracted in spite of image degradation, and thus disparity-based methods are suitable for such scenes. Foggy road scenes are typical example in the real world and motor vehicles will utilize the developed method. On the other hand, as medium density becomes higher, observed signals are more degraded, which makes the extraction of spatial features difficult. In such scenes, ToF- or shading-based methods are suitable because they use not spatial features but signal values directly. The proposed ToF-based method supposes that the depth of target objects is about several meters and a typical example of such scenes is an indoor fire scene filled with smoke. The proposed shading-based method supposes that the depth of target objects is about several tens of centimeters and typical application includes underwater exploration, where strong light scattering will be caused.

1.4 Overview of dissertation

The rest of this dissertation is organized as follows: Chapther 2 discusses a physics model of light scattering and attenuation in scattering media. In particular, two scattering models commonly used in computer vision are discussed, the single scattering model [11] and the atmospheric scattering model [12]. In addition, we overview conventional image restoration methods based on these models.

Chapter 3 discusses MVS in scattering media as a disparity-based method. We use deep learning based MVS, where the input of the neural network is a cost volume to describe geometric constraints between multiple cameras. The atmospheric scattering model can be used as the image formation model because the system consists of only cameras. We discuss a novel method to incorporate this model into the cost volume to consider the geometric constraint and image degradation simultaneously.

Chapter 4 discusses photometric stereo in scattering media as a shading-based method. Photometric stereo takes multiple images captured under different lighting conditions as input, and thus the system consists of a single camera and multiple light sources. Therefore, the single scattering model can be adopted to describe the observation in scattering media. The analysis of the single scattering model is more difficult than that of the atmospheric scattering model. In this chapter, we describe the efficient computation of forward scattering in the single scattering model.

Chapter 5 discusses a ToF-based method in scattering media. In general, a camera and a light source are internally mounted in ToF devices. Therefore, we can describe the observation with the single scattering model. We use an amplitude-modulated continuous-wave ToF camera such as Kinect v2. Note that raw data of such ToF cameras is an amplitude image and a phase image for the measurement of scene depth, and thus the image formation should be modeled in amplitude and phase space.

Chapter 6 conclude the dissertation with some discussion and future works.

Chapter 2

Image Formation Models in Scattering Media

In scattering media such as fog, smoke, and murky water, light traveling through a medium interacts with suspended particles. A typical phenomenon is the scattering and attenuation. In this chapter, we formulate the light scattering and attenuation on the basis of physics. This formulation leads to the single scattering model and atmospheric scattering model commonly used in image processing and computer vision. Conventional image restoration methods based on these models are briefly discussed at the end of this chapter. Note that in this dissertation, the density of scattering media is assumed to be homogeneous.

2.1 Scattering and attenuation

First of all, we provide a simple formulation of light scattering and attenuation. For more details, please see [12].

2.1.1 Scattering

We consider an environment filled with particles. As shown in Fig. 2.1, light is incident on an unit volume in this environment, and its cross section is regarded as an unit area. Let E be irradiance at this cross section, the unit of which is denoted by [W/m^2]. Light scattering in this volume changes the traveling direction of the incident light. We denote this scattering direction by θ. Its radiant intensity $I(\theta)$ is written as follows:

$$I(\theta) = \beta(\theta)E, \tag{2.1}$$

where $\beta(\theta)$ is an angular scattering coefficient. Note that $I(\theta)$ is an energy per unit volume and unit solid angle. If this unit volume is regarded as a point light

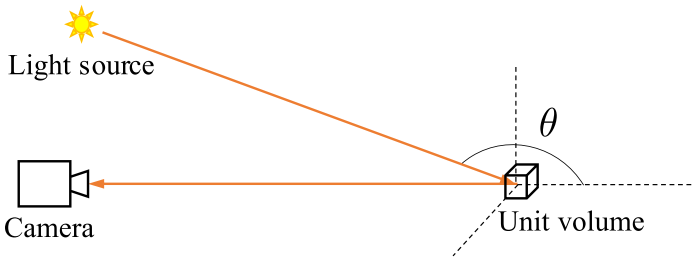


Figure 2.1: Light is incident on unit volume in scattering media. Light scattering in this volume changes traveling direction of incident light.

source, its radiant flux Φ is calculated by integrating Eq. (2.1) on a sphere:

$$\Phi = \int_{\Omega_{4\pi}} \beta(\theta) E d\omega_\theta = \beta E, \tag{2.2}$$

where β is a scattering coefficient, the unit of which is $[/m]$. β represents the proportion of the scattered light to the total amount of the incident light.

2.1.2 Attenuation

When light is incident on scattering media, light going in the same direction as the incident direction is attenuated due to light scattering. In addition, light absorption in scattering media also causes light attenuation. Let the coefficient of this absorption be α. An extinction coefficient σ can be defined as the sum of α and the scattering coefficient β as follows:

$$\sigma = \alpha + \beta. \tag{2.3}$$

Now consider attenuation from $x = 0$ to $x = d$ as shown in Fig. 2.2. The attenuation of irradiance in an infinitesimal length dx is given by

$$\frac{dE(x)}{E(x)} = -\sigma dx. \tag{2.4}$$

Let the irradiance at $x = 0$ be $E(0) = E_0$. Solving the differential equation with the integration on x along $[0, d]$ and the boundary condition $E(0) = E_0$, we can obtain the following relationship:

$$E(d) = E_0 e^{-\sigma d}. \tag{2.5}$$

This means that light is attenuated exponentially with distance in scattering media.

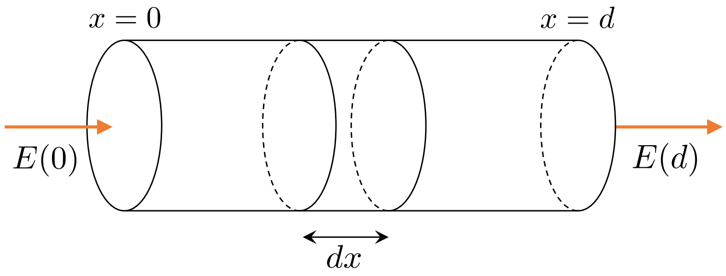


Figure 2.2: Light scattering and absorption in scattering media causes light attenuation. This attenuation is modeled by exponential function, i.e., $E(d) = E_0 e^{-\sigma d}$, where $E(0) = E_0$ and σ is extinction coefficient.

2.2 Single scattering model

In scattering media, light emitted from an light source is incident on particles and it get scattered and attenuated by these particles. On the other hand, the scattered light is also incident on other particles, and light scattering and attenuation is caused successively. This phenomenon is called multiple scattering. Although the multiple scattering is observed commonly, the analysis of this phenomenon is very complicated. In computer vision, this multiple scattering is often considered to be negligible under the assumption that the effect of more than second-order scattering is sufficiently small. In this section, we discuss the image formation model under this assumption, called the single scattering model.

Now a camera and a point light source are located in scattering media as shown in Fig. 2.3. The camera observes an infinitesimal volume at a distance y from the light source, and light scattering occurs in this volume. Let the radiant intensity of the light source be I [W/sr], the irradiance of this volume can be written as follows:

$$\frac{I}{y^2} e^{-\sigma y}, \tag{2.6}$$

where the attenuation term $1/y^2$ is called the inverse square law. $e^{-\sigma y}$ is an additional attenuation term due to scattering media. The scattered light in this volume with the scattering coefficient β can be written as follows:

$$\beta \frac{I}{y^2} e^{-\sigma y}. \tag{2.7}$$

This represents the energy per unit volume.

Let the solid angle of this volume with respect to the camera be $d\omega$. The observed area of this volume with distance x from the camera is $d\omega x^2$. Let the length of this volume be dx. The volume can be denoted by $dV = d\omega x^2 dx$. Therefore, if this volume is regarded as a point light source, the total energy

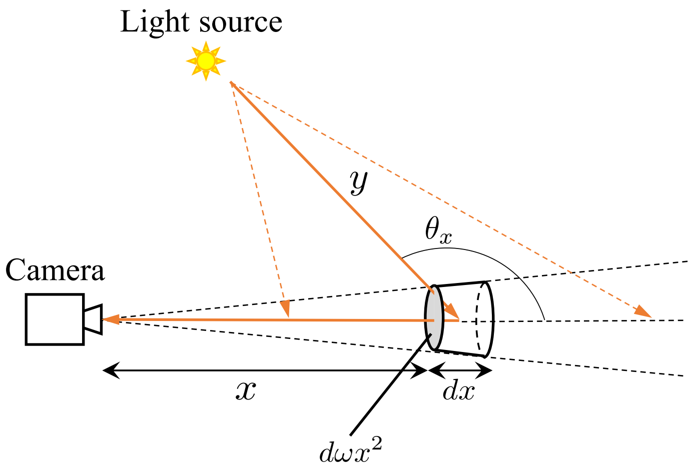


Figure 2.3: Single scattering model. Light scattering occurs at infitesimal volume. Observed area of this volume is $d\omega x^2$ and length is dx. This volume can be regarded as point light source, and its emitted light is observed as scattered light at camera. Total scattering component is sum of light scattered on light of sight.

can be written as follows:

$$\beta \frac{I}{y^2} e^{-\sigma y} dV = \beta \frac{I}{y^2} e^{-\sigma y} d\omega x^2 dx. \tag{2.8}$$

The camera observes only scattered light with scattering angle θ_x. The radiant intensity of the scattered light toward the camera via this volume is given by

$$dI(x) = P(\theta_x) \beta \frac{I}{y^2} e^{-\sigma y} d\omega x^2 dx, \tag{2.9}$$

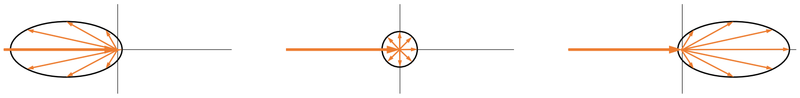
where $P(\theta)$ is a phase function, describing the angular distribution of scattered light. The following Henyey-Greenstein phase function is commonly used in computer vision:

$$P(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta)^{3/2}}. \tag{2.10}$$

The parameter g controls the distribution of scattered light as shown in Fig. 2.4. For example, $g = 0$ represents isotropic scattering, that is, the intensity of scattered light is equal in all directions. On the other hand, when $g < 0$ or $g > 0$, backscattering or forward scattering is dominant, respectively.

The irradiance of this scattered light at the camera is written as follows:

$$dE(x) = dI(x) \frac{1}{x^2} e^{-\sigma x} = P(\theta_x) \beta \frac{I}{y^2} e^{-\sigma y} d\omega e^{-\sigma x} dx, \tag{2.11}$$



(a) $g < 0$ (b) $g = 0$ (c) $g > 0$

Figure 2.4: Plot of phase function. $g = 0$ (b) represents isotropic scattering, that is, intensity of scattered light is equal in all directions. When $g < 0$ (a) or $g > 0$ (c), backscattering or forward scattering is dominant, respectively.

and the radiance at the camera is written as follows:

$$dL(x) = \frac{dE(x)}{d\omega} = P(\theta_x)\beta\frac{I}{y^2}e^{-\sigma y}e^{-\sigma x}dx. \tag{2.12}$$

The unit of which is [W/m^2·sr]. Light scattering occurs on the line of sight from the camera, and thus the total radiance from the camera to a point at distance d is the sum of these scattered light,

$$L(d) = \int_0^d dL(x) = \int_0^d P(\theta_x)\beta\frac{I}{y^2}e^{-\sigma y}e^{-\sigma x}dx. \tag{2.13}$$

Note that y is also a function of x.

When an object exists in the scene, the camera observes the reflected light on its surface. In addition, this reflected light also gets scattered and reaches the camera or other object surfaces. In Chapter 4, we give a more detailed formulation and discuss the efficient computation of these components.

2.3 Atmospheric scattering model

As described in Section 2.2, the single scattering model can be used for a scene under an active light source such as a spotlight. In this section, we discuss the atmospheric scattering model, which is simpler than the single scattering model. This model commonly used for describing the scattering effect in outdoor scenes. Overcast sky illumination is considered as a main light source in such scenes.

Similar to the single scattering model, a camera observes an infinitesimal volume. If this volume is regarded as a light source, the radiant intensity at this volume is written as follows:

$$dI(x) = k\beta d\omega x^2 dx. \tag{2.14}$$

Differing from Eq. (2.9), a light source is not modeled explicitly. On the other hand, the constant k, which is uniform in the scene, represents lighting condition including ambient light and scattering property. This model is the specific case of the single scattering model, that is, if a single distant light source such as sunlight is assumed in the single scattering model, k is given by

$$k = P(\theta)L_0, \tag{2.15}$$

where L_0 and θ denote the radiance and lighting direction of the light source. Note that these parameters are spatially-invariant.

The irradiance at the camera is also written in the same manner as follows:

$$dE(x) = dI(x)\frac{1}{x^2}e^{-\beta x} = k\beta d\omega e^{-\beta x}dx, \tag{2.16}$$

where we omitted the absorption coefficient α from the extinction coefficient ($\sigma = \beta$). This condition holds in scattering media such as fog [13]. The radiance at the camera is written as follows:

$$dL(x) = \frac{dE(x)}{d\omega} = k\beta e^{-\beta x}dx. \tag{2.17}$$

Now an object is located at distance d from the camera. The integration between $x = 0$ and $x = d$ yields the total scattered light as follows:

$$L(d) = \int_0^d dL(x) = k(1 - e^{-\beta d}). \tag{2.18}$$

If an object is located at an infinite distance, we can obtain

$$k = L_\infty, \tag{2.19}$$

where L_∞ is called airlight. Therefore, the observed scattering component at the camera is given by

$$L(d) = L_\infty(1 - e^{-\beta d}). \tag{2.20}$$

In addition to this scattering component, we consider light reflected on an object surface as follows:

$$\frac{L_\infty \rho}{d^2}e^{-\beta d}, \tag{2.21}$$

where ρ depends on object properties such as a shape, color, or reflectance. $1/d^2$ and $e^{-\beta d}$ represent attenuation due to the inverse square law and scattering media. Thus, the total observation $\hat{L}(d)$ at the camera is the sum of the reflected and scattering components as follows:

$$\hat{L}(d) = \frac{L_\infty \rho}{d^2}e^{-\beta d} + L_\infty(1 - e^{-\beta d}). \tag{2.22}$$

In the literature of image resotration, the following notation is commonly used as the atmospheric scattering model:

$$I = Jt + \mathbf{A}(1 - t), \tag{2.23}$$

where $I = [I^r, I^g, I^b]^\top \in \mathbb{R}^3$ is observed pixel RGB values, $J = [J^r, J^g, J^b]^\top \in \mathbb{R}^3$ is the pixel values of a latent clear image, and $\mathbf{A} = [A^r, A^g, A^b]^\top \in \mathbb{R}^3$ is airlight. $t = e^{-\beta z}$, where z is scene depth, is called transmission or optical depth. Compared with the single scattering model, the atmospheric scattering model is simpler formulation without complicated integral calculation. As discussed in this section, light sources are not modeled explicitly in the atmospheric scattering model. Thus, the requirement of active light sources is a major factor to determine the scattering model when constructing a 3D reconstruction method in scattering media.

2.4 Image restoration in scattering media

In the fields of image processing and computer vision, image restoration methods in scattering media have been proposed, where image degradation due to scattering media is modeled with physics-based scattering models.

As discussed in Sections 2.2 and 2.3, image degradation due to scattering media depends on scene depth. Intuitively, light reflected in a scene get attenuated with respect to the scene depth. On the other hand, scattered light observed at a camera increases with the depth because it is the sum of light scattered between the camera and scene objects. Image restoration in scattering media is thus difficult because depth estimation and image restoration is the chicken-and-egg relationship, which is discussed also in the following chapters of this dissertation.

To solve this problem, some priors or assumptions are made in most image restoration methods. For example, backscatter components are assumed to be saturated under the single scattering model in [14, 15] because the inverse square law in Eq. (2.13) reduces scattered light dramatically. Under this assumption, the scattering component no longer depends on the scene depth, and thus the scattering component are simply subtracted from a captured image by using a no-object image.

In the case of the atmospheric scattering model (Eq. (2.23)), unknown parameters, J, $\mathbf{A}$, and t, need to be estimated from I. The estimation of these parameters from a single image is an ill-posed problem. In the fields of image processing and computer vision, this task is commonly referred to as dehazing or defogging [16, 17, 18, 19, 20]. To solve the ill-posed nature, He et al. [17] proposed a dark channel prior with which a clear image having a dark pixel in a local image patch is assumed. They also assumed that transmission is the same within a local image patch. Based on these assumptions, the transmission $\tilde{t}$ in local image patch $\Omega(\mathbf{x})$ centered on pixel $\mathbf{x}$ is computed as follows:

$$J_{dark} = \min_{\mathbf{y}\in\Omega(\mathbf{x})}\Big(\min_{c\in\{r,g,b\}} J^c(\mathbf{y})\Big), \tag{2.24}$$

$$\begin{aligned} \min_{\mathbf{y}\in\Omega(\mathbf{x})}\Big(\min_{c}\frac{I^c(\mathbf{y})}{A^c}\Big) &= \tilde{t}\min_{\mathbf{y}\in\Omega(\mathbf{x})}\Big(\min_{c\in\{r,g,b\}}\frac{J^c(\mathbf{y})}{A^c}\Big) + 1 - \tilde{t} && (2.25)\\ \tilde{t} &\to 1 - \min_{\mathbf{y}\in\Omega(\mathbf{x})}\Big(\min_{c}\frac{I^c(\mathbf{y})}{A^c}\Big) \quad (J_{dark} \to 0). && (2.26) \end{aligned}$$

Note that $\mathbf{A}$ is assumed to be estimated beforehand using the brightest pixels that are considered to be the most haze-opaque regions [16]. Berman et al. [20] proposed a haze-line prior with which the same intensity pixels of the latent clear image forms a line in RGB space. This means that a point corresponding to an observed pixel intensity is a dividing point between those of the pixel intensity of a latent clear image and airlight in RGB space. The value of transmission corresponds to its dividing ratio. Airlight can be estimated in the same framework [21].

Many learning-based methods using neural networks have also recently been proposed to learn the priors of natural images from large scale training dataset [22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. The earliest work by Ren et al. [23] was proposed to learn a convolutional neural network (CNN), which takes a hazy image as input then outputs a transmission map. Similar to non-learning-based dehazing methods, a clear image is computed with this estimated transmission map and pre-estimated airlight by using the physics-based model (Eq. (2.23)). Li et al. [24] transformed Eq. (2.23) to derive an equation where unknown parameters $\mathbf{A}$ and t are incorporated into a single parameter then trained a network to estimate this parameter. This suppressed the effect of the additional estimation error of airlight. Yang et al. [28] proposed a method that bridges the gap between learning- and non-learning-based methods by making a network learn the dark channel prior.

In contrast to these physics-based approaches, methods without explicit physics-based models have also recently provided highly accurate results [26, 29, 30, 31]. Liu et al. [29] claimed that a physics-based scattering model hardly constrains the solution space and makes gradient descent optimization stop at a local minimum.

Chapter 3

Multi-view Stereo in Scattering Media

In this chapter, we discuss MVS in scattering media as a disparity-based 3D reconstruction method. MVS methods [1] are used for reconstructing the 3D geometry of a scene from multiple images. They exploits the dense pixel correspondence between multiple images.

We discuss a learning-based MVS method in scattering media. Learning-based MVS methods have recently been proposed and provided highly accurate results [32, 33, 34]. The proposed method is based on MVDepthNet [35], which is one such MVS method.

MVDepthNet estimates scene depth by taking a cost volume as input for the network. The cost volume is based on a plane sweep volume [36], i.e., it is constructed by sweeping a fronto-parallel plane to a camera in the scene and evaluates the photometric consistency between multiple cameras under the assumptions that the scene lies on each plane. As described in Chapter 2, however, an image captured in scattering media degrades; thus, using the ordinary cost volume leads to undesirable results, as shown in Fig. 3.1(b).

To solve this problem, we propose a novel cost volume for scattering media, called *the dehazing cost volume*. As described in Chapter 2, degradation due to a scattering medium depends on the scene depth. Our dehazing cost volume can restore images with such depth-dependent degradation and compute the effective cost of photometric consistency simultaneously. It enables robust 3D reconstruction in scattering media, as shown in Fig. 3.1(c).

The rest of this chapter is organized as follows: In Section 3.1, we first overview MVS methods in clear scenes then describe related work including existing stereo and MVS methods in scattering media. In Section 3.2, an ordinary cost volume and our dehazing cost volume for scattering media is discussed. In Section 3.3, we describe a method for estimating scattering parameters in the atmospheric scattering model, which is required to construct our dehazing cost volume. This can be achieved in the same framework as depth estimation with

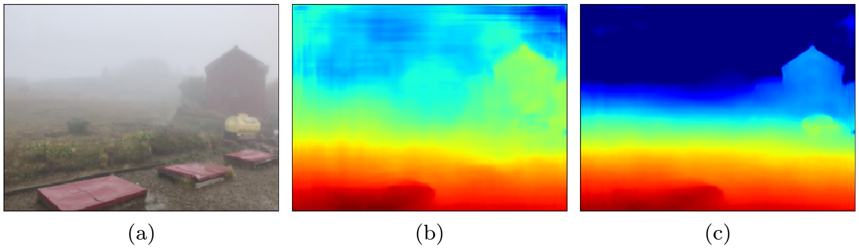


Figure 3.1: Estimated depth in scattering media. (a) Image captured in actual foggy scene. (b) Output depth of fine-tuned MVDepthNet [35] with ordinary cost volume. (c) Output depth of network with our dehazing cost volume.

our dehazing cost volume. In Section 3.4, we demonstrate the effectiveness of depth estimation with our dehazing cost volume on synthetic and real data. Finally, Section 3.5 concludes this chapter.

3.1 Related work

3.1.1 Multi-view stereo

MVS methods [1] are used for reconstructing 3D geometry using multiple cameras. In general, it exploits the dense pixel correspondence between multiple images for 3D reconstruction. The correspondence is referred to as photometric consistency and computed on the basis of the similarity measure of pixel intensity. One of the difficulties in the computation of photometric consistency is occlusion, i.e., the surface of a target object is occluded from certain cameras. This leads to incorrect correspondence and inaccurate 3D reconstruction. To solve this problem, methods have been proposed for simultaneous view selection to compute effective photometric consistency and 3D reconstruction with MVS, achieving highly accurate 3D reconstruction [37, 38].

Along with the above problem, there are many cases in which it is difficult to obtain accurate 3D geometry with traditional MVS methods. A textureless surface and an object with a view-dependent reflectance property, such as specular reflection, are typical cases. Learning-based MVS methods have recently been used to learn semantic information on large-scale training data and enable robust 3D reconstruction in such scenes.

Learning-based MVS methods often construct a cost volume to constrain 3D geometry between multiple cameras. For example, Wang and Shen [35] proposed MVDepthNet, which constructs a cost volume from multi-view images by setting one of the images as a reference image. It can take an arbitrary number of input images to construct the cost volume. The convolutional neural network (CNN) takes the reference image and cost volume as input then estimates the depth map of the reference camera. DeepMVS proposed by Huang et al. [33] first constructs a plane sweep volume, then the patch matching network is applied to the reference image and each slice of the volume to extract features to measure the correspondence, which is followed by feature aggregation networks and depth refinement with a fully connected conditional random field. Yao et al. [32] and Im et al. [34] respectively proposed MVSNet and DPSNet, in which input images are first passed through the networks to extract features, then the features are warped instead of constructing the cost volume in the image space. Our proposed method is based on MVDepthNet [35], which is the simplest and light-weight method, and we extended the ordinary cost volume and constructs our dehazing cost volume for scattering media.

3.1.2 Disparity-based method in scattering media

The proposed method is based on stereo 3D reconstruction without active light sources. There have been several works for applying such disparity-based methods to scattering media. Caraffa et al. [39] proposed a binocular stereo method in scattering media. With this method, image enhancement and stereo reconstruction are simultaneously modeled on the basis of a Markov random field. Song et al. [40] proposed a learning-based binocular stereo method in scattering media, where dehazing and stereo reconstruction are trained as multi-task learning. The features from the networks of each task are simply concatenated at the intermediate layer. The most related method to ours is the MVS method proposed by Li et al. [41]. They modeled dehazing and MVS simultaneously and regularized the output depth using an ordering constraint, which was based on a transmission map that was the output of dehazing with Laplacian smoothing.

These previous studies [39, 41] designed photometric consistency measures considering the scattering effect. However, this requires scene depth because degradation due to scattering media depends on this depth. Thus, they relied on iterative implementation of an MVS method and dehazing, which leads to large computation cost. In contrast, our dehazing cost volume can solve this chicken-and-egg problem by computing the scattering effect in the cost volume. The scene depth is then estimated effectively by taking the cost volume as input for a CNN, making fast inference possible.

3.2 MVS with dehazing cost volume in scattering media

In this section, we describe MVS in scattering media with our dehazing cost volume. We first overview the proposed method then discuss the ordinary cost volume and our dehazing cost volume, followed by implementation details.

3.2.1 Overview

MVS methods are roughly categorized by output representations, e.g., point-cloud, volume, or mesh-based reconstruction. The proposed method is formulated as depth-map estimation, i.e., given multiple cameras, we estimate a depth

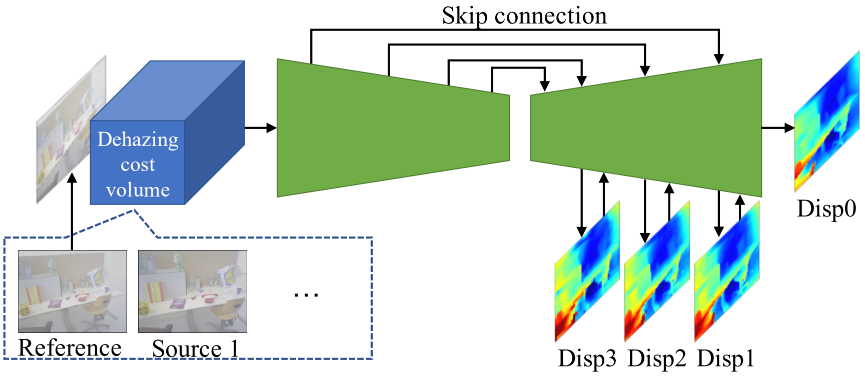


Figure 3.2: Overview of MVS in scattering media. Input of network is reference image captured in scattering medium and our dehazing cost volume. Our dehazing cost volume is constructed from reference image and source images. Network architecture of our method is same as that of MVDepthNet [35], which has encoder-decoder with skip connections. Output of network is disparity maps (inverse depth maps) at different resolutions.

map for one of the cameras. We refer to a target camera to estimate a depth map as a reference camera r and the other cameras as source cameras $s \in \{1, \cdots, S\}$, and images captured with these cameras are denoted as a reference image I_r and source images I_s, respectively. We assume that the camera parameters are calibrated beforehand.

An overview of the proposed method is shown in Fig. 3.2. Our dehazing cost volume is constructed from a hazy reference image and source images captured in a scattering medium. The network takes the reference image and our dehazing cost volume as input then outputs a disparity map (inverse depth map) of the reference image. The network architecture is the same as that of MVDepthNet [35], while the ordinary cost volume used in MVDepthNet is replaced with our dehazing cost volume for scattering media.

3.2.2 Dehazing cost volume

In this section, we explain our dehazing cost volume, which is taken as input to the network. The dehazing cost volume enables effective computation of photometric consistency in scattering media.

Before explaining our dehazing cost volume, we show the computation of the ordinary cost volume in Fig. 3.3(a). We first sample the 3D space in the reference-camera coordinate system by sweeping a fronto-parallel plane. We then back-project source images onto each sampled plane. Finally, we take the residual between the reference image and each warped source image, which corresponds to the cost of photometric consistency on the hypothesis that the scene exists on the plane. Let the image size be $W \times H$ and number of sampled depths be N. We denote the cost volume as $\mathcal{V} : \{1, \cdots, W\} \times \{1, \cdots, H\} \times \{1, \cdots, N\} \to \mathbb{R}$, and each element of the cost volume is given as follows:

$$\mathcal{V}(u, v, i) = \frac{1}{S} \sum_{s} \| I_r(u, v) - I_s(\pi_{r \to s}(u, v; z_i)) \|_1, \tag{3.1}$$

where z_i is the depth value of the i-th plane. The operator $\pi_{r \to s} : \mathbb{R}^2 \to \mathbb{R}^2$ projects the camera pixel (u, v) of the reference camera r onto the source image I_s with the given depth, which is defined as follows:

$$\begin{bmatrix} \pi_{r \to s}(u, v; z) \\ 1 \end{bmatrix} \sim z \mathbf{K}_s \mathbf{R}_{r \to s} \mathbf{K}_r^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \mathbf{K}_s \mathbf{t}_{r \to s}, \tag{3.2}$$

where $\mathbf{K}_r$ and $\mathbf{K}_s$ are the intrinsic parameters of the reference camera r and the source camera s, and $\mathbf{R}_{r \to s}$ and $\mathbf{t}_{r \to s}$ are a rotation matrix and translation vector from r to s, respectively. The cost volume evaluates the photometric consistency of each pixel with respect to the sampled depth; thus, the element of the cost volume with correct depth ideally becomes zero.

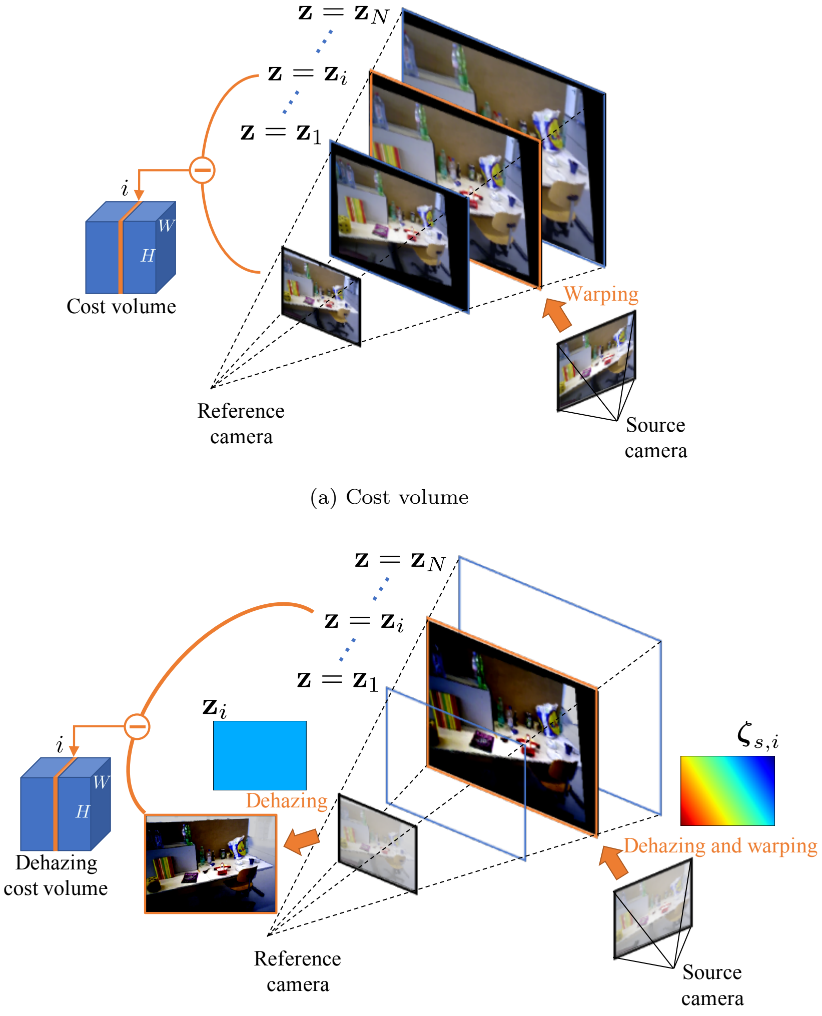
As described in Chapter 2, an observed image captured in scattering media without active light sources can be modeled with the atmospheric scattering model. Let an RGB value at the pixel (u, v) of a degraded image captured in scattering media and its latent clear image be $I(u, v) \in \mathbb{R}^3$ and $J(u, v) \in \mathbb{R}^3$, respectively. We assume that the pixel value of each color channel is within 0 and 1. We recall the atmospheric scattering model here:

$$I(u, v) = J(u, v) e^{-\beta z(u, v)} + \mathbf{A}(1 - e^{-\beta z(u, v)}), \tag{3.3}$$

where $z(u, v) \in \mathbb{R}$ is the depth at pixel (u, v), $\beta \in \mathbb{R}$ is a scattering coefficient that represents the density of a medium, and $\mathbf{A} \in \mathbb{R}^3$ is global airlight. For simplicity, we assume that $\mathbf{A}$ is given by $\mathbf{A} = [A, A, A]^\top, A \in \mathbb{R}$, i.e., the color of scattering media is achromatic (gray or white). This degradation leads to undesirable results with the ordinary cost volume defined in Eq. (3.1). In contrast, our dehazing cost volume dehazes the image and computes photometric consistency cost simultaneously. Degradation due to scattering media depends on scene depth; thus, our dehazing cost volume restores degraded images using the depth of a swept plane.

Figure 3.3(b) shows the computation of our dehazing cost volume. A reference image is dehazed directly using the depth of a swept plane. A source image is dehazed using the swept plane from a source camera view, then the dehazed source image is warped to the reference-camera coordinate system. Similar to the ordinary cost volume, we define our dehazing cost volume as $\mathcal{D} : \{1, \cdots, W\} \times \{1, \cdots, H\} \times \{1, \cdots, N\} \to \mathbb{R}$, and each element of our dehazing cost volume is given as

$$\mathcal{D}(u, v, i) = \frac{1}{S} \sum_{s} \| J_r(u, v; z_i) - J_s(\pi_{r \to s}(u, v; z_i)) \|_1, \tag{3.4}$$



(a) Cost volume

(b) Dehazing cost volume

Figure 3.3: Cost volume and dehazing cost volume. (a) Ordinary cost volume is constructed by sweeping fronto-parallel plane in reference-camera coordinate. Cost of photometric consistency is simply computed as residual between reference image and warped source image on each swept plane $\mathbf{z} = \mathbf{z}_i$. (b) In our dehazing cost volume, reference image is dehazed using sampled depth, $\mathbf{z}_i$, which is constant over all pixels. Source image is dehazed using depth of swept plane from source-camera view, then dehazed source image is back-projected onto plane. Cost is computed by taking residual between both dehazed images.

where $J_r(u, v; z_i)$ and $J_s(\pi_{t\to s}(u, v; z_i))$ are dehazed reference and source images. From Eq. (3.3), if A and β are estimated beforehand, they are computed as follows:

$$\begin{aligned} J_r(u, v; z_i) &= \frac{I_r(u, v) - \mathbf{A}}{e^{-\beta z_i}} + \mathbf{A}, && (3.5) \\ J_s(\pi_{r\to s}(u, v; z_i)) &= \frac{I_s(\pi_{r\to s}(u, v; z_i)) - \mathbf{A}}{e^{-\beta \zeta_{s,i}(\pi_{r\to s}(u,v;z_i))}} + \mathbf{A}. && (3.6) \end{aligned}$$

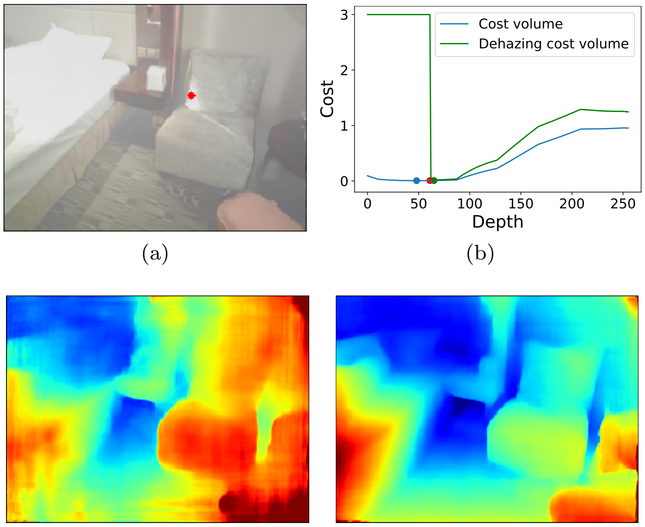
As shown in Fig. 3.3(b), the reference image is dehazed using the swept plane with depth z_i, whose depth map is denoted as $\mathbf{z}_i$. On the other hand, the source image is dehazed using $\boldsymbol{\zeta}_{s,i}$, which is a depth map of the swept plane from the source camera view. The depth $\zeta_{s,i}(\pi_{r\to s}(u, v; z_i))$ is used for the cost computation of the pixel (u, v) of the reference camera because the pixel $\pi_{r\to s}(u, v; z_i)$ on the source camera corresponds to pixel (u, v) of the reference camera. Our dehazing cost volume exploits the dehazed images with much more contrast than the degraded ones; thus, the computed cost is robust even in scattering media. In accordance with this definition of our dehazing cost volume, the photometric consistency between the latent clear images is preserved.

Our dehazing cost volume computes photometric consistency with dehazed images in the cost volume. This is similar to the previous methods [39, 41] that compute photometric consistency considering scattering effect. However, this is a chicken-and-egg problem because the effect of scattering media depends on scene depth, and they rely on iterative implementation of MVS and dehazing to compute the scattering effect. Our method, on the other hand, can compute the scattering effect using a depth hypothesis of a swept plane without an explicit scene depth, which can eliminate the iterative optimization.

Our dehazing cost volume restores an image using all depth hypotheses; thus, image dehazing with depth that greatly differs from the correct scene depth results in an unexpected image. The extreme case is when a dehazed image has negative values at certain pixels. This includes the possibility that a computed cost using Eq. (3.4) becomes very large. To avoid such cases, we revise the definition of our dehazing cost volume as follows:

$$\mathcal{D}(u, v, i) = \frac{1}{S} \sum_s \begin{cases} \|J_r(u, v; z_i) - J_s(\pi_{r\to s}(u, v; z_i))\|_1 \\ \quad \mathit{if}\ 0 \leq J_r^c(u, v; z_i) \leq 1\ \mathit{and} \\ \quad 0 \leq J_s^c(\pi_{r\to s}(u, v; z_i)) \leq 1 \quad c \in \{r, g, b\} \\ \gamma \quad \mathit{otherwise}, \end{cases} \tag{3.7}$$

where $J_r^c(u, v; z_i)$ and $J_s^c(\pi_{r\to s}(u, v; z_i))$ are the pixel values of the channel $c \in \{r, g, b\}$ of the reconstructed clear images. A constant γ is a parameter that is set as a penalty cost when the dehazed result is not contained in the domain of definition. This makes the training of the network stable because our dehazing cost volume is upper bounded by γ. We can also reduce the search space of depth by explicitly giving the penalty cost. In this study, we set $\gamma = 3$, which is the maximum value of the ordinary cost volume defined in Eq. (3.1) when the pixel value of each color channel is within 0 and 1.



(a) (b)

(c) (d)

Figure 3.4: Visualization of our dehazing cost volume. (b) Computed ordinary cost volume and our dehazing cost volume at red point in (a). In (b), red dot indicates location of ground-truth, and blue and green dots indicate minimum value of ordinary cost volume and our dehazing cost volume, respectively. (c) and (d) Output depth of MVDepthNet [35] with ordinary cost volume and our dehazing cost volume, respectively.

Figure 3.4(b) visualizes the ordinary cost volume and our dehazing cost volume at the red point in (a). Each dot in (b) indicates a minimum cost, and the red dot in (b) indicates ground-truth depth. The curve of the cost volume is smoother than that of our dehazing cost volume due to the degradation in image contrast, which leads to a depth error. Our dehazing cost volume can also reduce the search space with the dehazing constraint γ on the left part in (b), where its cost value is constantly large.

3.2.3 Network architecture and loss function

As shown in Fig. 3.2, a network takes a reference image and our dehazing cost volume as input. To compute our dehazing cost volume, we should predetermine the target 3D space for scene reconstruction and number of depth hypotheses for plane sweep. We uniformly sample the depth on the disparity space between 0.02 and 2 and set the number of samples to $N = 256$. The network architecture is the same as that of MVDepthNet [35], which has an encoder-decoder architecture with skip connections. The network outputs disparity maps at different resolutions. The training loss is defined as the sum of L1 loss between these estimated disparity maps and the ground-truth disparity map. (For more details, please refer to [35].)

Table 3.1: Network architecture of airlight estimator. Network takes single RGB image as input then outputs single scalar value A. Stride of convolution layers from conv1 to conv6 is 2. Each convolution layer except for conv8 has batch normalization and ReLU activation. glb_avg_pool denotes global average pooling layer.

Layer	Kernel	Channel	Input
conv1	7	3/16	I
conv2	5	16/32	conv1
conv3	3	32/64	conv2
conv4	3	64/128	conv3
conv5	3	128/256	conv4
conv6	3	256/256	conv5
glb_avg_pool	-	256/256	conv6
conv7	1	256/64	glb_avg_pool
conv8	1	64/1	conv7

3.3 Scattering parameter estimation

As mentioned in Section 3.2, our dehazing cost volume requires scattering parameters, airlight A and a scattering coefficient β in Eq. (3.6). In this section, we first explain the estimation of A then describe the difficulty of estimating β. Finally, we discuss the simultaneous estimation of the scattering parameters and depth with our dehazing cost volume.

3.3.1 Estimation of airlight A

We first describe the estimation of A. Although methods for estimating A from a single image have been proposed, we implement and evaluate a CNN-based estimator, the architecture of which is shown in Table 3.1. It takes a single RGB image as input, which is passed through several convolution layers with stride 2. Global average pooling is then applied to generate a $256 \times 1 \times 1$ feature map. This feature map is passed through two 1×1 convolutions to yield 1D output A. Note that each convolution layer except for the final layer (conv8) is followed by batch normalization and then by rectified linear unit (ReLU) activation. For training and test, we used the synthesized image dataset described in Section 3.4.1. Figure 3.5 shows the error histogram of A on the test dataset. In this dataset, the value of A is randomly sampled from $[0.7, 1.0]$, indicating that the estimation of A can be achieved from a single image.

3.3.2 Difficulty of estimating scattering coefficient β

In contrast to A, it is difficult to estimate β from a single image. As shown in Eq. (3.3), image degradation due to scattering media depends on β and scene depth z through $e^{-\beta z}$ with the scale-invariant property, i.e., the pairs of $k\beta$ and $(1/k)z$ for arbitrary $k \in \mathbb{R}$ lead to the same degradation. Since the depth

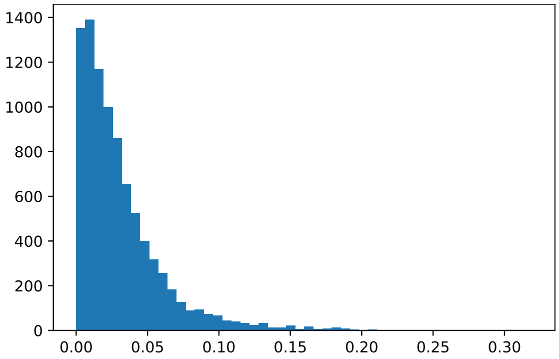


Figure 3.5: Error histogram of our airlight estimator on synthesized test dataset. Simple L1 error is computed on each estimate. In this dataset, A is randomly sampled from $[0.7, 1.0]$.

scale cannot be determined from a single image, the estimation of the scattering coefficient from a single image is infeasible.

In response to this problem, Li et al. [41] proposed a method for estimating β from multi-view images. With this method, it is assumed that a sparse 3D point cloud and camera parameters can be obtained by SfM from noticeable image edges even in scattering media. From a pixel pair and corresponding 3D point, two equations can be obtained from Eq. (3.3). Additionally, if we assume that the pixel value of the latent clear image is equal to the corresponding pixel value of the other clear image, this simultaneous equations can be solved for β. However, this multi-view-based method involves several strong assumptions. First, the pixel value of the latent clear image should be completely equal to the corresponding pixel value of the other clear image. Second, the values of the observed pixels should be sufficiently different to ensure numerical stability. This assumption means the depth values of both images should be sufficiently different, and it is sometimes very difficult to find such points. Finally, A is assumed to be properly estimated beforehand. These limitations indicate that we should avoid using the pixel values directly for β estimation.

3.3.3 Estimation with geometric information

In this study, the scattering coefficient was estimated without using pixel intensity. Our method ensures the correctness of the output depth with the estimated scattering coefficient.

As well as the MVS method proposed by Li et al. [41], a sparse 3D point cloud is assumed to be obtained by SfM in advance. Although our dehazing cost volume, which is taken as input for a network, requires A and β, this means that the network can be regarded as a function that takes A and β as variables and outputs a depth map. Now, the network with fixed parameters is denoted by $\mathcal{F}$, and the output depth can be written by $\mathbf{z}_{A,\beta} = \mathcal{F}(A, \beta)$ as a function of A

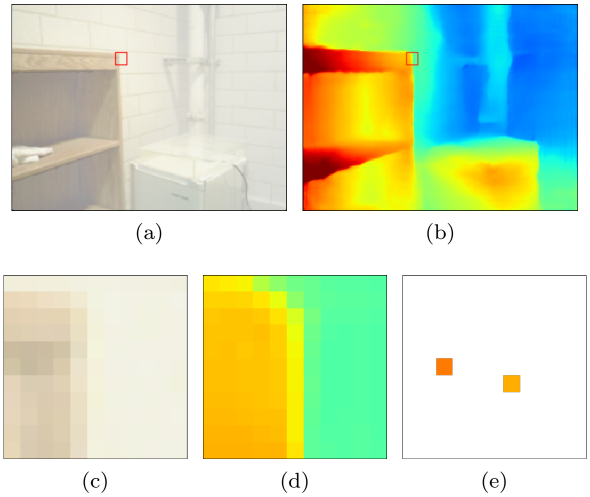


Figure 3.6: Consideration of depth discontinuities. (a) Input image. (b) Output depth with ground-truth scattering parameters. Depth discontinuities exist in red boxed region. Zoom of regions in (a) and (b) are shown in (c) and (d), respectively. (e) Depth map of sparse 3D point cloud obtained by SfM in this region. It is uncertain whether feature point obtained by SfM is located on background or foreground around depth discontinuities. This includes possibility that output depths of network and SfM are completely different such as right pixel in (e).

and β. Note that for simplicity, we omitted the input image from the notation. Let a depth map that corresponds to a sparse 3D point cloud by SfM be $\mathbf{z}_{sfm}$. The scattering parameters are estimated by solving the following optimization problem:

$$A^*, \beta^* = \underset{A,\beta}{\operatorname{argmin}} \sum_{u,v} m(u,v)\rho\Big(z_{sfm}(u,v), z_{A,\beta}(u,v)\Big), \tag{3.8}$$

where $z_*(u,v)$ denotes a value at the pixel (u,v) of a depth map $\mathbf{z}_*$, and $m(u,v)$ is an indicator function, where $m(u,v) = 1$ if a 3D point estimated by SfM is observed at pixel (u,v), and $m(u,v) = 0$ otherwise. A function ρ computes the residual between the argument depths. Therefore, the solution of Eq. (3.8) minimizes the difference between the output depth of the network and the sparse depth map obtained by SfM. A final dense depth map can then be computed with the estimated A^* and β^*, i.e., $\mathbf{z}^* = \mathcal{F}(A^*, \beta^*)$. Differing from the previous method [41], our method does not require pixel intensity because the optimization is based on only geometric information, and the final output depth is ensured to match at least the sparse depth map obtained by SfM.

We use the following function as ρ to measure the difference between depth

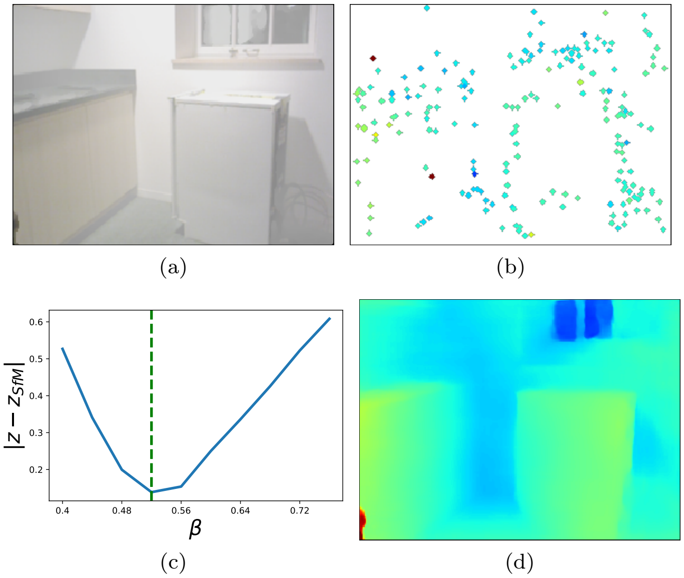


Figure 3.7: Example of parameter search. (a) Input image. (b) Sparse depth map obtained by SfM. (c) Error plot with respect to β. (d) Final output depth.

values:

$$\rho\Big(z_{sfm}(u,v), z_{A,\beta}(u,v)\Big) = \min \left\{ \begin{array}{l} |z_{sfm}(u,v) - z_{A,\beta}(u,v)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u+\delta,v)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u-\delta,v)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u,v+\delta)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u,v-\delta)| \end{array} \right\}. \tag{3.9}$$

As shown in Fig. 3.6, it is uncertain whether the feature point obtained by SfM is located on the background or foreground around depth discontinuities. This includes the possibility that the output depths of the network and SfM are completely different. To suppress the effect of this error on the scattering parameter estimation, we use the neighboring pixels when calculating the residual of the depths. As shown in Eq. (3.9), we use the depth values of the pixels at a distance of δ pixel in the horizontal and vertical direction. The minimum value among these residuals is used for the optimization. Note that we set $\delta = 5$ pixels in this study.

3.3.4 Solver

The network with our dehazing cost volume is differentiable with respect to A and β. Standard gradient-based methods can thus be adopted for the optimization problem. However, we found that an iterative algorithm based on back-propagation easily falls into a local minimum. Therefore, we perform grid

Algorithm 1 Depth and scattering parameter estimation

```
Require: Reference image $I_r$, source images $\{I_s | s \in \{1, \cdots, S\}\}$, depth estimator $\mathcal{F}$, airlight estimator $\mathcal{G}$, $\beta_{min}$, $\beta_{max}$,$\Delta_A$, $\Delta_\beta$, and $\mathbf{z}_{sfm}$
Ensure: $A^*$, $\beta^*$, $\mathbf{z}^*$
  $A_0 \leftarrow \mathcal{G}(I_r)$
  $\beta_0 \leftarrow \underset{\beta \in [\beta_{min}, \beta_{max}]}{\operatorname{argmin}} \sum_{u,v} m(u,v)\rho\Big(z_{sfm}(u,v), z_{A_0,\beta}(u,v)\Big)$
  where $\mathbf{z}_{A,\beta} = \mathcal{F}(A, \beta; I_r, \{I_1, \cdots, I_S\})$
  $A^*, \beta^* \leftarrow \underset{A \in \Omega_A, \beta \in \Omega_\beta}{\operatorname{argmin}} \sum_{u,v} m(u,v)\rho\Big(z_{sfm}(u,v), z_{A,\beta}(u,v)\Big)$
  where $\Omega_A = [A_0 - \Delta_A, A_0 + \Delta_A]$ and $\Omega_\beta = [\beta_0 - \Delta_\beta, \beta_0 + \Delta_\beta]$
  $\mathbf{z}^* \leftarrow \mathcal{F}(A^*, \beta^*; I_r, \{I_1, \cdots, I_S\})$
```

search to find the best solution. Figure 3.7 shows an example in which we search for β under ground-truth A. Figure 3.7(a) shows an input image, and (b) shows the sparse depth map obtained by SfM. The horizontal axis of (c) represents β, and we plot the value of Eq. (3.8) with respect to each β. The green dashed line, which represents the ground-truth β, corresponds to the global minimum. Figure 3.7(d) shows the final output depth of the network with this global optimal solution.

As discussed in Section 3.3.1, we can roughly estimate A with the CNN-based estimator. We initialize A by this estimate. Let A_0 be the output of this estimator, and we search for β_0 in the predetermined search space $[\beta_{min}, \beta_{max}]$ as follows:

$$\beta_0 = \underset{\beta \in [\beta_{min}, \beta_{max}]}{\operatorname{argmin}} \sum_{u,v} m(u,v)\rho\Big(z_{sfm}(u,v), z_{A_0,\beta}(u,v)\Big). \tag{3.10}$$

We then search for A^* and β^* that satisfy Eq. (3.8) in the predetermined search space $[A_0 - \Delta_A, A_0 + \Delta_A]$ and $[\beta_0 - \Delta_\beta, \beta_0 + \Delta_\beta]$. Algorithm 1 shows the overall procedure of depth and scattering parameter estimation.

3.4 Experiments

In this study, we used MVDepthNet [35] as a baseline method. As mentioned previously, the ordinary cost volume is replaced with our dehazing cost volume in the proposed method, so we can directly evaluate the effect of our dehazing cost volume by comparing our method with this baseline method. We also compared the proposed method with simple sequential methods of dehazing and 3D reconstruction using the baseline method. DPSNet [34], the architecture of which is more complicated such as a multi-scale feature extractor, 3D convolutions, and a cost aggregation module, was also trained on hazy images for further comparison. In addition to the experiments with synthetic data, we give an example of applying the proposed method to actual foggy scenes.

3.4.1 Dataset

We used the DeMoN dataset [42] for training. This dataset consists of the SUN3D [43], RGB-D SLAM [44], and MVS datasets [45], which have sequences of real images. The DeMoN dataset also has the Scenes11 dataset [46, 42], which consists of synthetic images. Each image sequence in the DeMoN dataset includes RGB images, depth maps, and camera parameters. In the real-image datasets, most of the depth maps have missing regions due to sensor sensibility. As we discuss later, we synthesized hazy images from the clean images in the DeMoN dataset for training the proposed method, where we need dense depth maps without missing regions to compute pixel-wise degradation due to haze. Therefore, we first trained MVDepthNet using clear images then filled the missing regions of each depth map with the output depth of MVDepthNet. To suppress boundary discontinuities and sensor noise around missing regions, we applied a median filter after depth completion. For the MVS dataset, which has larger noise than other datasets, we reduced the noise simply by thresholding before inpainting. Note that the training loss was computed using only pixels that originally had valid depth values. We generated 419,046 and 8,842 samples for training and test data, respectively. Each sample contained one reference image and one source image. All images were resized to 256×192.

We synthesized a hazy-image dataset for training the proposed method from clear images. The procedure of generating a hazy image is based on Eq. (3.3). For A, we randomly sampled $A \in [0.7, 1.0]$ for each data sample. For β, we randomly sampled $\beta \in [0.4, 0.8], [0.4, 0.8], [0.05, 0.15]$ for the SUN3D, RGB-D SLAM, and Scenes11 datasets, respectively. We found that for the MVS dataset, it was difficult to determine the same sampling range of β for all images because it contains various scenes with different depth scales. Therefore, we determined the sampling range of β for each sample of the MVS dataset as follows. We first set the range of a transmission map $e^{-\beta z}$ to $[0.2, 0.4]$ for all samples then computed the median of a depth map z_{med} for each sample. Finally, we determined the β range for each sample as $\beta \in [-\log(0.4)/z_{med}, -\log(0.2)/z_{med}]$.

Similar to Wang and Shen [35], we adopted data augmentation to enable the network to reconstruct a wide depth range. The depth of each sample was scaled by a factor between 0.5 and 1.5 together with the translation vector of the camera. Note that when training the proposed method, β should also be scaled by the inverse of the scale factor.

3.4.2 Training details

All networks were implemented in PyTorch. The training was done on a NVIDIA V100 GPU with 32-GB memory. The size of a minibatch was 32 for all training.

We first trained MVDepthNet from scratch on the clear-image dataset. We used Adam [47] with a learning rate of 1.0×10^{-4}. After the initial 100K iterations, the learning rate was reduced by 20% after every 20K iterations.

We then fine-tuned MVDepthNet on hazy images and trained the proposed method with our dehazing cost volume. The parameters of both methods were

Table 3.2: Quantitative results. We compared proposed method (MVDepthNet w/ dcv) with MVDepthNet [35] fine-tuned on hazy images (MVDepthNet), simple sequential methods of dehazing [24, 30] and depth estimation with MVDepthNet (AOD-Net + MVDepthNet, FFA-Net + MVDepthNet), and DPSNet [34] trained on hazy images (DPSNet). Red and blue values are best and second-best, respectively.

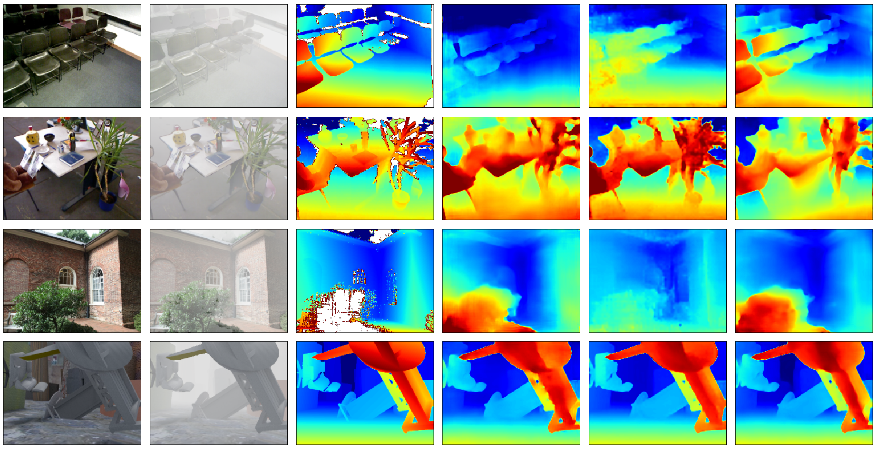
Dataset	Method	L1-rel	L1-inv	sc-inv	C.P. (%)
SUN3D	AOD-Net + MVDepthNet	0.249	0.132	0.250	47.8
	FFA-Net + MVDepthNet	0.180	0.111	0.211	55.5
	MVDepthNet	0.155	0.093	0.184	60.3
	DPSNet	0.145	0.082	0.183	64.7
	MVDepthNet w/ dcv	0.100	0.058	0.161	79.0
RGB-D SLAM	AOD-Net + MVDepthNet	0.205	0.127	0.315	58.9
	FFA-Net + MVDepthNet	0.179	0.114	0.288	65.0
	MVDepthNet	0.157	0.091	0.254	70.7
	DPSNet	0.152	0.090	0.234	71.6
	MVDepthNet w/ dcv	0.162	0.089	0.231	68.8
MVS	AOD-Net + MVDepthNet	0.323	0.123	0.309	51.9
	FFA-Net + MVDepthNet	0.215	0.112	0.288	55.6
	MVDepthNet	0.184	0.100	0.241	57.1
	DPSNet	0.191	0.088	0.239	67.9
	MVDepthNet w/ dcv	0.160	0.091	0.222	58.1
Scenes11	AOD-Net + MVDepthNet	0.330	0.036	0.539	52.3
	FFA-Net + MVDepthNet	0.377	0.041	0.600	51.3
	MVDepthNet	0.151	0.022	0.279	64.0
	DPSNet	0.105	0.018	0.381	81.8
	MVDepthNet w/ dcv	0.134	0.019	0.216	72.3

initialized by that of the trained MVDepthNet on clear images. The initial learning rate was set to 1.0×10^{-4} and reduced by 20% after every 20K iterations.

We also trained the dehazing methods, AOD-Net [24] and FFA-Net [30], and the MVS method DPSNet [34] on our hazy image dataset for comparison. The dehazing networks were followed by MVDepthNet trained on clear images for depth estimation. DPSNet was trained with the same loss function and learning schedule as in the original paper [34].

3.4.3 Evaluation of dehazing cost volume

We first evaluated our dehazing cost volume with ground-truth scattering parameters. Table 3.2 shows the quantitative evaluation. We used four evaluation metrics following Wang and Shen [35]: L1-rel is the mean of the relative L1 error between the ground-truth depth and estimated depth, L1-inv is the mean of the L1 error between ground- truth inverse depth and estimated inverse depth, sc-inv is the scale-invariant error of depth proposed by Eigen et al. [48], and correctly estimated depth percentage (C.P.) [49] is the percentage of pixels whose relative L1 error is within 10%. The red and blue values are the best and



(a) Clear image (b) Hazy input (c) Ground truth (d) MVDepthNet [35] (e) DPSNet [34] (f) Proposed

Figure 3.8: Qualitative results. (a) clear image, (b) hazy input, (c) ground-truth depth, (d) output of fine-tuned MVDepthNet [35], (e) output of DPSNet [34], and (f) output of proposed method. From top to bottom, each row shows results of input images in SUN3D, RGB-D SLAM, MVS, and Scenes11 datasets, respectively.

second-best, respectively.

The proposed method (MVDepthMet w/ dcv, where "dcv" denotes our dehazing cost volume) was compared with MVDepthNet [35] fine-tuned on hazy images (MVDepthNet), simple sequential methods of dehazing [24, 30] and depth estimation with MVDepthNet [35] (AOD-Net + MVDepthNet, FFA-Net + MVDepthNet), and DPSNet [34] trained on hazy images (DPSNet).

In most evaluation metrics, the proposed method outperformed the fine-tuned MVDepthNet, demonstrating the effectiveness of our dehazing cost volume. For the RGB-D SLAM dataset, the fine-tuned MVDepthNet was comparable to the proposed method. This is because many scenes in the RGB-D SLAM dataset are close to a camera. In such case, the degradation of an observed image is small and exists uniformly in the image, which has little effect on photometric consistency.

The proposed method also performed better than the sequential methods of dehazing [24, 30] and MVDepthNet [35]. Therefore, we can see that the simultaneous modeling of dehazing and 3D reconstruction on the basis of our dehazing cost volume is effective. DPSNet [34] first extracts feature maps from input images, and then constructs a cost volume in the feature space. Thus, the feature extractor might be able to deal with image degradation caused by light scattering. Nevertheless, our dehazing cost volume allows the consideration of image degradation with a simple network architecture.

The output depth of each method is shown in Fig. 3.8. From top to bot-

Table 3.3: Quantitative results of depth and scattering parameter estimation. "MVDepthNet w/ dcv, pe" denotes the proposed method with scattering parameter estimation. Red and blue values are best and second-best, respectively. As evaluation metric of A and β, we used mean absolute error (MAE$_A$ and MAE$_\beta$).

Dataset	Method	L1-rel	L1-inv	sc-inv	C.P. (%)	MAE$_A$	MAE$_\beta$
L1-rel ≤ 0.1 1364 samples	FFA-Net + MVDepthNet	0.141	0.104	0.152	57.0	-	-
	MVDepthNet	0.130	0.090	0.135	59.9	-	-
	DPSNet	0.109	0.069	0.125	65.2	-	-
	MVDepthNet w/ dcv	0.069	0.043	0.104	80.7	-	-
	MVDepthNet w/ dcv, pe	0.081	0.050	0.116	76.3	0.028	0.043
L1-rel ≤ 0.2 2661 samples	FFA-Net + MVDepthNet	0.154	0.102	0.172	52.4	-	-
	MVDepthNet	0.138	0.088	0.152	56.0	-	-
	DPSNet	0.120	0.072	0.138	61.1	-	-
	MVDepthNet w/ dcv	0.077	0.044	0.116	78.4	-	-
	MVDepthNet w/ dcv, pe	0.092	0.053	0.132	72.9	0.028	0.042
L1-rel ≤ 0.3 3157 samples	FFA-Net + MVDepthNet	0.162	0.103	0.182	50.7	-	-
	MVDepthNet	0.143	0.089	0.158	54.7	-	-
	DPSNet	0.124	0.072	0.144	59.9	-	-
	MVDepthNet w/ dcv	0.079	0.045	0.120	77.6	-	-
	MVDepthNet w/ dcv, pe	0.100	0.056	0.141	70.3	0.027	0.044

tom, each row shows the results of the input images in the SUN3D, RGB-D SLAM, MVS, and Scenes11 datasets, respectively. DPSNet failed to construct correspondence in some scenes, although it has the multi-scale feature extractor. Note that the results from the Scenes11 dataset indicate that the proposed method can reconstruct the 3D geometry of a distant scene where the image is heavily degraded due to scattering media.

3.4.4 Evaluation of scattering parameter estimation

Next, we evaluated the proposed method with scattering parameter estimation. Each sample of the test dataset presented above consists of image pairs. Parameter estimation requires a 3D point cloud obtained by SfM. To ensure the accuracy of SfM, which requires high visual overlap between images and a sufficient number of images observing the same objects, we created a new test dataset for the evaluation of the scattering parameter estimation. From the SUN3D dataset [43], we selected 68 scenes and extracted 80 frames from each scene. The resolution of each image is 680×480. We cropped the image patch with 512×384 from the center and downsized the resolution to 256×192 for the input of the proposed method. Similar to the previous test dataset, missing regions were compensated with the output of MVDepthNet [35]. The scattering parameters were randomly sampled for each scene, where the sampling ranges were $A \in [0.7, 1.0]$ and $\beta \in [0.4, 0.8]$. SfM [38, 3] was applied to all 80 frames of each scene to estimate a sparse 3D point cloud, and then the proposed method took the image pair as input. To evaluate the output depth on the ground-truth depth of the original SUN3D dataset, the sparse depth obtained by SfM was

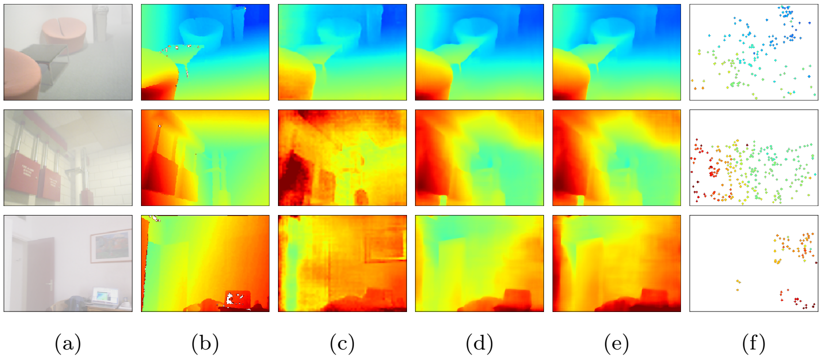


Figure 3.9: Output depth after scattering parameter estimation. (a) Hazy input, (b) ground-truth depth, (c) DPSNet [34], (d) proposed method with ground-truth scattering parameters, (e) proposed method with scattering parameter estimation, and (f) sparse depth obtained by SfM.

rescaled to match the scale of the ground-truth depth, and we used the camera parameters of the original SUN3D dataset.

For the parameter search, we set the first β range as $\beta_{min} = 0.4$ and $\beta_{max} = 0.8$ with 10 steps for the grid search. We then searched for A and β with the search range $\Delta_A = 0.05$, $\Delta_\beta = 0.05$ and 4×4 steps. The total number of the forward computation of the network was 26, and the total computation time was about 15 seconds in our computational environment.

Table 3.3 shows the quantitative results of depth and scattering parameter estimation. "MVDepthNet w/ dcv, pe" denotes the proposed method with scattering parameter estimation. As the evaluation metric of A and β, we used mean absolute error (MAE$_A$ and MAE$_\beta$). To evaluate the effect of the error at the SfM step, we created three test datasets, where the relative L1 error of the sparse SfM depth of the samples is less than 0.1, 0.2, and 0.3, respectively, and show the number of samples in the table. These results indicate that the proposed method with ground-truth scattering parameters (MVDeptNet w/ dcv) performed the best. On the other hand, even when we incorporated scattering parameter estimation into the proposed method, it outperformed the other methods. In addition, scattering parameter estimation is robust to the estimation error of the sparse depth at the SfM step since the MAE values for A and β did not vary so much for the three datasets with different SfM errors.

The qualitative results of the following depth estimation after scattering parameter estimation are shown in Fig. 3.9. Figure 3.9(f) shows the input sparse depth obtained by SfM. Compared with the proposed method with ground-truth scattering parameters, the method with the scattering parameter estimation resulted in almost the same output depth. In the third row in the figure, the left part in the image has slight error because no 3D sparse points were observed in that region.

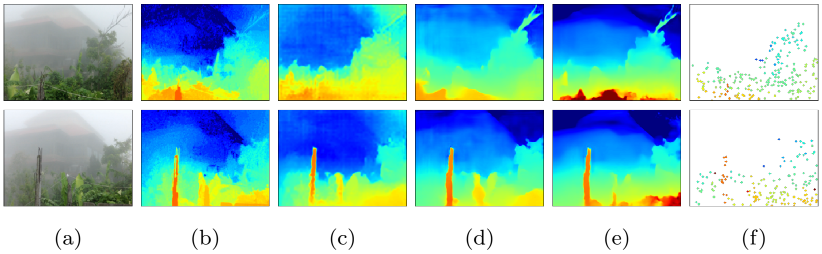


Figure 3.10: Experimental results on *bali* [41]. (a) foggy input, (b) estimated depth of Li et al. [41], (c) output of DPSNet [34], (d) output of fine-tuned MVDepthNet [35], (e) output of proposed method with scattering parameter estimation, and (f) sparse depth obtained by SfM.

3.4.5 Experiments with actual foggy scenes

Finally, we give an example of applying the proposed method to actual outdoor foggy scenes. We used the image sequence *bali* [41] for the actual data. This data consists of about 200 frames, and we applied the SfM method [38, 3] to all these frames to obtain camera parameters and a sparse 3D point cloud. The proposed method took the estimated camera parameters, a sparse depth, and image pair as input. We set the search space of the scattering parameter estimation as $\beta_{min} = 0.01$, $\beta_{max} = 0.1$, $\Delta_A = 0.05$, and $\Delta_\beta = 0.01$ with the same step size in the experiments of the synthesized data.

The results are shown in Fig. 3.10. The output depths of the proposed method were rescaled to match the scale of the output of [41], because the camera parameters were different between these methods. Compared with [41], the proposed method can reconstruct distant region, which have large image degradation due to light scattering, and the other learning-based methods also failed to reconstruct such distant regions. Moreover, the proposed method could recover less noisy depth maps as a trade-off for loss of small details due to over-smoothing. The method proposed by Li et al. [41] requires iterative graph-cut optimization, so it takes a few minutes to estimate depth for one image. Our method, on the other hand, requires only a few seconds to estimate depth for one reference image after estimating scattering parameters. Although scattering parameter estimation takes several ten of seconds, if we assume the medium density of a scene is homogeneous, the estimated scattering parameters at a certain frame can be used for another frame without additional parameter estimation.

We also captured a video with a smartphone camera in an actual foggy scene. Similar to the previous experiments, we applied the SfM method [38, 3] to all frames. The proposed method took the estimated camera parameters, a sparse depth, and image pair as input, and the parameters search space was set as the same in the previous experiments.

The results are shown in Fig. 3.11. Figures (a) and (b) show the input reference and source images, respectively. This results also indicate that the proposed method can reconstruct distant regions with large image degradation

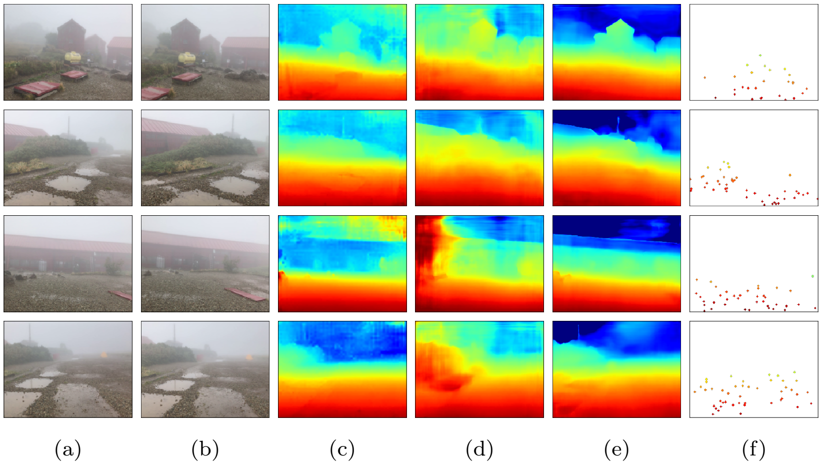


Figure 3.11: Experimental results on our captured data in actual foggy scenes. (a) input reference image, (b) input source image, (c) output of DPSNet [34], (d) output of fine-tuned MVDepthNet [35], (e) output of proposed method with scattering parameter estimation, and (f) sparse depth obtained by SfM.

due to light scattering.

3.5 Conclusion

In this chapter, we discussed a disparity-based 3D reconstruction method in scattering media. We proposed a learning-based MVS method with a novel cost volume, called the dehazing cost volume, which enables MVS methods to be used in scattering media. Differing from the ordinary cost volume, our dehazing cost volume can compute the cost of photometric consistency by taking into account image degradation modeled by the atmospheric scattering model. This is the first study to solve the chicken-and-egg problem of depth and scattering estimation by computing the scattering effect using each swept plane in the cost volume without explicit scene depth. We also proposed a method for estimating scattering parameters such as airlight and a scattering coefficient. This method leverages geometric information obtained at an SfM step, and ensures the correctness of the following depth estimation. The experimental results on synthesized hazy images indicate the effectiveness of our dehazing cost volume in scattering media. We also demonstrated its applicability using images captured in actual foggy scenes.

Chapter 4

Photometric Stereo in Scattering Media

In this chapter, we discuss a shading-based 3D reconstruction method in scattering media. Shading-based methods, such as shape-from-shading [5] and photometric stereo [6], directly use the pixel intensity of input images. These methods are thus affected by light scattering and attenuation in scattering media. We propose a photometric stereo method in scattering media as shown in Figure 4.1.

Several shading-based 3D reconstruction methods in scattering media have been proposed. Photometric stereo methods are an effective approach for reconstructing a 3D shape in scattering media [50, 15, 51]. They reconstruct surface normals from images captured under different lighting conditions. As described in Chapter 2, the single scattering model can be used for modeling light scattering under active light sources. Figure 4.2 shows a capture setting with a single camera and light source under the single scattering model. As shown, backscatter and forward scatter occur in scattering media; thus, the irradiance observed at a camera includes a direct component reflected on the surface, as well as a backscatter and forward scatter components. Narasimhan et al. [50] modeled single backscattering under a directional light source in scattering media and estimated surface normals using a nonlinear optimization technique. Tsiotsios et al. [15] assumed that backscatter saturates close to the camera when illumination follows the inverse square law, and subtracted the backscatter from the captured image. Note that forward scatter is not modeled in these methods. Forward scatter depends on the object's shape locally and globally, and in highly turbid media such as port water, 3D reconstruction accuracy is affected by forward scatter. Although Murez et al. [51] proposed a photometric stereo technique that considers forward scatter, they assumed that the scene is approximated as a plane, which enables prior calibration of forward scatter. Therefore, this assumption deteriorates the estimation of normals because forward scatter is intrinsically dependent on the object's shape.

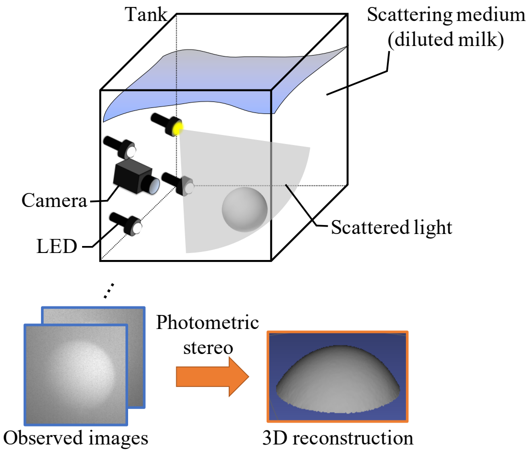


Figure 4.1: Photometric stereo in scattering media

We propose a forward scatter model and implement the model into a photometric stereo framework. Differing from previous studies [52, 51], we compute forward scatter, which depends on the object's shape. To overcome the mutual dependence between shape and forward scatter, we develop an iterative algorithm that performs a forward scatter removal and 3D shape reconstruction alternately.

As mentioned in Chapter 2, the single scattering model is more complicated model than the atmospheric scattering model. We thus also propose an effective method for computing forward scatter with an analytical form of single scattering. In computer graphics, Monte Carlo and finite element techniques have been used to simulate light scattering in scattering media. Although such techniques provide accurate simulations, realtime rendering is difficult. Thus, analytical or closed-form solutions have been proposed for efficient computation [11, 53, 54]. For example, Sun et al. [11] proposed an analytical single scattering model of backscatter and forward scatter between the source and the surface (source-surface forward scatter) using 2D lookup tables. Similar to their model, in this study, forward scatter between the surface and the camera (surface-camera forward scatter) is computed using a lookup table.

The rest of this chapter is organized as follows: In Section 4.1, we describe the theory of photometric stereo including surface normal integration to reconstruct a 3D shape. In Section 4.2, analytical solution of the single scattering model by using lookup tables is introduced. In Section 4.3, an efficient method for removing forward scatter components is described. Shape-dependent forward scatter is modeled as spatially-variant kernels. To address computational complexity issues, we approximate the kernel matrix as a sparse matrix. In Section 4.4, we discuss the approximation in the proposed method using synthesized data, then demonstrate the effectiveness of the proposed method with

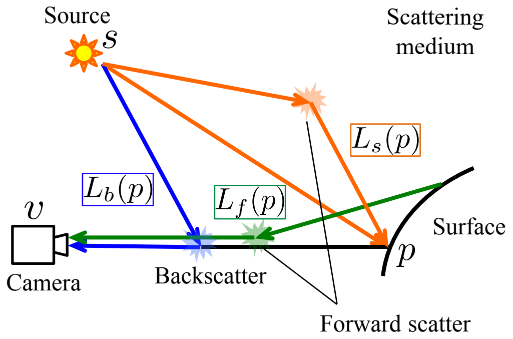


Figure 4.2: Single scattering model. Observed irradiance at camera includes direct component reflected on surface, and both backscatter and forward scatter components.

real data. Finally, Section 4.5 concludes this chapter.

4.1 Photometric stereo

4.1.1 Theory of photometric stereo

In this section, we explain the theory of photometric stereo, which is related to radiometry. The details also can be seen in [6, 55].

Here a surface is illuminated by a point light source as shown in Figure 4.3. In radiometry, the intensity of the light source is represented as a radiant flux Φ [W]. Let the distance between the light source and surface be d. Light arrived at the surface is described by irradiance E [W/m^2] as follows:

$$E = \frac{I_0}{d^2}\mathbf{n}^\top\mathbf{l}, \tag{4.1}$$

where I_0 [W/sr] is a radiant intensity, $\mathbf{n}$ is a surface normal, and $\mathbf{l}$ is a lighting direction. If the light source is isotropic, I_0 is given as

$$I_0 = \frac{\Phi}{4\pi}. \tag{4.2}$$

Now, we assume that the light source is infinitely distant from the surface. In this case, radiance at the surface $L_0 = I_0/d^2$ [W/m^2 $\cdot$ sr] is constant and the lighting direction $\mathbf{l}$ is all the same in the scene.

The radiance of light that bounces off the surface then arrives at the camera is described as follows:

$$L = f(\mathbf{n}, \mathbf{l}, \mathbf{v})L_0\mathbf{n}^\top\mathbf{l}, \tag{4.3}$$

where $\mathbf{v}$ is a viewing direction and $f(\mathbf{n}, \mathbf{l}, \mathbf{v})$ is a bidirectional reflectance distribution function (BRDF). BRDF takes a surface normal and the directions of

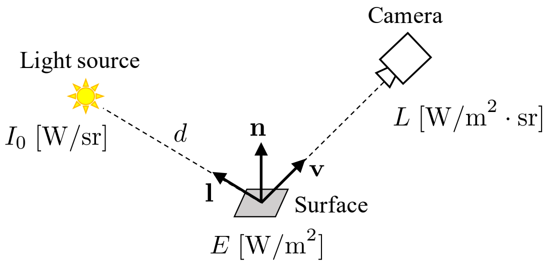


Figure 4.3: Camera, light source, and surface

incident and reflected light for modeling the distribution of reflected light. For Lambertian surfaces, BRDF becomes isotoropic:

$$L = \rho L_0 \mathbf{n}^\top \mathbf{l}, \tag{4.4}$$

where ρ is called a diffuse albedo. This is the basics of photometric stereo and the goal of photometric stereo is to estimate the surface normal $\mathbf{n}$ from the observed irradiance L and known lighting direction $\mathbf{l}$.

We assume that three images are captured under different lighting conditions and each lighting direction is calibrated beforehand (These lighting directions must be linearly independent). For the sake of brevity, each radiance of the light source is normalized as $L_0 = 1$. The following equation can be obtained from Eq. (4.4):

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \rho \begin{bmatrix} \mathbf{l}_1^\top \\ \mathbf{l}_2^\top \\ \mathbf{l}_3^\top \end{bmatrix} \mathbf{n}. \tag{4.5}$$

We can then compute the surface normal $\mathbf{n}$ as follows:

$$\tilde{\mathbf{n}} = \rho \mathbf{n} = \begin{bmatrix} \mathbf{l}_1^\top \\ \mathbf{l}_2^\top \\ \mathbf{l}_3^\top \end{bmatrix}^{-1} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix}, \tag{4.6}$$

$$\mathbf{n} = \frac{\tilde{\mathbf{n}}}{\|\tilde{\mathbf{n}}\|}, \quad \rho = \|\tilde{\mathbf{n}}\|. \tag{4.7}$$

If we have more light sources, the surface normal is computed in the least squares sense.

Traditional photometric stereo assumes Lambertian surfaces and distant light sources whose directions are known as described above. Thus, recent works focus on uncalibrated and near lighting photometric stereo [56, 57], arbitrary BRDF [58, 59], and robustness to outliers like shadows or specular reflection [60, 61]. In addition, several works deal with global effects (e.g., ambient light [62], interreflection [63, 64, 65], or subsurface scattering [66, 67]), and our study is related to these works.

4.1.2 Shape reconstruction from surface normals

The output of photometric stereo is surface normals, which requires the integral of surface normals for reconstructing a 3D shape. In this section, we first explain the integral of surface normals under orthogonal projection formulated as Poisson's equation [68]. Then, it is extended to the case of perspective projection [69].

First of all, we rewrite a surface normal $\mathbf{n}$ as follows:

$$p = \frac{\partial Z}{\partial x}, \quad q = \frac{\partial Z}{\partial y}, \tag{4.8}$$

$$\mathbf{n} = [p, q, -1]^{\top} / \sqrt{p^2 + q^2 + 1}, \tag{4.9}$$

where (x, y) is the coordinates of 3D space, which are aligned with the image axes in the case of orthogonal projection. The output normal map of photometric stereo is thus represented as a vector field $N : \mathbb{R}^2 \to \mathbb{R}^2, (x, y) \mapsto (p(x, y), q(x, y))$. We aim to reconstruct a depth map $Z : \mathbb{R}^2 \to \mathbb{R}$ from this vector field. Here, let $\nabla Z : \mathbb{R}^2 \to \mathbb{R}^2, (x, y) \mapsto (Z_x(x, y), Z_y(x, y))$ be the gradient field of Z. The depth map is obtained by solving the functional minimization problem as follows:

$$Z^* = \underset{Z}{\operatorname{argmin}} \int\int \left((Z_x(x, y) - p(x, y))^2 + (Z_y(x, y) - q(x, y))^2\right) dxdy. \tag{4.10}$$

The Euler-Lagrange equation yields

$$\nabla^2 Z = \operatorname{div}(p, q). \tag{4.11}$$

This solution is called Poisson Solver [68].

To extend this solution to perspective projection, we first introduce the intrinsic parameters of the camera as follows:

$$\begin{bmatrix} f_u & \gamma & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.12}$$

where f_u and f_v represent focal lengths, c_u and c_v represent the coordinates of the principal point, and γ describes the skew of the two image axes. Here, the image coordinate is defined as (u, v), and a depth map and its gradient field are represented as $Z : \mathbb{R}^2 \to \mathbb{R}, (u, v) \mapsto Z(u, v)$ and $\nabla Z : \mathbb{R}^2 \to \mathbb{R}^2, (u, v) \mapsto (Z_u(u, v), Z_v(u, v))$, respectively. $Z_x(u, v)$ and $Z_y(u, v)$ are computed as follows:

$$\begin{aligned} Z_x(u, v) &= \frac{\partial Z}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial Z}{\partial v}\frac{\partial v}{\partial x} \\ &= \frac{Z_u(u, v) f_u}{Z(u, v) + Z_u(u, v)(u - c_u) + Z_v(u, v)(v - c_v)}, \end{aligned} \tag{4.13}$$

$$\begin{aligned} Z_y(u, v) &= \frac{\partial Z}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial Z}{\partial v}\frac{\partial v}{\partial y} \\ &= \frac{Z_u(u, v)\gamma + Z_v(u, v) f_v}{Z(u, v) + Z_u(u, v)(u - c_u) + Z_v(u, v)(v - c_v)}. \end{aligned} \tag{4.14}$$

The depth map reconstruction under perspective projection is formulated as follows:

$$g(u,v) = \log\left(Z(u,v)\right), \tag{4.15}$$

$$g_u(u,v) = \frac{\partial g}{\partial u} = \frac{Z_u(u,v)}{Z(u,v)}, \;\; g_v(u,v) = \frac{\partial g}{\partial v} = \frac{Z_v(u,v)}{Z(u,v)}, \tag{4.16}$$

$$g^* = \operatorname*{argmin}_g \int\int \left((g_u(u,v) - \tilde{g}_u(u,v))^2 + (g_v(u,v) - \tilde{g}_v(u,v))^2\right) dudv. \tag{4.17}$$

The Euler-Lagrange equation yields

$$\nabla^2 g = \mathrm{div}(\tilde{g}_u, \tilde{g}_v), \tag{4.18}$$

where $\tilde{g}_u$ and $\tilde{g}_v$ are computed with the output surface normal defined as

$$\mathbf{n}(u,v) = [n(u,v,x), n(u,v,y), n(u,v,z)]^\top. \tag{4.19}$$

To compute $\tilde{g}_u$ and $\tilde{g}_v$, $r_1(u,v)$ and $r_2(u,v)$ are introduced as follows:

$$r_1(u,v) = \frac{n(u,v,x)}{n(u,v,z)} = -Z_x(u,v), \tag{4.20}$$

$$r_2(u,v) = \frac{n(u,v,y)}{n(u,v,z)} = -Z_y(u,v). \tag{4.21}$$

From Eqs. (4.13) and (4.14), we can obtain

$$r_1(u,v)Z(u,v) + (r_1(u,v)\bar{u} + f_x)Z_u(u,v) + r_1(u,v)Z_v(u,v)\bar{v} = 0, \tag{4.22}$$

$$r_2(u,v)Z(u,v)) + (r_2(u,v)\bar{u} + \gamma)Z_u(u,v) + (r_2(u,v)\bar{v} + f_y)Z_v(u,v) = 0, \tag{4.23}$$

where $\bar{u} = u - c_x$ and $\bar{v} = v - c_y$. From Eqs (4.16), (4.22), and (4.23), we can obtain $\tilde{g}_u$ and $\tilde{g}_v$ as follows:

$$\tilde{g}_u(u,v) = \frac{r_1(u,v)f_y}{r_1(u,v)\gamma\bar{v} - r_1(u,v)f_y\bar{u} - r_2(u,v)f_x\bar{v} - f_xf_y}, \tag{4.24}$$

$$\tilde{g}_v(u,v) = \frac{r_2(u,v)f_x - r_1(u,v)\gamma}{r_1(u,v)\gamma\bar{v} - r_1(u,v)f_y\bar{u} - r_2(u,v)f_x\bar{v} - f_xf_y}. \tag{4.25}$$

Note that in $\mathrm{div}(\tilde{g}_u, \tilde{g}_v) = \partial\tilde{g}_u/\partial u + \partial\tilde{g}_v/\partial v$, derivatives $\partial\tilde{g}_u/\partial u$ and $\partial\tilde{g}_v/\partial v$ can be computed on image space as

$$\frac{\partial\tilde{g}_u}{\partial u} = \tilde{g}_u(u,v) - \tilde{g}_u(u-1,v), \tag{4.26}$$

$$\frac{\partial\tilde{g}_v}{\partial v} = \tilde{g}_v(u,v) - \tilde{g}_v(u,v-1). \tag{4.27}$$

4.2 Analytical form of single scattering model

First of all, we provide an analytical form of the single scattering model using lookup tables. In computer graphics, analytical or closed-form solutions for single scattering in scattering media have been proposed to overcome computational complexity issues. Sun et al. [11] assumed single and isotropic scattering and used 2D lookup tables to analytically describe backscatter and source-camera forward scatter. Zhou et al. [53] extended this approach to inhomogeneous single scattering media with respect to backscatter. Pegoraro et al. [54] derived a closed-form solution for single backscattering under a general phase function and light distribution. In this study, owing to its simplicity, we use a lookup table similar to that of Sun et al. [11], and we model surface-camera forward scatter analytically. Note that we assume perspective projection, near lighting, and Lambertian objects.

Here, let $L(p)$ be irradiance at a camera when the 3D position p on an object surface is observed. As shown in Fig. 4.2, $L(p)$ is decomposed into a reflected component $L_s(p)$ (orange arrow), a backscatter component $L_b(p)$ (blue arrow), and a surface-camera forward scatter component $L_f(p)$ (green arrow) as follows:

$$L(p) = L_s(p)e^{-\sigma d_{vp}} + L_b(p) + L_f(p). \tag{4.28}$$

Here, parameters σ and d_{vp} denote an extinction coefficient and the distance between the camera and position p, respectively. In scattering media, light is attenuated exponentially relative to distance. The extinction coefficient σ is the sum of the absorption coefficient α and the scattering coefficient β as described in Eq. (2.3).

As shown in Fig. 4.4, the reflected component $L_s(p)$ consists of a direct component $L_{s,d}(p)$ (yellow arrow) and a source-surface forward scatter component $L_{s,f}(p)$ (red arrow),

$$L_s(p) = L_{s,d}(p) + L_{s,f}(p). \tag{4.29}$$

Thus, the observed irradiance is written as follows:

$$L_s(p) = \left(L_{s,d}(p) + L_{s,f}(p)\right)e^{-\sigma d_{vp}} + L_b(p) + L_f(p). \tag{4.30}$$

In the rest of this section, we describe these four components.

4.2.1 Direct component

The direct component reaches the surface directly from the source as shown in Fig. 4.4. Considering diffuse reflection and attenuation in scattering media, $L_{s,d}(p)$ is expressed as follows:

$$L_{s,d}(p) = \frac{I_0}{d_{sp}^2}e^{-T_{sp}}\rho_p \mathbf{n}_p^\top \mathbf{l}_{sp}, \tag{4.31}$$

where ρ_p is a diffuse albedo at p, $\mathbf{n}_p$ is a surface normal, and $\mathbf{l}_{sp}$ is the direction from p to the source. $T_{sp} = \sigma d_{sp}$ is optical thickness. In the following, T_{xy}

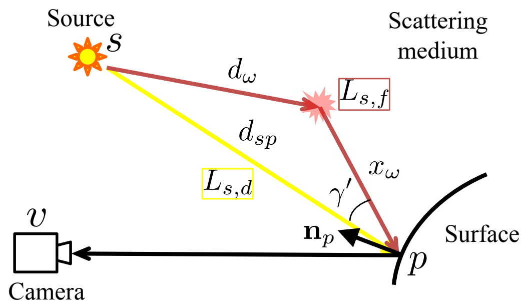


Figure 4.4: Reflected component $L_s(p)$ (yellow arrow) consists of direct component $L_{s,d}(p)$ (red arrow) and source-surface forward scatter component $L_{s,f}(p)$. Direct component reaches surface directly from light source. Source-surface forward scatter is reflected component whose incident light reaches surface via forward scatter.

denotes the product of σ and distance d_{xy}. Without the attenuation $e^{-T_{sp}}$, this is the same as the formulation of traditional photometric stereo as described in Section 4.1.

4.2.2 Backscatter component

Figure 4.5 shows the observation of the backscatter component. As described in Section 2.2, the backscatter component is the sum of scattered light on the viewline without reaching the surface. We rewrite Eq. (2.13) as follows:

$$L_b(p) = \int_0^{d_{vp}} \frac{I_0}{d^2}\beta P(\theta) e^{-\sigma(x+d)} dx, \tag{4.32}$$

where d is the distance between the source and a scattering point, x is the distance between the scattering point and camera, I_0 denotes the radiant intensity of the source, θ is a scattering angle, and $P(\alpha)$ is a phase function that describes the angular scattering distribution. Although Eq. (4.32) cannot be computed in closed-form, an analytical solution can be acquired using a lookup table. However, variables related to the integration in Eq. (4.32) are d_{vp}, d_{sv}, γ, and σ (d_{sv} is a distance between the source and camera, and γ is an angle between the light source and viewing ray. From d_{vp}, d_{sv}, and γ, we can describe complete geometry among the source, surface, and camera); thus, the entry of the table is four-dimensional. Sun et al. [11] assumed isotropic scattering (i.e., $P(\theta) = 1/4\pi$) and derived an analytical solution using a 2D lookup table $F(u, v)$:

$$L_b(p) = I_0 H_0(T_{sv}, \gamma)\left[F(H_1(T_{sv}, \gamma), H_2(T_{vp}, T_{sv}, \gamma)) - F(H_1(T_{sv}, \gamma), \frac{\gamma}{2})\right], \tag{4.33}$$

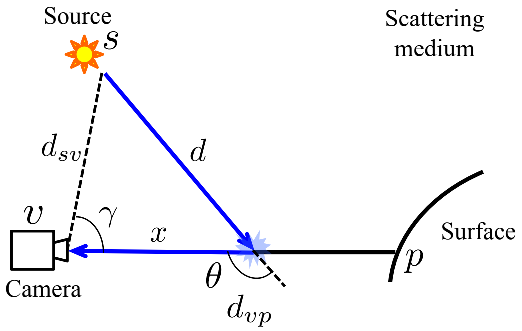


Figure 4.5: Backscatter component is sum of scattered light on viewline without reaching surface.

where $H_0(T_{sv},\gamma)$, $H_1(T_{sv},\gamma)$, and $H_2(T_{vp},T_{sv},\gamma)$ are defined as follows:

$$H_0(T_{sv},\gamma) = \frac{\beta\sigma e^{-T_{sv}\cos\gamma}}{2\pi T_{sv}\sin\gamma}, \tag{4.34}$$

$$H_1(T_{sv},\gamma) = T_{sv}\sin\gamma, \tag{4.35}$$

$$H_2(T_{vp},T_{sv},\gamma) = \frac{\pi}{4}+\frac{1}{2}\arctan\frac{T_{vp}-T_{sv}\cos\gamma}{T_{sv}\sin\gamma}. \tag{4.36}$$

$F(u,v)=\int_0^v e^{-u\tan\xi}d\xi$ is a 2D lookup table computed numerically in advance.

4.2.3 Source-surface forward scatter component

The source-surface forward scatter is a reflected component whose incident light reaches the surface via forward scatter (see Fig. 4.4). This component is the integral of scattered light on a hemisphere centered on p:

$$L_{s,f}(p) = \int_{\Omega_{2\pi}} L_b(\omega)\rho_p \mathbf{n}_p^\top \mathbf{l}_\omega d\omega, \tag{4.37}$$

where $\mathbf{l}_\omega$ is a incident direction. We define $L_b(\omega)$ as the sum of scattered light from direction $\mathbf{l}_\omega$:

$$L_b(\omega) = \int_0^\infty \frac{I_0}{d_\omega^2}\beta P(\theta)e^{-\sigma(x_\omega+d_\omega)}dx_\omega, \tag{4.38}$$

where d_ω is the distance between the source and a scattering point and x_ω is the distance between the scattering point and surface. As discussed in Section 4.2.2, Sun et al. [11] derived an analytical solution using a 2D lookup table as follows:

$$L_{s,f}(p) = \frac{\beta\sigma I_0\rho_p}{2\pi T_{sp}}G(T_{sp},\mathbf{n}_p^\top\mathbf{l}_{sp}), \tag{4.39}$$

where $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$ is a 2D lookup table given as

$$G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) = \int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma'}}{\sin\gamma'} \left[F(H_1(T_{sp}, \gamma'), \frac{\pi}{2}) - F(H_1(T_{sp}, \gamma'), \frac{\gamma'}{2}) \right] \mathbf{n}_p^\top \mathbf{l}_\omega d\omega, \tag{4.40}$$

where γ' is an angle between the light source and the incident direction.

4.2.4 Surface-camera forward scatter component

When we observe surface point p in scattering media, the light reflected on point q is scattered on the viewline, and the scattered light is also observed as a forward scatter component (see Fig. 4.6). In this study, we describe this component analytically using a lookup table.

As shown in Fig. 4.6, irradiance at the camera includes reflected light from the small facet centered at q. If we consider this small facet as a virtual light source, similar to Eq. (4.32), the irradiance can be expressed as follows:

$$\int_0^{d_{vp'}} \frac{L_s(q)dA_q}{d^2} \beta P(\theta) e^{-\sigma(x+d)} dx, \tag{4.41}$$

where dA_q is the area of the facet. At the camera, a discrete point on the surface corresponding to the pixel is observed. Thus, $L_f(p)$ is the sum of these discrete points:

$$L_f(p) = \sum_{q \neq p} \int_0^{d_{vp'}} \frac{L_s(q)dA_q}{d^2} \beta P(\theta) e^{-\sigma(x+d)} dx. \tag{4.42}$$

Note that the domain of integration $[0, d_{vp'}]$ differs from that of Eq. (4.32), i.e., $[0, d_{vp}]$. We define p' as the intersection point of the viewline and the tangent plane to q. If $d_{vp'} > d_{vp}$, i.e., p' is inside the object, we set $d_{vp'} = d_{vp}$. If $d_{vp'} < 0$ which means that p' is behind the camera, we set $d_{vp'} = 0$. Similar to Eq. (4.33), the isotropic scattering assumption yields the following:

$$L_f(p) = \sum_{q \neq p} L_s(q) dA_q H_0(T_{vq}, \gamma) \left[F(H_1(T_{vq}, \gamma), H_2(T_{vp'}, T_{vq}, \gamma)) - F(H_1(T_{vq}, \gamma), \frac{\gamma}{2}) \right]. \tag{4.43}$$

This is the analytical expression of the surface-camera forward scatter. Note that we define the area of the small facet as follows [63]:

$$dA_q = \frac{dI}{\mathbf{v_q}^\top \mathbf{n}_q}, \tag{4.44}$$

where dI is the area of the camera pixel and $\mathbf{v_q}$ is the direction from q to the camera.

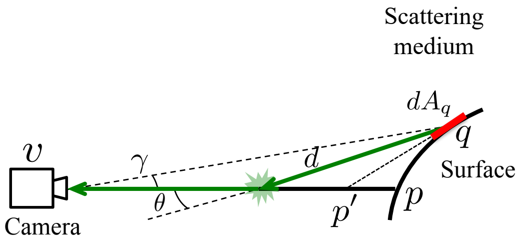


Figure 4.6: Surface-camera forward scatter component. When we observe surface point p in scattering media, light reflected on point q is scattered on viewline, and scattered light is also observed.

4.3 Photometric stereo considering shape-dependent forward scatter

In Section 4.2, we model the image formation in scattering media using four components in Eq. (4.30). To reconstruct surface normals using photometric stereo, we must restore the direct component $L_{s,d}(p)$. Previous photometric stereo methods that only model backscatter [50, 15] consider the direct component $L_{s,d}(p)$ and the backscatter component $L_b(p)$. Besides these components, we deal with both the surface-camera forward scatter $L_f(p)$ and the source-surface forward scatter $L_{s,f}(p)$.

In this section, we first explain the compensation of the backscatter component [15]. Then, we discuss how to remove the surface-camera forward scatter. Finally, we explain photometric stereo that considers the source-surface forward scatter.

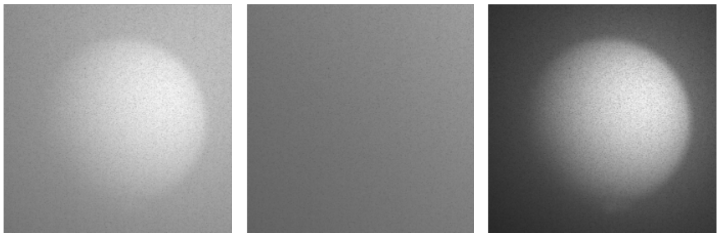
4.3.1 Backscatter removal

As mentioned previously, to remove backscatter, Tsiotsios et al. [15] leveraged backscatter saturation without computing it explicitly, i.e., subtracting no object image from an input image. We also use an image without the target object to remove the backscatter component $L_b(p)$ from the input image. Figure 4.7 shows the example of the backscatter removal.

4.3.2 Approximation of a large-scale dense matrix

Here, let $\mathbf{L}' = \left[L(p^1) - L_b(p^1), \cdots, L(p^N) - L_b(p^N)\right]^\top \in \mathbb{R}^N$ be a backscatter removed image, where N is the number of pixels. Then from Eq. (4.28) and (4.43), reflected light at the surface $\mathbf{L}_s = \left[L_s(p^1), \cdots, L_s(p^N)\right]^\top \in \mathbb{R}^N$ is expressed as follows:

$$\mathbf{L}' = \mathbf{K}\mathbf{L}_s, \tag{4.45}$$



(a) Observed image (b) No object image (c) Subtracted

Figure 4.7: Example of the backscatter removal

where $\mathbf{K} \in \mathbb{R}^{N \times N}$ is a large-scale dense matrix. Each element K_{pq} is given by

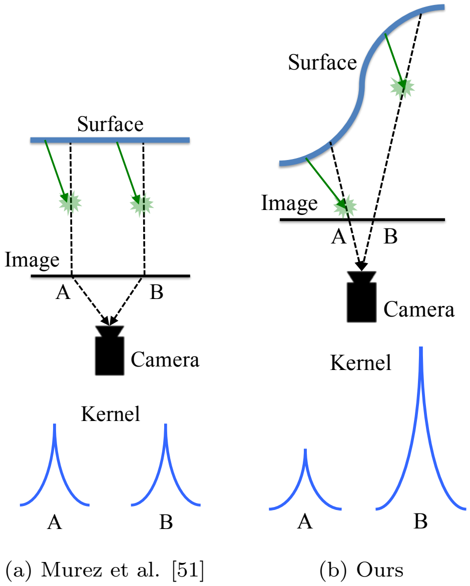
$$K_{pq} = \begin{cases} e^{-T_{vp}} \qquad (p = q) \\ dA_q H_0(T_{vq}, \gamma) \left[F(H_1(T_{vq}, \gamma), H_2(T_{vp'}, T_{vq}, \gamma)) \right. \\ \left. -F(H_1(T_{vq}, \gamma), \frac{\gamma}{2}) \right] \qquad (p \neq q). \end{cases} \tag{4.46}$$

Theoretically, the reflected light is recovered using an inverse matrix $\mathbf{K}^{-1}$.

$$\mathbf{L}_s = \mathbf{K}^{-1}\mathbf{L}' \tag{4.47}$$

Our model is similar to that of Murez et al. [51], i.e., they also modeled the surface-camera forward scatter as a kernel matrix. However, our model is different in that each row of $\mathbf{K}$ is spatially-variant because we compute the forward scatter considering the object's shape. We show this difference in Figure 4.8. In the model presented by Murez et al. [51], the plane approximation of the scene under orthogonal projection yields a spatially-invariant point spread function. Therefore, Eq. (4.47) is effectively computed using a Fast Fourier Transform. Our spatially-variant kernel matrix makes it infeasible to solve Eq. (4.45) directly.

On the other hand, if the kernel matrix $\mathbf{K}$ can be regarded as a sparse matrix, the computation is feasible. Unfortunately, although the forward scatter effect from a sufficiently distant (infinite) point converges to zero due to attenuation, we found that the effect from points captured within an image cannot be negligible. Figure 4.9(a) shows a row of $\mathbf{K}$ reshaped in a 2D when we observe a plane in a scattering medium. This shows how the observed irradiance of the center of the plane is affected by other points. Figure 4.9(b) shows the profile of the blue line in Fig. 4.9(a). From these figures, we observe that the effect between two points gets close to a very small value as the distance of the points increases; however, it does not converge to zero. Therefore, let $S_r(p)$ be a kernel support which is given as $r \times r$ region centered at pixel p, the total amount $\Theta(r) = \sum_{q \in S_r(p), q \neq p} K_{pq}$ has large effect. For example, in Fig. 4.9 (a), $\Theta(51)/\Theta(101) \approx 51.2\%$, and this means the region outside 51×51 has 48.8% effect due to forward scatter even though the contribution of each point is small. Thus, $\mathbf{K}$ should be considered as a dense matrix and this also makes the computation of solving Eq. (4.45) infeasible.



(a) Murez et al. [51]

(b) Ours

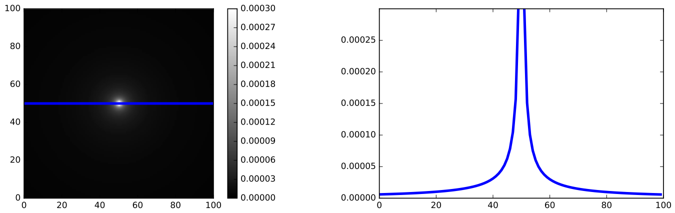
Figure 4.8: Comparison between model of Murez et al. [51] and of ours. Murez et al. [51] assumed that scene can be approximated as plane. Under orthogonal projection, this assumption yields spatially-invariant point spread function. In contrast, we compute forward scatter considering object's shape under perspective projection. Thus, kernel is spatially-variant.

To overcome this problem, we propose an approximation of a large-scale dense matrix $\mathbf{K}$ as a sparse matrix and a constant term which represents global effect. Here, we assume that the value of K_{pq} is close to ϵ $(0 < \epsilon \ll 1)$ in the neighboring support $S(p)$, and we obtain the following approximation:

$$\begin{aligned} L'(p) &= \sum_q K_{pq} L_s(q) && (4.48) \\ &\approx \sum_{q \in S(p)} K_{pq} L_s(q) + \sum_{q \notin S(p)} \epsilon L_s(q) && (4.49) \\ &\approx \sum_{q \in S(p)} K_{pq} L_s(q) + C, && (4.50) \end{aligned}$$

where $C = \sum_q \epsilon L_s(q)$ and we use $\sum_{q \in S(p)} \epsilon L_s(q) \approx 0$ from Eq. (4.49) to (4.50). Then, we define a sparse matrix $\hat{\mathbf{K}}$ as follows:

$$\hat{K}_{pq} = \begin{cases} K_{pq} & (q \in S(p)) \\ 0 & (q \notin S(p)). \end{cases} \qquad (4.51)$$



(a) (b)

Figure 4.9: Visualization of kernel. (a) 2D visualization of row of **K** when we observe plane; (b) profile of blue line in (a).

This yields the following linear system:

$$\begin{bmatrix} \mathbf{L}' \\ 0 \end{bmatrix} = \begin{bmatrix} & & & 1 \\ & \hat{\mathbf{K}} & & \vdots \\ & & & 1 \\ \epsilon & \cdots & \epsilon & -1 \end{bmatrix} \begin{bmatrix} \mathbf{L}_s \\ C \end{bmatrix} . \tag{4.52}$$

We solve this linear system using BiCG stabilization [70] to remove the surface-camera forward scatter.

Note that the size of the kernel support $S(p)$ and the convergence value ϵ have ambiguity. In Section 4.4.3, we evaluate and discuss the size of the kernel support. The plausible value of ϵ might be obtained if we compute all the elements of **K**; however, it requires a large amount of computation. Therefore, we approximated ϵ as follows:

$$\epsilon = \min_{p,q} \left\{ K_{pq} \mid q \in S(p) \right\} . \tag{4.53}$$

4.3.3 Photometric stereo using approximation of lookup table

After removing the backscatter and the surface-camera forward scatter, we can obtain the reflected components $L_s(p)$. We reconstruct the surface normals by applying photometric stereo to $L_s(p)$. From Eqs. (4.29), (4.31) and (4.39), $L_s(p)$ is given as follows:

$$L_s(p) = \frac{I_0}{d_{sp}^2} e^{-T_{sp}} \rho_p \mathbf{n}_p^\top \mathbf{l}_{sp} + \frac{\beta \sigma I_0 \rho_p}{2\pi T_{sp}} G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}). \tag{4.54}$$

Note that this equation is not linear with respect to the normal due to the source-surface forward scatter. We use the following approximation of table $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$ to apply photometric stereo directly to the equation:

$$G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) \approx G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp}). \tag{4.55}$$

In Fig. 4.10, we plot $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$ and $G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp})$ when $T_{sp} = 0.6$ and $T_{sp} = 2$. In each figure, the blue line represents $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$ and the green line represents $G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp})$. Although the error gets to be larger as $\arccos(\mathbf{n}_p^\top \mathbf{l}_{sp})$ increases, these graphs validate this approximation. The detailed discussion of this approximation is given in Section 4.4.2. From this approximation, we can obtain

$$L_s(p) \approx \rho_p I_0 \left(\frac{e^{-T_{sp}}}{d_{sp}^2} + \frac{\beta\sigma}{2\pi T_{sp}} G(T_{sp}, 1) \right) (\mathbf{n}_p^\top \mathbf{l}_{sp}). \tag{4.56}$$

This is a linear equation about normal $\mathbf{n}_p$; hence we apply photometric stereo to this equation as follows:

$$\tilde{\mathbf{n}}_p = \rho_p \mathbf{n}_p = \mathbf{D}^+ \begin{bmatrix} \vdots \\ L_s^{(i)}(p) \\ \vdots \end{bmatrix}, \tag{4.57}$$

$$\mathbf{n}_p = \frac{\tilde{\mathbf{n}}_p}{\|\tilde{\mathbf{n}}_p\|}, \quad \rho_p = \|\tilde{\mathbf{n}}_p\|, \tag{4.58}$$

where $\mathbf{D} \in \mathbb{R}^{m \times 3}$ is

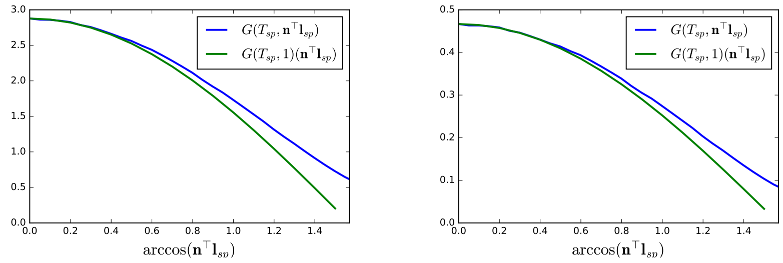
$$\mathbf{D} = \begin{bmatrix} \vdots \\ I_0^{(i)} \left(\frac{e^{-T_{sp}^{(i)}}}{d_{sp}^{(i)2}} + \frac{bc}{2\pi T_{sp}^{(i)}} G(T_{sp}^{(i)}, 1) \right) \mathbf{l}_{sp}^{(i)\top} \\ \vdots \end{bmatrix} \tag{4.59}$$

and $\mathbf{D}^+$ is the pseudo-inverse matrix of $\mathbf{D}$. m is the number of sources. With this linearization, we can avoid the explicit initialization and estimation of the albedo ρ_p during the iteration.

Now, we explain the physical meaning of this approximation. Incident light into a surface point via source-surface forward scatter is asymmetric with respect to the source direction from the surface point. Murez et al. [51] assumed symmetry and approximated these components as one direct beam (see Fig. 4(a) in [51]), i.e., they were described as the constant multiple of the cosine of the source direction and the normal vector. While they estimated this constant by optimization, we compute it by an analytical form $\frac{\rho_p I_0 \beta\sigma}{2\pi T_{sp}} G(T_{sp}, 1)$.

4.3.4 Implementation

In this section, we explain our overall algorithm. Note that the kernel of Eq. (4.46) is only defined on the object's surface; thus, we input a mask image and perform the proposed method on only the object region. In our implementation, the mask image was generated manually, while it may be obtained using a standard segmentation method. Backscatter is removed using a previously proposed



(a) (b)

Figure 4.10: Approximation of lookup table. $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$ (blue line) and $G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp})$ (green line) when (a) $T_{sp} = 0.6$ and (b) $T_{sp} = 2$. Although the error increases as $\arccos(\mathbf{n}_p^\top \mathbf{l}_{sp})$ increases, these graphs validate the approximation $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) \approx G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp})$.

method [15]; however, the resulting image contains high-frequency noise due to SNR degradation. Therefore, we apply a 3 × 3 median filter after removing the backscatter to reduce this high-frequency noise. The optimization of Eq. (4.52) has no restriction so that $\mathbf{L}_s \geq \mathbf{0}$. Thus, the number of positive values in $\{L_s^{(1)}(p), \cdots, L_s^{(m)}(p)\}$ might be less than three and this makes the normal estimation impossible. We fill in the blanks with adjacent normals. We used Poisson solver [68] extended to perspective projection [69], which is described in Section 4.1.2, for normal integration to reconstruct the shape. We additionally intoroduced the following weights to consider the smoothness of normals when computing the derivatives (Eqs. (4.26) and (4.27)):

$$\frac{\partial \tilde{g}_u}{\partial u} = \frac{w_1}{w_1 + w_2}\Big(\tilde{g}_u(u+1, v) - \tilde{g}_u(u, v)\Big) + \frac{w_2}{w_1 + w_2}\Big(\tilde{g}_u(u, v) - \tilde{g}_u(u-1, v)\Big), \tag{4.60}$$

$$w_1 = \exp\left\{\frac{-(1 - \mathbf{n}(u+1, v)^\top \mathbf{n}(u, v))}{s}\right\}, \tag{4.61}$$

$$w_2 = \exp\left\{\frac{-(1 - \mathbf{n}(u, v)^\top \mathbf{n}(u-1, v))}{s}\right\}, \tag{4.62}$$

where we set $s = 0.02$. $\partial \tilde{g}_v / \partial v$ is computed in the same manner.

The overall algorithm is described as follows:

1. Input images and a mask. Initialize the shape and normals.

2. Remove backscatter [15] and apply a median filter to the resulting images.

3. Remove surface-camera forward scatter (Eq. (4.52)).

4. Reconstruct the normals using Eq. (4.56).

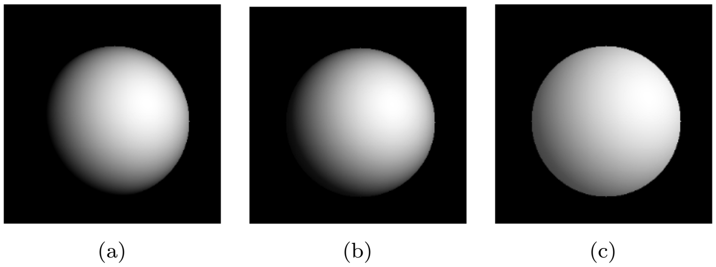


Figure 4.11: Examples of synthesized images. (a) Synthesized image without scattering medium, (b) reflected component $\mathbf{L}_s$, and (c) backscatter subtracted image $\mathbf{L}'$.

5. Integrate the normals and update them from the reconstructed shape.
6. Repeat steps 3–5 until convergence.

4.4 Experiments

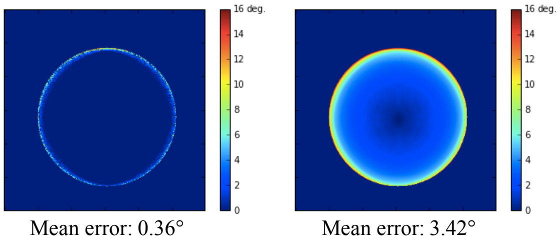
In this section, we describe experiments and evaluation of the proposed method. First, the approximation in the proposed method is evaluated with synthesized data. Then, we demonstrate 3D reconstruction in scattering media with real data. All experiments were done on Intel Core i5 @3.1GHz with 8GB RAM and code was written in C++.

4.4.1 Synthesized data

We evaluate the approximation of the lookup table $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$ and the size of the kernel support for the approximation of the large-scale dense matrix with synthesized data. We generated 8 synthesized images with a 3D model of a sphere under different light sources using our scattering model in Section 4.2. The reflectance property of the surface is Lambertian, and the scattering property was assumed to be isotropic and the parameters were set as $\beta = \sigma = 5.0 \times 10^{-3}$. We show the examples of the synthesized images in Fig. 4.11, where (a) an image without a scattering medium, (b) a reflected component $\mathbf{L}_s$, and (c) a backscatter subtracted image $\mathbf{L}'$. Each image in Fig. 4.11 is 300 × 300 pixels in size.

4.4.2 Discussion of lookup table approximation

We evaluate the effect of the approximation of the lookup table $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp})$. In the synthesized images, diffuse albedos of the 3D model are known. Thus, the source-surface forward scatter component can be subtracted directly. We input the ground truth shape and the synthesized images, and compare the output normals with and without the approximation $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) \approx G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp})$.



(a) (b)

Figure 4.12: Comparison between output normals (a) without and (b) with lookup table approximation. Error maps show angular error of estimated normals. Mean angular error is shown below each figure.

Figure 4.12 (a) and (b) show the angular error of the output normals without and with the lookup table approximation, respectively. These results demonstrate that the accuracy of the normal estimation is more deteriorated as the normal vector has a larger angle to the optical axis.

4.4.3 Discussion of kernel support

We discuss the size of the kernel support. Here, we define the support $S(p)$ as the set of 3D points captured in a $r \times r$ region centered at an observed pixel p. The synthesized images and the ground truth shape are input to the proposed method with a different support size r. The effects of the support size are evaluated from the output normals.

The results are shown in Fig. 4.13. A horizontal axis in Fig. 4.13 denotes the support size r and a vertical axis denotes the mean angular error of the output normals. A green line and a blue line show the results without and with the lookup table approximation, and two dashed lines show each error in Fig. 4.12, respectively. The memory complexity of the matrix $\mathbf{K}$ is $\mathcal{O}(Nr^2)$ where N is the number of processed pixels, and we show the actual computation time for each support size by a red line in Fig. 4.13.

As can be seen in Fig. 4.13, the normals estimation is basically improved as the support size r increases. However, when the support size is small, the angular error without the lookup table approximation is larger than with the approximation. The reason is that the estimated ϵ in Eq. (4.53) is larger than the true value when the support size is small. We approximate the large-scale dense matrix to a sparse matrix by introducing the constant term $C = \sum_q \epsilon L_s(q)$. This constant term increases as ϵ becomes larger.

Now, we discuss the influence of the larger constant term. In the following discussion, for simplicity, the blur effect is not considered. Here, let $[L_{min}, L_{max}]$ be the range of the radiance of an original image. We quantify the contrast of the image as $\kappa = L_{max}/L_{min} > 1$. If the constant C is added, the range becomes $[L_{min} + C, L_{max} + C]$. Similarly, let C' be the estimated constant and

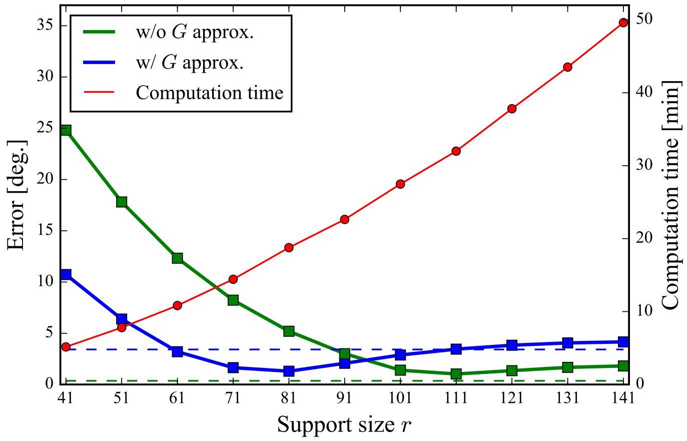


Figure 4.13: Experiments with different support sizes. Horizontal axis denotes size r and vertical axis denotes mean angular error of output normals. Green line and blue line show results without and with lookup table approximation, and two dashed lines denote each mean angular error in Fig. 4.12, respectively. Red line shows actual computation time for kernel calculation.

$[L'_{min}, L'_{max}]$ be the range of the reconstructed image, that is,

$$L'_{min} = L_{min} + C - C', \quad (4.63)$$

$$L'_{max} = L_{max} + C - C'. \quad (4.64)$$

We define the contrast of the reconstructed image as $\kappa' = L'_{max}/L'_{min} > 1$. The following relation can be obtained:

$$\kappa' - \kappa = (1 - \kappa')\frac{C - C'}{L_{min}}. \quad (4.65)$$

When $C' > C$, the right side of this equation is positive. Thus, the larger constant term increases the contrast of the image. As a result, the output normals have larger curvature than the true normals.

On the other hand, the lookup table approximation yields globally smaller curvature described in Sec. 4.4.2. As a result, these two approximation compensate each error. Subsequently, each result without and with the lookup table approximation approaches the dashed line as shown in Fig. 4.13. These dashed lines correspond to the mean angular error in Fig. 4.12. However, when we adopt a too large support size, the estimation is affected by the error due to the assumption $\sum_{q \in S(p)} \epsilon L_s(q) \approx 0$.

From Fig. 4.13 and the above discussion, the error can be suppressed within the almost ideal error, which is shown in Fig. 4.12, if we set sufficiently large support size, whereas the small support size is preferable with respect to memory

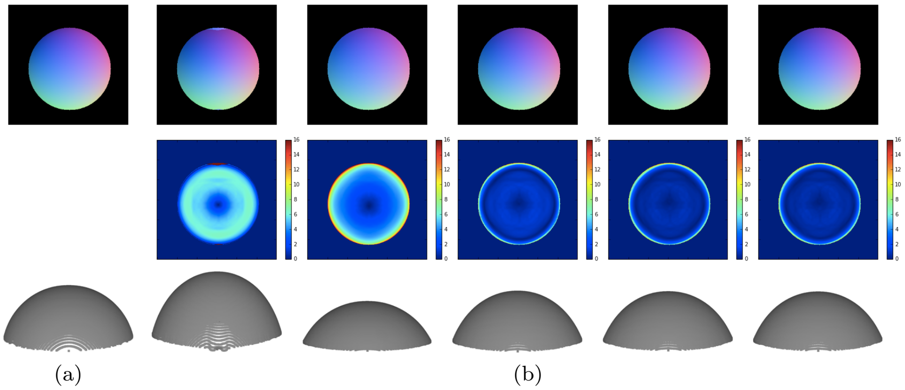


Figure 4.14: Results of synthesized data. (a) Ground truth and (b) output of each iteration from left to right. (top row) output normals. (middle row) reconstructed shapes.

Table 4.1: Mean angular error of output of each iteration with synthesized data

	Iteration 1	2	3	4	5	Input GT
Error (deg.)	5.20	4.65	1.43	1.29	1.29	1.30

and computational complexity. In our experiments with synthesized and real data, the support sizes from $r = 61$ to $r = 81$ gave the efficient results.

4.4.4 Results and convergence

In this section, we describe experiments with synthesized data when the input shape is initialized as a plane and demonstrate how the output shape converges during the iteration. In this experiment, we set the support size as $r = 81$.

The results are shown in Fig. 4.14 and Table 4.1. Figure 4.14(a) shows the ground truth and (b) shows the output of each iteration from left to right. The top row shows the normals map, the middle row shows the angular error of the output, and the bottom row shows the reconstructed shapes. Table 4.1 shows the mean angular error of each output. Input GT in Table 4.1 denotes the error when we removed scattering effects with the ground truth shape and reconstructed the 3D shape inversely. As shown in Fig. 4.14, the shape converged while oscillating in height. This convergence was seen in the experiments with the real data (see Fig. 4.17). Murez et al. [51] approximated an object as a plane. In this experiment, we initialized the object as a plane. The improvement from the first to the last iteration shows the effect of our method.

4.4.5 Experiments with real data

In this section, we describe the results of experiments with real data. We demonstrate that the proposed method can reconstruct the shapes of objects in scat-

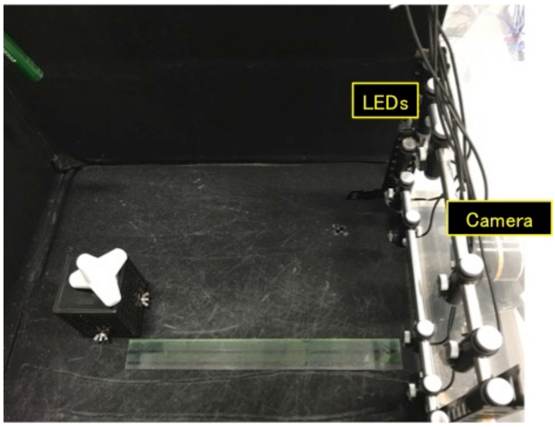
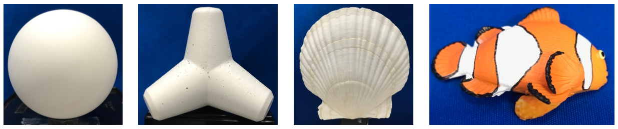


Figure 4.15: Experimental environment. This is top view of tank.



(a) *sphere* (b) *tetrapod* (c) *shell* (d) *fish*

Figure 4.16: Target object

tering media where forward scatter occurs.

Experimental environment

The experimental environment is shown in Fig. 4.15. We used a 60-cm cubic tank and placed a target object in it. We used diluted milk as the scattering medium. The medium parameters were set with reference to [71]. Table 4.2 shows medium parameters used in our experiments. A ViewPLUS Xviii 18-bit linear camera whose spatial resolution is 1024 × 1280 pixels was mounted in close contact with the tank, and eight LEDs were located in the tank. The input images were captured at an exposure of 33 ms. We captured 60 images under the same condition, and these images were averaged to make input images robust to noise caused by the imaging system; thus, eight averaged images were input to the proposed method.

The camera was calibrated using the method presented in [72]. To consider refraction on the wall of the tank, calibration was performed when the tank was full of water. The locations of the LEDs were measured manually, and each radiant intensity I_0 was calibrated using a white Lambertian sphere.

The target objects are shown in Fig. 4.16 (*sphere*, *tetrapod*, *shell*, and *fish*).

Table 4.2: Medium parameters used in our experiments

Milk / Water	Scattering coef. β [/mm]	Extinction coef. σ [/mm]
10mL / 120L	1.67×10^{-3}	1.67×10^{-3}
20mL / 120L	3.33×10^{-3}	3.33×10^{-3}
30mL / 120L	5.00×10^{-3}	5.00×10^{-3}

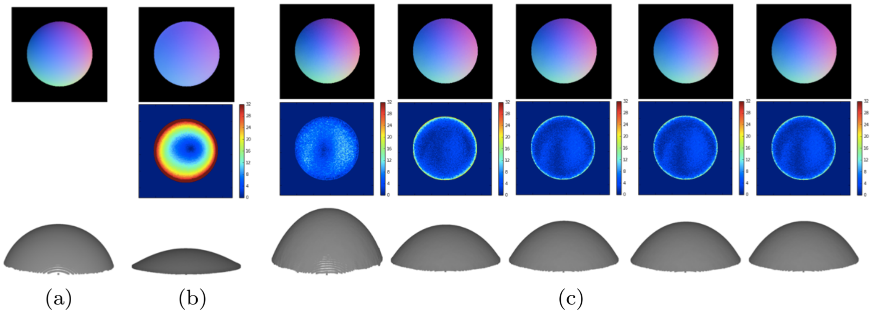


Figure 4.17: Results of *sphere*. (a) Ground truth, (b) result of [15], and (c) proposed method. (top row) output normals, (middle row) error map of angles, (bottom row) reconstructed shape. These experiments demonstrate that our method can reconstruct shape in highly turbid media, in which forward scatter is caused.

Results

We compared the proposed method with a previously proposed method [15] that models only backscatter. In each experiment, the target object was initialized as a plane for the iteration. We set the size of the kernel support as $r = 61$.

First, we evaluated the proposed method quantitatively using *sphere*. In this experiment, we placed 120 L of water and 30 mL of milk in the tank. In Fig. 4.1, a part of the input images are shown. The size of the image is 280 × 280 pixels.

The results are given in Fig. 4.17, where Fig. 4.17(a) shows the ground truth, (b) shows the result of the backscatter-only modeling [15], and (c) shows the result of the proposed method, which depicts the output of each iteration from left to right. The top row shows the output normals, the middle row shows the error maps of angles, and the bottom row shows the reconstructed shapes. These experimental results demonstrate that the proposed method can reconstruct the object's shape in highly turbid media, in which the method that does not consider forward scatter fails. Table 4.3 shows the mean angular error of the results of the backscatter-only modeling [15] and the output of each iteration of the proposed method. The error reaches convergence during a few iterations. This convergence is similar to that of the synthesized result (see Fig. 4.14).

Figures 4.18, 4.19, and 4.20 show the results for *tetrapod*, *shell*, and *fish*.

Table 4.3: Mean angular error of *sphere*. Error of proposed method is lower than that of backscatter-only modeling [15], and a few iterations are sufficient to reach convergence.

	[15]	Iteration 1	2	3	4	5
Error (deg.)	19.48	5.96	4.38	3.62	3.66	3.66

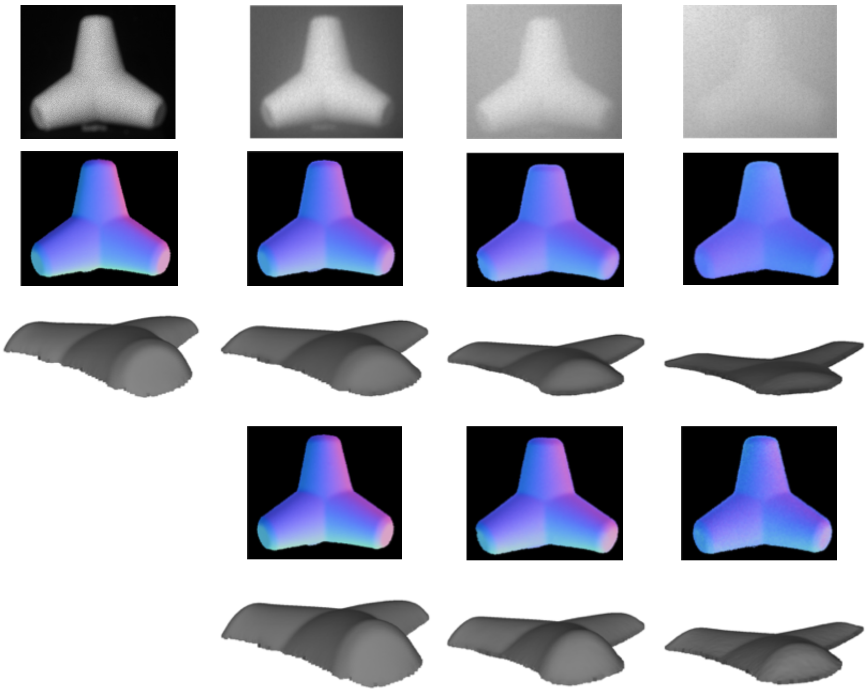
In each figure, (a) shows the result obtained in clear water and (b) shows the results of the existing [15] (second and third rows) and proposed (fourth and fifth rows). The top row shows one of the input images. The size of each input image shown in Figs. 4.18, 4.19, and 4.20 is 250×290, 420×400, and 200×320 pixels, respectively. We changed the concentration of the scattering medium during these experiments (we mixed 10, 20, and 30 mL of milk with 120 L of water from left to right). As can be seen, the result of the existing method [15] becomes flattened as the concentration of the scattering medium increases. In contrast, the proposed method reconstructs the shape correctly in highly turbid media. However, although the proposed method can reconstruct the local gradient of *shell*, the results have globally larger curvature than reconstruction in clear water. As mentioned previously, a larger constant term makes the image more contrasted. The constant term $C = \sum_q \epsilon L_s(q)$ also depends on the number of processed pixels. In this experiment, the number of the processed pixels in the *shell* image is more than that in the other object images due to its size. As a result, the reconstructed shape of *shell* has a large curvature.

The result of *fish* in Fig. 4.20 demonstrates the effectiveness on objects with texture. The bottom row of Fig. 4.20(b) is estimated surface albedos. The proposed method can recover albedos as well as a 3D shape.

4.5 Conclusion

In this chapter, we have proposed a photometric stereo method in scattering media that considers forward scatter. The proposed analytical model of the single scattering model differs from the previous works [51] in that forward scatter depends on the object's shape. The shape dependency of the forward scatter makes it infeasible to remove. To address this problem, we have proposed an approximation of the large-scale dense matrix that represents the forward scatter as a sparse matrix. Our experimental results demonstrate that the proposed method can reconstruct a shape in highly turbid media.

However, the ambiguity of the optimized support size of the kernel remains. We set aside an adaptive estimation of the support size for future work. A limitation of the proposed method is that it requires a mask image of the target object. However, in highly turbid media, it may be difficult to obtain an effective mask image. In addition, we must initialize the object's shape, which may be solved using a depth estimation method in scattering media [14, 73, 74].



(a) (b)

Figure 4.18: Results of *tetrapod*. (a) Reconstruction in clear water and (b) results of [15] (second and third rows) and proposed method (fourth and fifth rows). Top row is one of input images. Concentration of scattering medium increases from left to right. Result of [15] becomes flattened as concentration of scattering medium increases. In contrast, proposed method reconstructs shape correctly in highly turbid media.

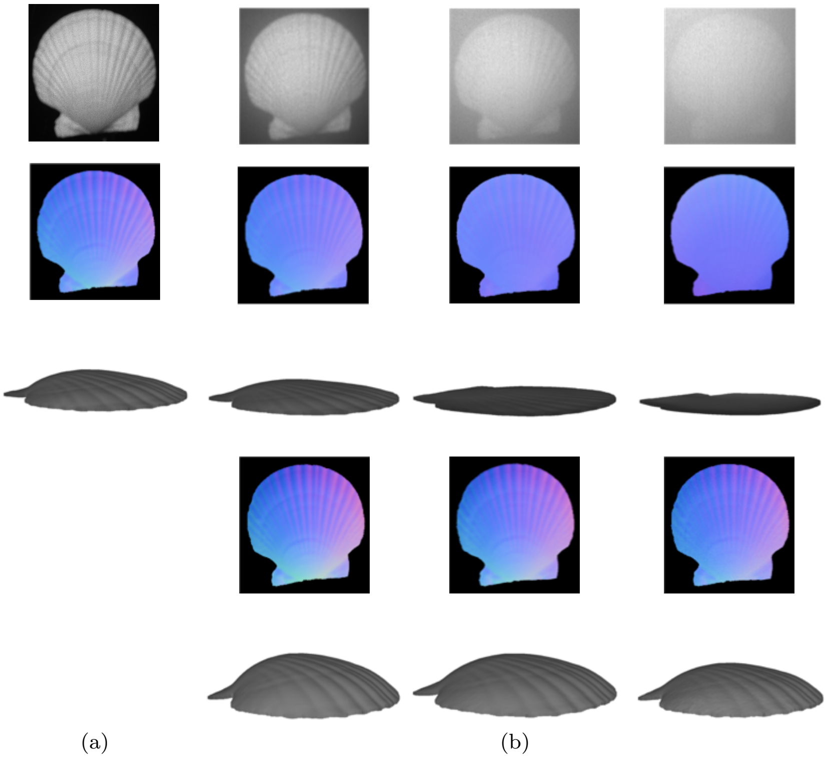


Figure 4.19: Results of *shell*. Details are similar to those of Fig. 4.18. Although proposed method can reconstruct local gradient of *shell*, globally, results have larger curvature than reconstruction in clear water due to constant term increase.

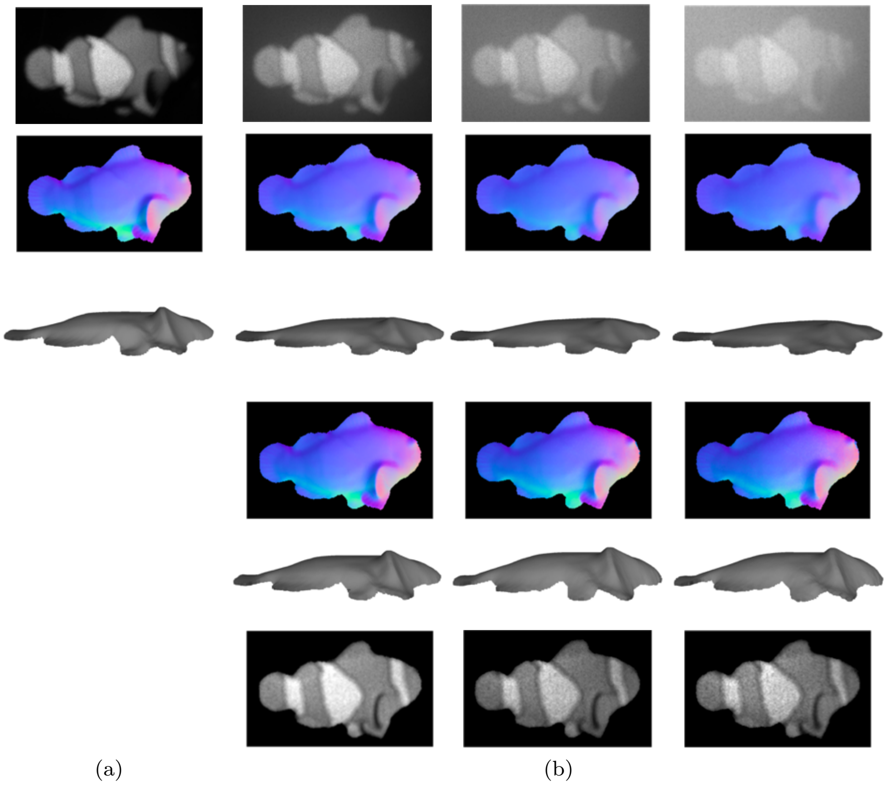


Figure 4.20: Results of *fish*. Regardless of texture, proposed method can improve 3D reconstruction in scattering media. Bottom row of (b) shows estimated albedos. Proposed method is also effective on albedos recovery.

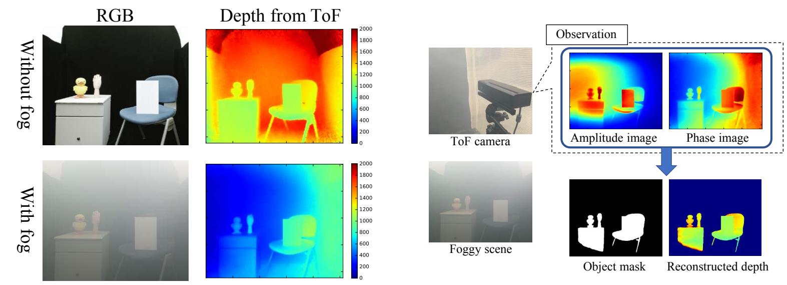
Chapter 5

Time-of-Flight in Scattering Media

In thic chapter, we discuss depth measurement with a ToF camera in scattering media. There are several architectures for ToF cameras. We use a continuous-wave ToF camera that emits a modulated sinusoid signal into a scene and then measures the amplitude of light that bounces off an object surface and the phase shift between the illumination and received signal. These observations are represented as an amplitude image and a phase image as shown in Fig. 5.1(b). Since the phase shift depends on an optical path, we can reconstruct the depth from the phase shift. We denote the observation of an object surface by direct component.

This architecture assumes that each camera pixel observes a single point in a scene. Similar to common RGB cameras, however, the observed signal in scattering media includes a scattering component due to light scattering as well as a direct component. The amplitude and phase shift suffer from the scattering effect, and this causes error of depth measurement as shown in Fig. 5.1(a). In Chapter 2, we provided scattering models where the pixel intensity of common RGB cameras is considered. In this chapter, we formulate a scattering model in amplitude and phase space. ToF cameras emit light signals from an internally mounted light source. Thus, the single scattering model can be used for the observation of ToF measurement in scattering media. We also leverage the saturation of a backscatter component, which occurs in RGB space [14, 15], to recover the direct component. We assume that a target scene consists of an object region and a background that only contains a scattering component. This allows us to estimate the scattering component simply by observing the background. The proposed automatic scene segmentation enables simultaneous obstacle detection and depth reconstruction as shown in Fig. 5.1(a).

The rest of this chapter is organized as follows: In Section 5.1, we overview related work on ToF measurement in scattering media, which can be considered as multipath interference (MPI) problems. In Section 5.2, the observation of



(a) Depth error due to light scattering (b) Overview of proposed method

Figure 5.1: ToF in scattering media. (a) Depth measurement suffers from scattering effect in scattering media such as foggy scene. (b) Overview of proposed method. Continuous-wave ToF camera captures amplitude image and phase image. From these images captured in participating media, we estimate object region and recover depth simultaneously.

a ToF camera is modeled with the single scattering model in amplitude and phase space. In Section 5.3, we discuss a method for scene segmentation that is formulated as robust estimation where the object region is regarded as outliers, and it enables the simultaneous estimation of an object region and depth on the basis of an iteratively reweighted least squares (IRLS) optimization scheme [75, 76, 77, 78]. In Section 5.4, we demonstrate the effectiveness of the proposed method using captured images from a ToF camera in real foggy scenes and evaluate the applicability with synthesized data. Finally, Section 5.5 concludes this chapter.

5.1 Related work

A ToF camera assumes that each camera pixel observes a single point in a scene. In scattering media, however, the measurement also includes scattered light. This problem is known as multipath interference (MPI). MPI is caused not just by light scattering in scattering media but also by subsurface scattering or interreflection in common scenes. Thus, many previous studies have tackled MPI compensation [79, 80, 81, 82, 83].

In this dissertation, we limit our focus to MPI caused by light scattering in scattering media. ToF measurement in scattering media has been proposed by [84, 85]. Heide et al. [84] developed a scattering model based on exponentially modified Gaussians for transient imaging using a photonic mixer device (PMD) [86]. Satat et al. [85] demonstrated that scattered photons observed with a SPAD have gamma distribution and leveraged this observation to separate received photons into a directly reflected component and a scattering component. Our method differs from these approaches in that we just use an off-the-shelf ToF camera such as Kinect v2 with no special hardware modification. The concurrent work by Muraji et al. [87] also used a continuous-wave ToF cam-

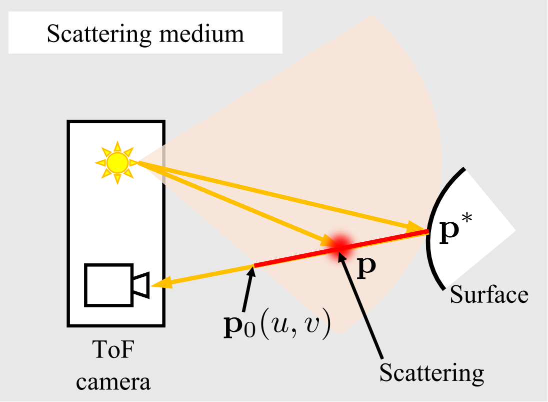


Figure 5.2: ToF camera with limited beam angle in scattering media. Light interacts with scattering medium on line of sight and then arrives at camera pixel. Total scattering component is sum of scattered light on red line in figure, which depends on limited beam angle of light source.

era. They removed scattering effect using multiple modulation frequencies. We address different problem settings as follows: (1) we model spatially varying scattering components due to a limited lighting angle as explained in Section 5.3.1; (2) we model the simultaneous estimation of object regions and scattering components as a single optimization problem.

5.2 ToF observation in scattering media

In this section, we describe our image formation model for a continuous-wave ToF camera in scattering media on the basis of the single scattering model. In Chapter 4, we introduce a backscatter component and a forward scatter component which is caused in highly turbid media. We assume here that the effect of forward scattering are negligibly small.

A continuous-wave ToF camera illuminates a scene with amplitude-modulated light and then measures the amplitude of received signal α and phase shift φ between the illumination and received signal. This observation can be described using a phasor [88], as

$$\alpha e^{j\varphi} \in \mathbb{C}. \tag{5.1}$$

Since the phase shift is proportional to the depth of an object, we can compute the depth as

$$z = \frac{c\varphi}{4\pi f}, \tag{5.2}$$

where z is depth, c is the speed of light, and f is the modulation frequency of the camera.

In scattering media, the observation contains scattered light. Figure 5.2 shows the observation of a ToF camera in scattering media. Similar to a RGB camera described in Chapter 2, light interacts with the medium on the line of sight and then arrives at the camera pixel. Thus, the observed scattering component is the sum of scattered light on the line of sight. Now, we consider the 3D coordinate, the origin of which is the camera center. When the camera observes a surface point $\mathbf{p}^* \in \mathbb{R}^3$ at a camera pixel (u, v), the total observation $\tilde{\alpha}(u, v; \mathbf{p}^*)e^{j\tilde{\varphi}(u,v;\mathbf{p}^*)}$ can be written as

$$\begin{aligned}&\tilde{\alpha}(u, v; \mathbf{p}^*)e^{j\tilde{\varphi}(u,v;\mathbf{p}^*)}\\&= \alpha_d(u, v; \mathbf{p}^*)e^{j\varphi_d(u,v;\mathbf{p}^*)} + \int_{\|\mathbf{p}\|=\|\mathbf{p}_0(u,v)\|}^{\|\mathbf{p}^*\|} \alpha(u, v; \mathbf{p})e^{j\varphi(u,v;\mathbf{p})} d\|\mathbf{p}\|,\end{aligned} \tag{5.3}$$

where $\alpha_d(u, v; \mathbf{p}^*)$ and $\varphi_d(u, v; \mathbf{p}^*)$ are the direct components. $\alpha_d(u, v; \mathbf{p}^*)$ depends on the surface albedo, shading, and attenuation, which is caused by the medium as well as the inverse square law. $\alpha(u, v; \mathbf{p})e^{j\varphi(u,v;\mathbf{p})}$ is the observation of scattered light at a position $\mathbf{p}$. Note that although the scattering component can be written using an integral, the domain of the integral (red line in Fig. 5.2) depends on the relative position between the light source and camera pixel. This is because an ideal point light source irradiates a scene with isotropic intensity, while a practical illumination such as a spotlight has a limited beam angle [15].

In Section 4.3.1, the backscatter component is assumed to be saturated close to the camera in RGB space. This assumtion holds under a near light source in scattering media [14, 15]. We also leverage this assumtion for ToF measurement, that is, there exists $\mathbf{p}_{saturate}$ for which

$$\|\mathbf{p}\| \geq \|\mathbf{p}_{saturate}\| \Rightarrow \alpha(u, v; \mathbf{p}) = 0. \tag{5.4}$$

Therefore, we can rewrite Eq. (5.3) as

$$\begin{aligned}&\tilde{\alpha}(u, v; \mathbf{p}^*)e^{j\tilde{\varphi}(u,v;\mathbf{p}^*)}\\&= \alpha_d(u, v; \mathbf{p}^*)e^{j\varphi_d(u,v;\mathbf{p}^*)} + \underbrace{\int_{\|\mathbf{p}\|=\|\mathbf{p}_0(u,v)\|}^{\|\mathbf{p}_{saturate}\|} \alpha(u, v; \mathbf{p})e^{j\varphi(u,v;\mathbf{p})} d\|\mathbf{p}\|}_{=\alpha_s(u,v)e^{j\varphi_s(u,v)}},\end{aligned} \tag{5.5}$$

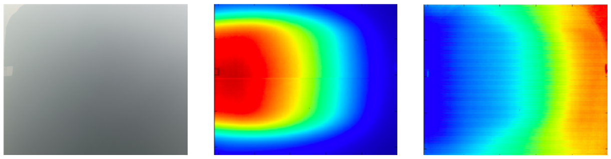
where $\alpha_s(u, v)$ and $\varphi_s(u, v)$ are the scattering components, which depend on only the camera pixel (u, v) rather than the object depth.

Although the observation consists of the direct component $\alpha_d(u, v; \mathbf{p}^*)e^{j\varphi_d(u,v;\mathbf{p}^*)}$ and the scattering component $\alpha_s(u, v)e^{j\varphi_s(u,v)}$, the attenuation due to the medium reduces the direct component dramatically. Thus, if the camera observes a distant point $\mathbf{p}_{far}$, the amplitude of the reflected light fades away, that is,

$$\alpha_d(u, v; \mathbf{p}_{far}) = 0. \tag{5.6}$$

Therefore, the observation of the distant point includes only a scattering component:

$$\tilde{\alpha}(u, v; \mathbf{p}_{far})e^{j\tilde{\varphi}(u,v;\mathbf{p}_{far})} = \alpha_s(u, v)e^{j\varphi_s(u,v)}. \tag{5.7}$$



(a) RGB image (b) Amplitude image (c) Phase image

Figure 5.3: Observation of black surface in foggy scene. Black surface approximates distant observation where only scattering component can be observed because reflected light from scene gets attenuated. Note that observed scattering component is inhomogeneous due to limited beam angle of illumination.

Figure 5.3 shows amplitude and phase images when the camera observes a black surface in a foggy scene. The intensity of reflected light from the black surface is very small, so this approximates a distant observation where only a scattering component can be observed. As discussed above, in both the amplitude and phase images, the scattering component is inhomogeneous because the illumination has a limited beam angle.

The proposed method is based on the assumption that the scattering component is saturated close to the camera. However, this assumption is not precise in some cases, for example, when a scene is extremely close to the camera. The measurement range of our method is between a saturation point and a background point that has no direct component. We investigate the effective range of the proposed method in Section 5.4.3.

5.3 Simultaneous estimation of object region and depth

As explained in the previous section, a scattering component depends on the position of a camera pixel rather than a target object. In addition, only the scattering component is observed in the background where an object is farther away. Thus, our goal is to estimate the scattering component in an object region from the observation of the background. After estimating scattering components $\alpha_s(u,v)$ and $\varphi_s(u,v)$ at each pixel, we compute the amplitude and phase shift of a direct component from Eq. (5.5):

$$\alpha_d = \sqrt{(\tilde{\alpha}\cos\tilde{\varphi} - \alpha_s\cos\varphi_s)^2 + (\tilde{\alpha}\sin\tilde{\varphi} - \alpha_s\sin\varphi_s)^2}, \tag{5.8}$$

$$\varphi_d = \arg\left((\tilde{\alpha}\cos\tilde{\varphi} - \alpha_s\cos\varphi_s) + j(\tilde{\alpha}\sin\tilde{\varphi} - \alpha_s\sin\varphi_s)\right), \tag{5.9}$$

where an operator arg returns the argument of a complex number. Then, depth is recovered substituting the phase into Eq. (5.2).

In this section, we describe how our method divides camera pixels into an object region and a background, and simultaneously estimates the scattering component in the object region. First, we introduce two priors to estimate the

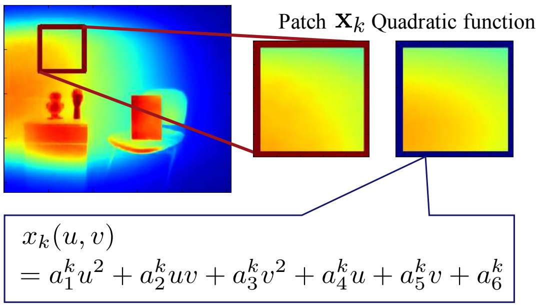

Patch $\mathbf{x}_k$
Quadratic function
$x_k(u,v)$
$= a_1^k u^2 + a_2^k uv + a_3^k v^2 + a_4^k u + a_5^k v + a_6^k$

Figure 5.4: Local quadratic prior. We assume that scattering component can be represented with quadratic function in local image patch.

scattering component, and then the problem is formulated as robust estimation, which allows us to extract the object region as outliers. In the following, with a slight alteration of notation, we refer to both an amplitude image and a phase image as an image, since we process both images in the same manner.

5.3.1 Prior of scattering component

We can estimate the scattering component of an object region from a background because the component does not depend on the object. Tsiotsios et al. [15] approximated backscatter as a quadratic function in a captured image. Similarly to their work, we also introduce priors, *local quadratic prior* and *global symmetrical prior*, that allow us to estimate the scattering component.

Local quadratic prior In our ToF setting, we found that a scattering component cannot be approximated globally with a simple function such as Tsiotsios et al. [15] where a scattering component is fitted with a quadratic function all over an image. Thus, as shown in Fig. 5.4, we assume that a scattering component can be represented with a quadratic function in a local image patch, that is,

$$\begin{aligned} x_k(u,v) &= a_1^k u^2 + a_2^k uv + a_3^k v^2 + a_4^k u + a_5^k v + a_6^k \\ &= \mathbf{a}_k^\top \mathbf{u}, \end{aligned} \tag{5.10}$$

where $x_k(u,v)$ is the value at a pixel (u,v) in a local image patch $\mathbf{x}_k$. $\mathbf{u} = [u^2 \;\; uv \;\; v^2 \;\; u \;\; v \;\; 1]^\top$ is a 6-dimensional vector and $\mathbf{a}_k = [a_1^k \;\; a_2^k \;\; a_3^k \;\; a_4^k \;\; a_5^k \;\; a_6^k]^\top$ denotes the coefficients of the quadratic function in patch $\mathbf{x}_k$.

Global symmetrical prior However, this local prior is not useful when there exists a large object region and a quadratic function is also fitted into the values in that region. To address this problem, we introduce a global prior to the scattering component.

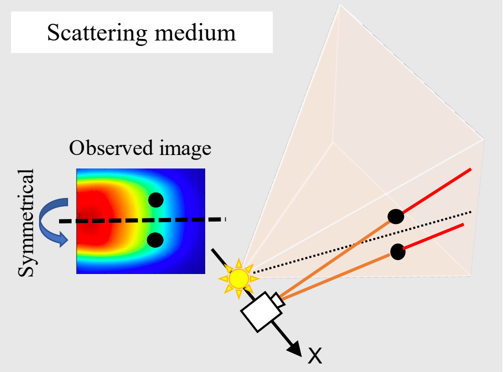


Figure 5.5: Global symmetrical prior. When camera and light source are collocated on line that is parallel to horizontal axis of image, observed scattering component has symmetry because integral domain of pixel is consistent with that of symmetrical pixel with respect to central axis of image.

As discussed in section 5.2, a scattering component depends on the relative position between a camera pixel and a light source. This is because the individual starting points of the integral in Eq. (5.3) differ from each other. Meanwhile, as shown in Fig. 5.5, we assume that the camera and light source are collocated on the line that is parallel to the horizontal axis of the image. ToF devices can easily be built on the basis of this setting (e.g., Kinect v2 has this setting). In this case, the integral domain of a pixel is consistent with that of the symmetrical pixel with respect to the central axis of the image. Thus, the observed scattering component also has symmetry, and we leverage this symmetry as a global prior.

5.3.2 Formulation as robust estimation

We formulate the scattering component estimation problem as robust estimation. Specifically, we solve the following optimization problem:

$$\min_{\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_K} \sum_{i=1}^{N} \rho\left(\frac{x_i - \tilde{x}_i}{\sigma_1}\right) + \gamma_1 \sum_{k=1}^{K} \|\mathbf{U}\mathbf{a}_k - \mathbf{x}_k\|^2 + \gamma_2 \|\mathbf{F}\mathbf{x} - \mathbf{x}\|^2 + \gamma_3 \|\nabla \mathbf{x}\|^2. \quad (5.11)$$

The first term of Eq. (5.11) is a data term where $\tilde{\mathbf{x}} = [\tilde{x}_1 \ \cdots \ \tilde{x}_N]^\top$ and $\mathbf{x} = [x_1 \ \cdots \ x_N]^\top$ are a captured image and a scattering component, respectively. N is the number of camera pixels, and σ_1 is a scale parameter. We use a nonlinear differentiable function $\rho(x)$ rather than square error x^2, which allows us to make the estimation robust against outliers. In this study, we simply use the residual of the observation and the scattering component as the data term, i.e., pixels that contain a direct component are regarded as outliers.

Algorithm 2 Simultaneous estimation of scattering component and object region

```
Require: Image $\tilde{\mathbf{x}}$
Ensure: Scattering component $\mathbf{x}$ and object mask $\mathbf{w}$
  Coarse level optimization (Eq. (5.16)):
      $\mathbf{W} \leftarrow \mathbf{I}$, $\mathbf{a}_k \leftarrow \underset{\mathbf{a}_k}{\operatorname{argmin}} \|\mathbf{U}\mathbf{a}_k - \tilde{\mathbf{x}}_k\|^2$
      repeat
        Solve Eq. (5.12) for $\mathbf{x}$
        Solve Eq. (5.12) for $\mathbf{a}_1, \cdots, \mathbf{a}_K$
        if first iteration then
              Compute $\sigma_2$
        end if
        Update $\mathbf{w}$ in patch-wise
      until converged
  Fine level optimization (Eq. (5.11)):
      Initialize $\mathbf{w}$ and $\mathbf{a}_1, \cdots, \mathbf{a}_K$ with the output of the coarse level
      repeat
        Solve Eq. (5.12) for $\mathbf{x}$
        Solve Eq. (5.12) for $\mathbf{a}_1, \cdots, \mathbf{a}_K$
        if first iteration then
              Compute $\sigma_1$
        end if
        Update $\mathbf{w}$ in pixel-wise
      until converged
  Binarize $\mathbf{w}$
```

We use three additional regularization terms. The second term represents the local prior. K is the number of patches for local quadratic function fitting. $\mathbf{U}$ is an $N_k \times 6$ matrix where N_k is the number of pixels in patch $\mathbf{x}_k \in \mathbb{R}^{N_k}$ and each row of $\mathbf{U}$ is a vector $\mathbf{u}$ that corresponds to each pixel coordinate. In this study, these patches do not overlap each other. The third term represents the global prior where $\mathbf{F} \in \mathbb{R}^{N \times N}$ is a matrix that flips an image vertically. The last term is a smoothing term where ∇ denotes a gradient operator. This smoothing accelerates the optimization. Hyperparameters $\gamma_1, \gamma_2, \gamma_3$ control the contribution of each term.

5.3.3 IRLS and object region estimation

We minimize Eq. (5.11) with respect to a scattering component $\mathbf{x}$ and the coefficients of quadratic functions $\mathbf{a}_1, \cdots, \mathbf{a}_K$. However, the nonlinearity of $\rho(x)$ makes it difficult to obtain a closed-form solution. For efficient computation, the IRLS optimization was developed in the literature [75, 76]. IRLS minimizes weighted least squares iteratively and the weight is updated using the current estimate in each iteration. The objective function in Eq. (5.11) is transformed into weighted least squares as follows:

$$\min_{\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_K} (\mathbf{x}-\tilde{\mathbf{x}})^\top \mathbf{W}(\mathbf{x}-\tilde{\mathbf{x}}) + \gamma_1' \sum_{k=1}^{K} \|\mathbf{U}\mathbf{a}_k - \mathbf{x}_k\|^2 + \gamma_2'\|\mathbf{F}\mathbf{x}-\mathbf{x}\|^2 + \gamma_3'\|\nabla\mathbf{x}\|^2, \tag{5.12}$$

where $\mathbf{W} = \mathrm{diag}(\mathbf{w})$ is an $N \times N$ matrix and $\mathbf{w} = [w_1, \cdots, w_N]^\top$ is the weight for each error $x_i - \tilde{x}_i$. Hyperparameters are given as $\gamma_*' = 2\sigma_1^2\gamma_*$. Equation (5.12) is quadratic with respect to the scattering component $\mathbf{x}$, and thus is easy to optimize. In each iteration, we solve Eq. (5.12) for $\mathbf{x}$ and $\mathbf{a}_1, \cdots, \mathbf{a}_K$, and the weight can be updated using the current estimate as

$$w_i = \frac{\rho'\left((x_i - \tilde{x}_i)/\sigma_1\right)}{(x_i - \tilde{x}_i)/\sigma_1}. \tag{5.13}$$

The specific update rule of the weight depends on the nonlinear function $\rho(x)$. In this study, we use the following function as $\rho(x)$:

$$\rho(x) = \begin{cases} \frac{c^2}{6}\left[1 - \left\{1 - \left(\frac{x}{c}\right)^2\right\}^3\right] & if \;\; |x| \le c \\ \frac{c^2}{6} & otherwise. \end{cases} \tag{5.14}$$

This function yields the following update:

$$w_i = \begin{cases} \left\{1 - \left(\frac{r_i}{c}\right)^2\right\}^2 & if \;\; |r_i| \le c \\ 0 & otherwise, \end{cases} \tag{5.15}$$

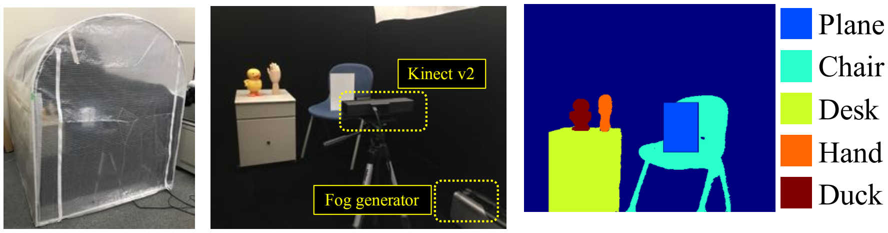
where $r_i = (x_i - \tilde{x}_i)/\sigma_1$, and c is a tuning parameter. This update is referred to as Tukey's biweight [89, 76], where $0 \le w_i \le 1$.

Object region from IRLS weight The weight controls the robust estimation, that is, a large error term reduces the corresponding weight. In this study, we consider the object region as outliers, and thus the weight in the object region should be small. Therefore, we can leverage the IRLS weight to extract the object region from the image.

Selection of $\rho(x)$ In Eq. (5.11), any robust functions can be selected for $\rho(x)$ (e.g., Huber loss [90] or t-distribution [91]). However, we found that Tukey's biweight [89, 76] is suitable because of its property. The robustness of Tukey's biweight is achieved by truncating outliers (Eq. (5.15)), which is an object region in our case. This property results in a clear object boundary as shown in the experiments. In contrast, the weight of Huber loss, for example, has long tails, and this makes the optimization unstable.

5.3.4 Coarse-to-fine optimization

The accurate object region extraction is critical for the effectiveness of the scattering component estimation. In Section 5.3.1, we introduced the local and



(a) (b)

Figure 5.6: (a) Experimental environment. (b) Target objects

global priors of the scattering component to deal with a large object region. To make the region extraction more robust, we developed a coarse-to-fine optimization scheme. Before solving Eq. (5.11), we optimize the following objective function:

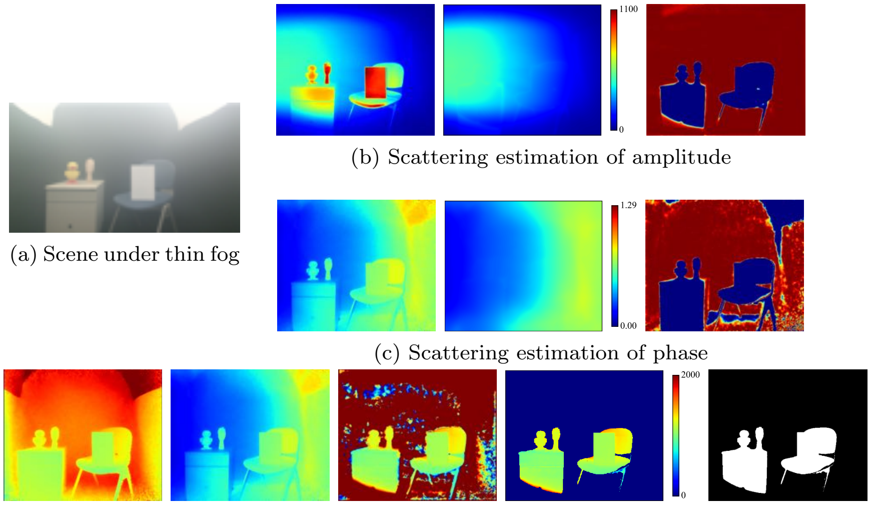
$$\min_{\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_K} \sum_{k=1}^{K} \rho\left(\frac{\|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|}{\sigma_2}\right) + \gamma_1 \sum_{k=1}^{K} \|\mathbf{U}\mathbf{a}_k - \mathbf{x}_k\|^2 + \gamma_2\|\mathbf{F}\mathbf{x} - \mathbf{x}\|^2 + \gamma_3\|\nabla\mathbf{x}\|^2. \tag{5.16}$$

This is similar to the patch-based robust regression proposed by [92]. The difference from Eq. (5.11) is that the data term consists of patch-wise errors. Equation (5.16) can be transformed into IRLS as well as Eq. (5.11) where $\gamma'_* = 2\sigma_2^2\gamma_*$, and the IRLS weight is updated patch-wise rather than pixel-wise.

Algorithm 2 shows the overall procedure of the simultaneous estimation of a scattering component and an object region. We first solve Eq. (5.16) for the weight in a patch level and then solve (5.11) in a pixel level. Each scale parameter is computed only at the first iteration and is fixed during subsequent iterations. We compute the scale parameters using a median absolute deviation, which is the robust measure of a deviation [76]. At the end of the algorithm, we binarize the IRLS weight to generate an object mask. This procedure is applied to an amplitude and a phase image in the same manner, and thus we can obtain the object mask in each domain. In this study, we determine the final object mask as their intersection.

5.4 Experiments

We evaluated the effectiveness of the proposed method using real and synthetic data. First, we show the experiments with real data, and then, the applicability to various scenes is discussed using synthetic data. Finally, we investigate the effective measurement range of the proposed method. All experiments were done on Intel Core i5@3.1GHz with 8GB RAM and code was written in Python.



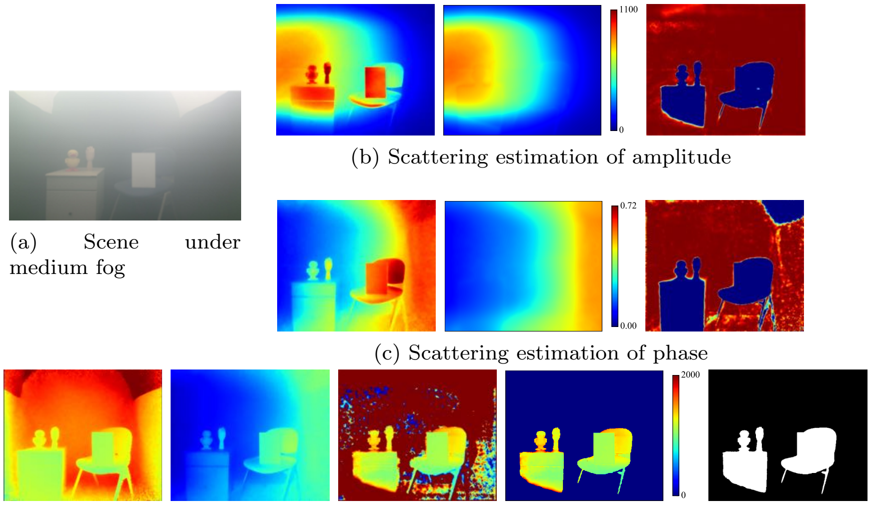
(a) Scene under thin fog

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.7: Results under thin fog. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.



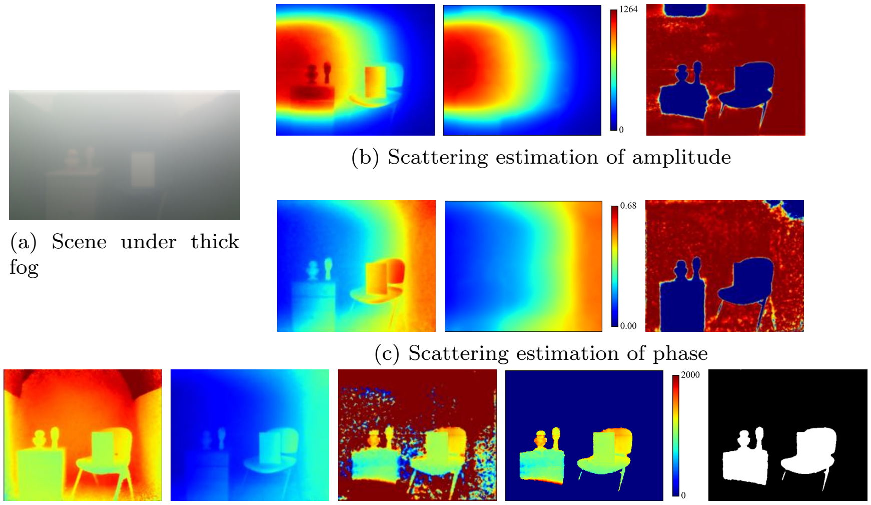
(a) Scene under medium fog

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.8: Results under medium fog. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.



(a) Scene under thick fog

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

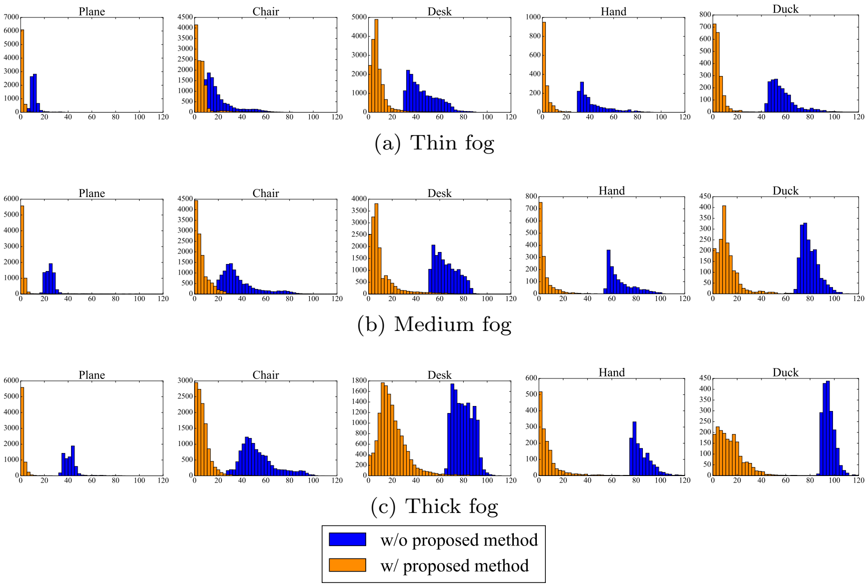
Figure 5.9: Results under thick fog. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.

5.4.1 Experiments with real data

First, we performed the experiments in a scene shown in Fig. 5.6. We set up a fog generator and a Kinect v2 in a closed space sized 186 × 161 cm with black walls and floor. This ToF camera has an IR camera and a light source inside it. This is a common configuration in ToF cameras. The light source emits modulated infrared light. Fog generated by the fog machine is made of liquid of propylene glycol. After generating fog, we waited for a few minutes so that fog is filled in the closed space and regarded as a homogeneous medium. The observation of the wall includes only a scattering component because incident light into the wall is absorbed. The Kinect v2 has three modulation frequencies: 120, 80,

Table 5.1: Mean / max depth error on each object of without considering scattering (top) and proposed (bottom) under different density conditions. [cm]

	Plane	Chair	Desk	Hand	Duck
Thin	11.7 / 64.4	19.9 / **83.1**	46.0 / 93.0	42.5 / 93.8	57.4 / 105.0
	1.8 / **47.7**	**6.5** / 89.4	**8.3** / **70.0**	**3.2** / **43.6**	**4.5** / **39.9**
Medium	25.3 / 97.3	37.2 / 98.2	65.6 / 92.1	67.9 / 102.5	79.8 / 111.0
	2.1 / **28.5**	**6.0** / **49.8**	**10.7** / **78.6**	**5.1** / **57.1**	**11.8** / **74.9**
Thick	42.1 / 105.5	53.1 / 110.6	79.9 / 110.4	84.4 / 123.2	95.4 / 121.9
	2.0 / **39.7**	**7.1** / **71.2**	**20.2** / **99.2**	**7.6** / **75.9**	**14.3** / **67.9**



(a) Thin fog

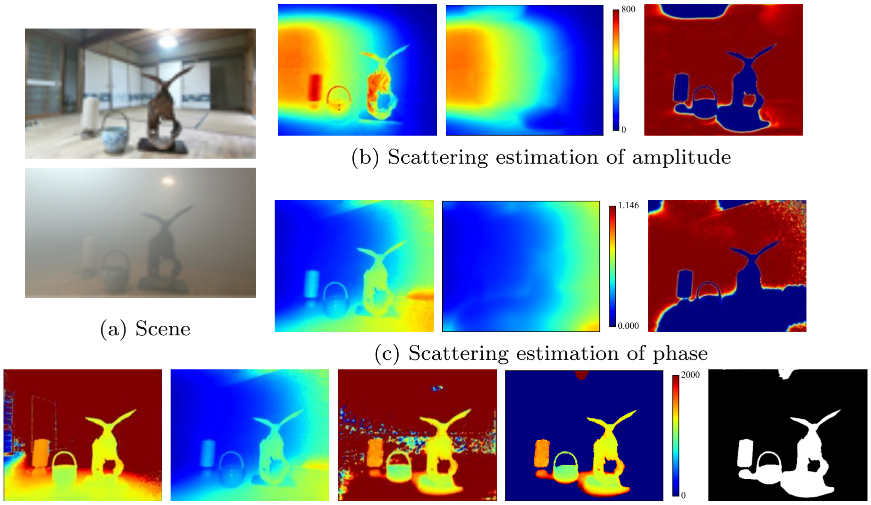
(b) Medium fog

(c) Thick fog

Figure 5.10: Error histograms of each object. (a)-(c) show error under thin, medium, and thick fog, respectively. Blue and orange bars represent error without and with proposed method.

and 16 MHz. We used images obtained with 16 MHz. Larger frequencies have shorter depth measurement range due to phase wrapping. The measurement capability of 16 MHz is about 9 meters, which is the largest measurement range of the frequencies of the Kinect v2. To acquire an amplitude and phase image, we used the source code given by [93]. Their code provides the average image of several frames, and we modified the code so that only a single frame was input. If we use multiple frames, the estimation will get to be more robust, while our method works well given only a single frame as shown in the experiments. To compensate for high frequency noise, we used a bilateral filter as preprocessing.

The spatial resolution of an image captured by Kinect v2 is 424 × 512 pixels. We divided a captured image into 4 × 4 patches ($K = 16$) for local quadratic prior. In Section 5.3.1, we assumed that the camera and the light source are collocated on a line that runs parallel to the horizontal axis of the image. In practice, the camera and light source in the Kinect v2 are slightly out of alignment, and it is difficult to modify the setup. Although this violates the symmetry of the scattering component, we found that error due to this misalignment is negligibly small as shown in Figs. 5.3(b)(c). In our implementation, we defined $\mathbf{F}$ as a matrix that flips an image with respect to the 200th row of the image. In addition, we did not use the 24 rows of the lower part of the image for the third term of Eq. (5.11), as these pixels have no information of global symmetrical prior. Just modifying the flip matrix $\mathbf{F}$ by changing the flip center is enough to estimate scattering components instead of modifying the hardware setup.



(a) Scene

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.11: Results of other real scene. (a) Target scene without and with fog. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.

For amplitude images, we set the hyperparameters of the objective function as $[\gamma_1', \gamma_2', \gamma_3'] = [0.1, 0.1, 10]$, and the tuning parameter of the function $\rho(x)$ is set as $c = 4, 7$ in the coarse and fine level optimization, respectively. For phase images, we set $[\gamma_1', \gamma_2', \gamma_3'] = [0.01, 0.1, 50]$ and $c = 2, 3$. The numbers of IRLS iterations were 5 and 50 for the coarse and fine optimization. One iteration required about from 0.3 to 1.0 seconds.

The results are shown in Figs. 5.7, 5.8, and 5.9. We tested the proposed method under different density conditions for investigating the robustness to the density (In Figs. 5.7, 5.8, and 5.9, the density is thin, medium, and thick, respectively). In each image, we show (a) the RGB image and (b)(c) the input image, the estimation of the scattering component, and object region for the amplitude and phase image. The object region depicted here is the IRLS weight before binarization. In (d), we show the depth without and with fog, the reconstructed depth, the masked depth, and the estimated object mask from left to right. The depth measurement in the foggy scene had large error here due to fog. On the other hand, the proposed method could estimate the scattering component and object region, and improve the depth measurement regardless of medium density. Of particular note is that thin regions such as the legs of the chair could be extracted. On the other hand, as shown in Fig. 5.9(b), some background pixels were regarded as outliers simply due to the fitting error of the quadratic function. Although the error will be suppressed by using smaller

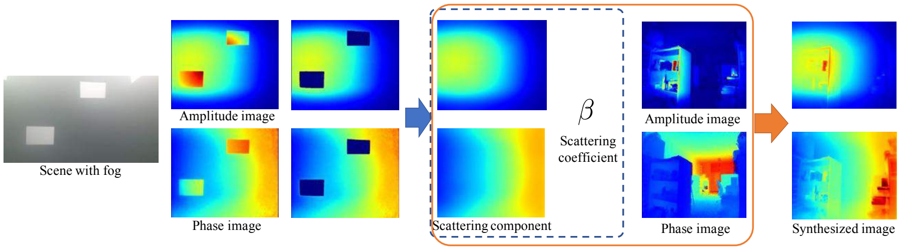


Figure 5.12: Procedure of synthesizing images. First, we captured scene that has calibration objects in foggy scene and masked region of calibration objects manually. After that, we compensated for defective region to estimate scattering component. Using observation without fog, scattering coefficient can be computed. Images of target scene without fog were captured separately, and attenuated direct component and estimated scattering component were combined into synthesized images.

patches for the fitting, this also lead to increase in the risk that the quadratic function is completely fitted to an object when the patch is enclosed by the object region. The mean and max depth error without considering scattering and with the proposed method under different density conditions is listed in Table 5.1; here, we define the ground truth as the measured depth without fog. The object label corresponds to that of Fig. 5.6(b). As shown, the proposed method could reduce the error significantly regardless of fog density. We also show the error histograms of each object in Fig. 5.10, where (a)-(c) show the error under thin, medium, and thick fog, respectively. The blue and orange bars represent the error without and with the proposed method. Under thick fog, there are more pixels with large error than under thin and medium fog. In each density, a few pixels have so large error even with the proposed method and they contain the max error shown in Table 5.1, while it is difficult to measure these pixels even in clear scenes due to sensor sensibility (e.g., object boundaries).

Next, we tested the proposed method in a scene shown in Fig. 5.11. We artificially generated fog in the same manner as the previous experiments, while the scene in Fig. 5.11 has neither dark walls nor floor. Note that the scene has materials with various types of reflectance, including a lamp made from paper, a glossy vase, and a wooden ornament. The estimation of the scattering component and object region for the amplitude and phase image is shown in Fig. 5.11(b)(c), respectively, and the result of the depth reconstruction is shown in Fig. 5.11(d). The proposed method could also extract the object region and improve the depth measurement in a scene that has a general background.

5.4.2 Experiments with synthesized data

To investigate the effectiveness in more varied scenes, we evaluated the proposed method with synthesized data. The procedure of generating the synthesized data is shown in Fig. 5.12. We assume that a scattering component does not depend on object depth, and thus we observed a direct component and a scattering component separately and then combined them into a synthesized image. First, we captured a foggy scene that includes calibration objects for the estimation of the scattering coefficient, and the region of the calibration objects was masked manually. After that, we compensated for the defective region by solving Eq. (5.12) to estimate the scattering component. In the scene in Fig. 5.12, a scattering coefficient was computed as $\beta = 3.5 \times 10^{-4}$ /mm. We also observed a scene without fog, which was used for the direct component after being attenuated by the scattering coefficient. We combined the attenuated signal and the scattering component to synthesize amplitude and phase images.

The results are shown in Figs. 5.13, 5.14, and 5.15. In each scene, we show (a) the target scene, (b)(c) the estimated scattering component and the IRLS weight for the amplitude and phase image, and (d) the result of the depth reconstruction. In Figs. (a)-(c), the proposed method effectively extracted the object region and estimated the scattering component. However, in Fig. 5.14, the center of the object region was regarded as the background. This is due to the quadratic function was partially fitted to the object region. This might be solved by introducing the spatial smoothness of the IRLS weight. We also show a failure case in Fig. 5.16. In a scene that has a large object region, our method was less effective because a quadratic function also fits to values in the object region. In Fig. 5.16, a large textureless object region exists on the left side. In addition, the global symmetrical prior did not work in this region because the object occupied the pixels from top to bottom in the image.

5.4.3 Discussion of measurement range

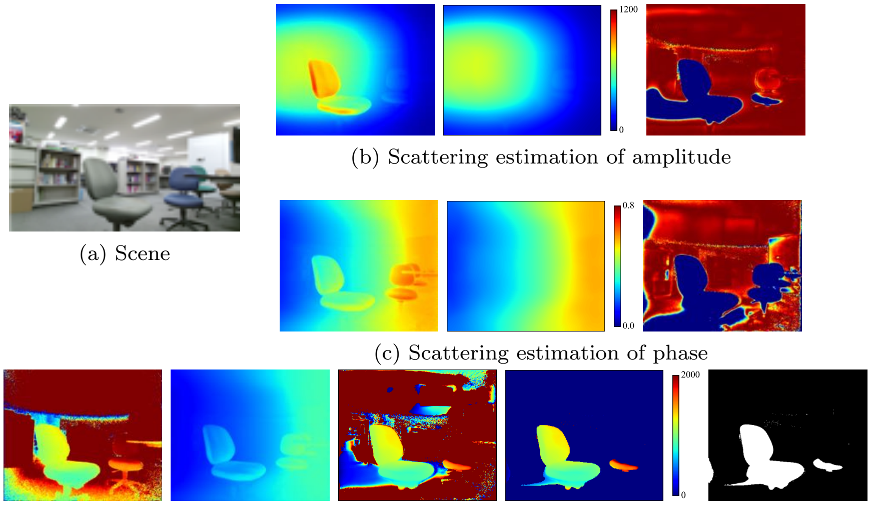
We assume that a scattering component is saturated close to a camera and there exists a background that has only a scattering component. In this section, we discuss the measurement capability of our method in this context.

As shown in Fig. 5.17, a scene has a saturation point and a background point denoted as $\mathbf{p}_{saturate}$ and $\mathbf{p}_{background}$. For simplicity, the camera and light source are assumed to be collocated in the same place. Far from $\mathbf{p}_{saturate}$, a scattering component is constant due to its saturation, while a direct component fades away far from $\mathbf{p}_{background}$. Therefore, the measurement range of our method is between $\mathbf{p}_{saturate}$ and $\mathbf{p}_{background}$.

We simulated the measurement range to evaluate the capability. Similarly to the process of synthesizing images, a scattering coefficient was computed for the scene in Fig. 5.8 to use for the simulation ($\beta = 3.2 \times 10^{-4}$ /mm). We use the Henyey-Greenstein phase function for scattering property (Eq. (2.10)). The parameter g was set as 0.9 for fog [13]. A scattering component from a camera to depth z is given as

$$\alpha_s(z)e^{j\varphi_s(z)} = \int_{z_0}^{z} \frac{1}{z^2}\beta P(\pi)e^{-2\beta z}e^{j\frac{4\pi f}{c}z}dz. \tag{5.17}$$

We set the starting point of the integral as $z_0 = 10$ mm. A direct component



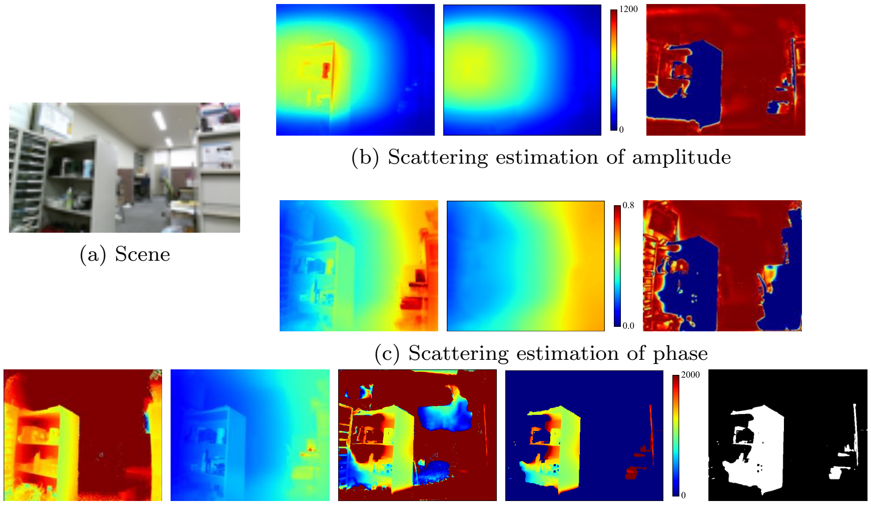
(a) Scene

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.13: Results of synthesized data. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.



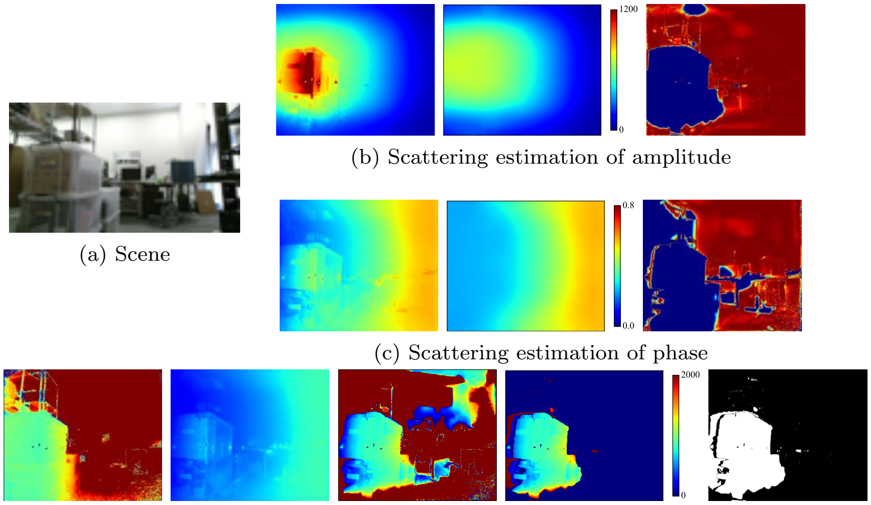
(a) Scene

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.14: Results of synthesized data. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.



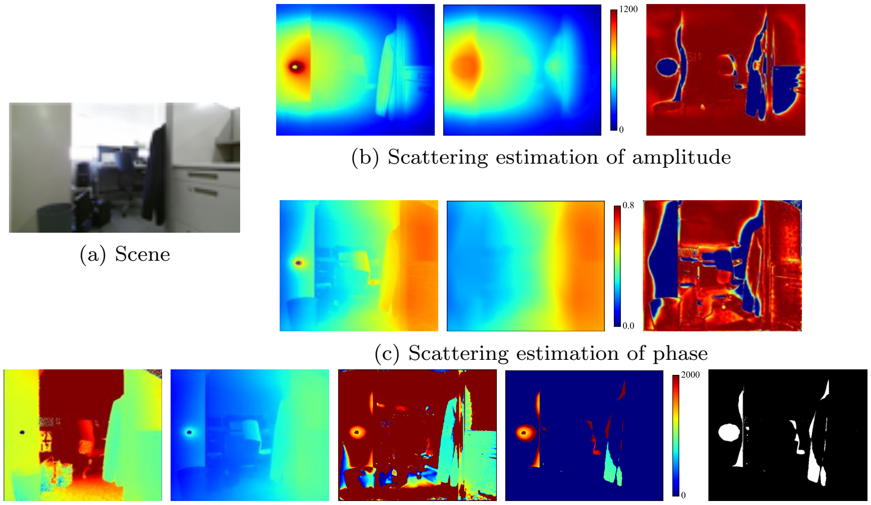
(b) Scattering estimation of amplitude

(a) Scene

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.15: Results of synthesized data. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.



(b) Scattering estimation of amplitude

(a) Scene

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Figure 5.16: Failure case. (a) Target scene. (b)(c) Left to right: input image, estimated scattering component, and IRLS weight for amplitude and phase image, respectively. (d) Left to right: depth without fog, depth with fog, reconstructed depth, masked reconstructed depth, and estimated object mask.

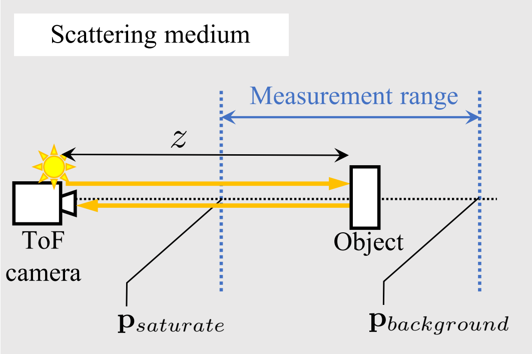


Figure 5.17: Simulation setting. Measurement range of our method is between $\mathbf{p}_{saturate}$ and $\mathbf{p}_{background}$ where scattering component is saturated and direct component remains.

from depth z is computed as

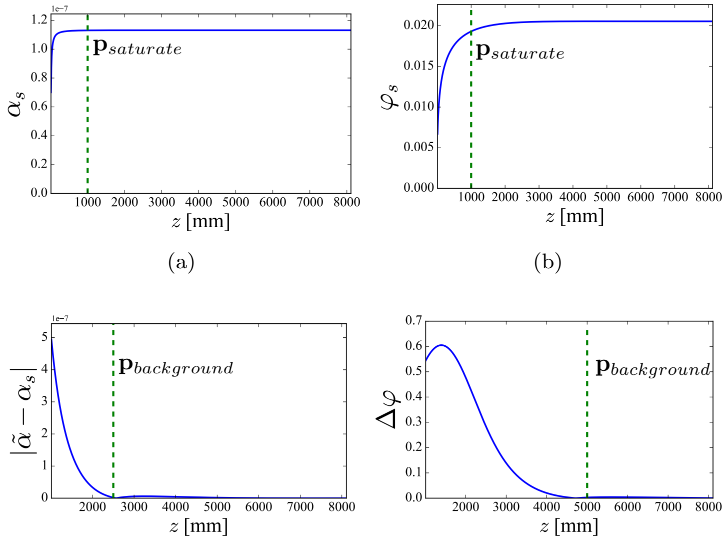
$$\alpha_d(z)e^{j\varphi_d(z)} = \frac{C}{z^2}e^{-2\beta z}e^{j\frac{4\pi f}{c}z}, \tag{5.18}$$

where C consists of a surface albedo and shading, and we set $C = 1$ in this simulation. The total observation $\tilde{\alpha}(z)e^{\tilde{\varphi}(z)}$ is the sum of these components as with Eq. (5.3).

Figure 5.18 shows the simulation results. In (a) and (b), the horizontal axis denotes depth z and the vertical axis denotes a scattering component for amplitude and phase. These figures validate the saturation characteristic. Meanwhile, in Fig. 5.18(c) and (d), the vertical axes denote the residual of the observation and scattering component. $\Delta\varphi$ is given by the residual angle of $\tilde{\alpha}(z)e^{j\varphi(z)}$ and $\alpha_s(z)e^{j\varphi_s(z)}$ on the complex plane. These values represent the remaining direct components from depth z.

Now, we can set $\|\mathbf{p}_{background}\| = 2500$ mm and 5000 mm for amplitude and phase from Fig. 5.18(c) and (d) because the direct component is close to zero. In contrast, in Fig. 5.18(a) and (b), if we set $\|\mathbf{p}_{saturate}\| = 1000$ mm, the estimation error of the scattering component for amplitude due to the saturation assumption can be considered almost zero because $1 - \alpha_s(1000)/\alpha_s(8000) \approx 0$, and for phase, the error is $1 - \varphi_s(1000)/\varphi_s(8000) \approx 6.0\%$.

In the experiments with real data, all of the target objects were located between 1000 mm and 2000 mm. If we assume the measurement range between $\|\mathbf{p}_{saturate}\| = 1000$ mm and $\|\mathbf{p}_{background}\| = 2500$ mm, the target objects are located in that range, and from above discussion, we have just 6.0% error of the scattering component estimation for phase due to the saturation assumption. As shown in the experiments, we can effectively reconstruct the object depth regardless of the error. The measurement range depends on the density of a participating medium, and it will get larger under thinner fog.



(a) (b)

(c) (d)

Figure 5.18: Simulation result. (a)(b) Scattering components for amplitude and phase observed at scene point whose depth is z. (c)(d) Residuals of observation and scattering component, which represent remaining direct components.

5.5 Conclusion

In this chapter, we discussed ToF-based depth measurement in scattering media. The proposed method simultaneously estimates an object region and depth with the observation of a continuous-wave ToF camera, which consists of an amplitude image and phase image. We modeled the effect of scattering media in amplitude and phase space. We leveraged the saturation of a scattering component and the attenuation of a direct component from a distant point in a scene. The formulation with a robust estimator and the IRLS optimization scheme allows us to estimate the scattering component and object region simultaneously. The limitation of the proposed method is that a scene is assumed to have a background region, which makes it difficult to apply the method to scenes filled with objects. This problem should be addressed in order to further enhance the real-world applicability of the proposed method.

Chapter 6

Conclusion

In this dissertation, we discussed 3D reconstruction in scattering media. Image degradation due to light scattering and attenuation in scattering media deteriorates the accuracy of traditional 3D reconstruction methods. Thus, image degradation should be taken into account when developing 3D reconstruction methods in scattering media. We divided the 3D reconstruction methods into three categories on the basis of their principles i.e., disparity-, shading-, and ToF-based methods. Each method was applied to scattering media with an appropriate scattering model.

In Chapter 2, the single scattering model and atmospheric scattering model, which are commonly used in computer vision and image processing, were discussed. The difference between these two models is the requirement of active light sources and thus this is a major factor to determine the scattering model. For example, the atmospheric scattering model can be used for conventional disparity-based methods that consist of only cameras. On the other hand, the single scattering model is suitable for shading- and ToF-based methods since they use avtive light sources.

Chapter 3 discussed MVS in scattring media as a disparity-basaed method. We used a learning-based MVS method, which takes a plane sweep volume as input to represent geometric constraints between multi-view images. Image degradation under the atmospheric scattering model depends on scene depth and thus we proposed the dehazing cost volume where input images are restored with the depth of a swept plane. The dehazing cost volume can model the image degradation and multi-view constraints simultaneously. We also proposed a method for estimating scattering parameters such as airlight and a scattering coefficient in the same framework. The output depth of the network with our dehazing cost volume can be regarded as a function of scattering parameters. Thus, these parameters are optimized so that the output depth map corresponds to a sparse depth map obtained at a SfM step. The experiments with actual foggy images demonstrated the effectiveness of the depth estimation in scattering media against an ordinary cost volume, particularly at distant regions with highly opacue haze.

Chapter 4 discussed photometric stereo in scattering media as a shading-based method. We used the single scattering model for modeling image degradation because photometric stereo requires multiple light sources. However, the analysis of shape-dependent forward scatter in the single scattering model is complicated. We proposed an analytical solution with lookup tables for the efficient computation of forward scatter. The effect of forwad scatter was then divided into a shape-dependent term and a global constant term for efficient image restoration. The proposed method was able to reconstruct the 3D shape of an object in highly turbid media, where the effect of forward scatter is not negligible.

Chapter 5 discussed depth measurement with a continuous-wave ToF camera as a ToF-based method in scattering media. Light scattering was modeled in amplitude and phase space as the image formation model of a continuous-wave ToF camera. Similar to RGB space, we exploited the assumption that backscatter is saturated close to a camera. We also assumed that a target scene consists of an object region and a background that only contains a scattering component, and then the formulation with a robust estimator and the IRLS optimization scheme enabled to estimate the scattering component and object region simultaneously. The experiments with Kinect v2 demonstrated the applicability of the proposed method in real foggy scenes.

The proposed methods rely on some assumptions about the physical phenomena of scattering media. We finally describe future work with the limitation of the current work.

Multiple scattering

As described in Chapter 2, multiple scattering is assumed to be negligibly small in the single scattering model. The analysis of this multiple scattering is often infeasible because we should take into account all multiple-scattered light that occurs at any aribirary point in a scene. Therefore, approximate models have been proposed for the analysis of the multiple scattering. In real situations, the multiple scattering causes a glow around light sources. For example, if a street lamp is observed by a camera under bad weather, the captured image contains the spread of glows around the lamp. Narasimhan and Nayar [13] proposed the atmospheric point spread function (PSF) for modeling such glows. This PSF-based model is similar to that of subsurface scattering, where multiple scattering beneath surfaces are modeled with PSF [67]. This model could be incorporated into the proposed 3D reconstruction methods for additionally modeling the effect of multiple scattering.

Inhomogenous or dynamic fluid media

This dissertation assumed that the density of scattering media is homogeneous. On the other hand, scattering media is often inhomogeneous or dynamically changing in the real world. A typical example is flowing water or smoke. However, it is difficult to model such inhomogeneous media with spatially-varying scattering and attenuation coefficients. Gu et al. [94] proposed a projector-camera system to reconstruct 3D volumes where each voxel contains the density of scattering media. The projector emits coded light patterns and the synchronized camera captures a scene at a high frame rate. The density volume is then reconstructed on the basis of compressive sensing. Satat et al. [85] uses a SPAD to observe photons bouncing off an object surface for depth measurement in inhomogeous scattering media. Instead of modeling light scattering explicitly, they demonstrated that scattered photons observed with a SPAD have gamma distribution. Wang et al. [95] combined a line sensor and line laser to generate a programmable light curtain. Light is adaptively emitted so that the sensor does not observes scattered light, and this enables depth measurement in inhomogeous scattering media. The use of these methods is however hindered due to the requirement of expensive sensors or special hardware settings.

As discussed in Chapter 2, most single image dehazing techniques also focus on homogeneous media with the atmospheric scattering model. Some attempts have been recently made to design dehazing methods for inhomogeneous scattering media [96, 97, 31]. However, these methods heavily rely on the representation ability of deep neural networks without physics-based models. Thus, it is left open to apply dehazing and 3D reconstruction methods to inhomogeneous scattering media.

Learning-based rendering or analysis

Light interaction in homogeneous or inhomogeneous scattering media including multiple scattering can be described with the radiative transfer equation (RTE) [98]. The single scattering model discussed in Chapter 2 is the specific case of the RTE. In computer graphics, scattering media such as clouds is rendered by solving the RTE. However, similar to the analysis of multiple scattering and inhomogeneous media, such rendering is also a challenging problem due to heavy computational cost in Monte Carlo integration. Recently, learning-based approaches have been proposed to achieve efficient and high-quality rendering. Kallweit et al. [99] replaced a multiple scattering term in the RTE with the output of a deep neural network. Learning-based approaches enable the analysis of complicated physical phenomena. There also have been some methods where scene geometry and reflectance is modeled with volume representation and images are rendered from the volume in learning framework [100, 101]. The simultaneous analysis of complicated scattering effect and scene geometry could be possible by combining these learning-based approaches.

Bibliography

[1] Y. Furukawa and C. Hernández. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1–2):1–148, 2015.

[2] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[3] J. L. Schönberger and J. M. Frahm. Structure-from-motion revisited. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[4] J. Geng. Structured-light 3d surface imaging: a turorial. 3(2):128–160, 2011.

[5] B. K. P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, 5:37–75, 1990.

[6] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):139–144, 1980.

[7] S. Xin, S. Nousias, K. N. Kutulakos, A. C. Sankaranarayanan, S. G. Narasimhan, and I. Gkioulekas. A theory of fermat paths for non-line-of-sight shape reconstruction. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6800–6809, 2019.

[8] C. Sakaridis, D. Dai, and L. V. Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[9] J. Huang, C. Lu, P. Chang, C. Huang, C. Hsu, Z. Ewe, P. Huang, and H. Wang. Cross-modal contrastive learning of representations for navigation using lightweight, low-cost millimeter wave radar for adverse environmental conditions. *arXiv preprint arXiv:2101.03525*, 2021.

[10] C. Tsiotsios, T.K. Kim, A.J. Davison, and S.G. Narasimhan. Model effectiveness prediction and system adaptation for photometric stereo in murky water. *Computer Vision and Image Understanding*, 150:126–138, 2016.

[11] B. Sun, R. Ramamoorthi, S. Narasimhan, and S. Nayar. A practical analytic single scattering model for real time rendering. *ACM Transaction on Graphics*, 24(3):1040–1049, 2005.

[12] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International Journal of Computer Vision*, 48(3):233–254, 2002.

[13] S. G. Narasimhan and S. K. Nayar. Shedding light on the weather. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 665–672, 2003.

[14] T. Treibitz and Y. Y. Schechner. Active polarization descattering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):385–399, 2009.

[15] C. Tsiotsios, M. E. Angelopoulou, T. Kim, and A. J. Davison. Backscatter compensated photometric stereo with 3 sources. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2259–2266, 2014.

[16] R. T. Tan. Visibility in bad weather from a single image. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[17] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011.

[18] K. Nishino, L. Kratz, and S. Lombardi. Bayesian defogging. *International Journal of Computer Vision*, 98(3):263–278, 2012.

[19] R. Fattal. Dehazing using color-lines. *ACM Transactions on Graphics (TOG)*, 34(1), 2014.

[20] D. Berman, T. Treibitz, and S. Avidan. Non-local image dehazing. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1674–1682, 2016.

[21] D. Berman, T. Treibitz, and S. Avidan. Air-light estimation using haze-lines. *The IEEE International Conference on Computational Photography (ICCP)*, 2017.

[22] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transaction on Image Processing*, 25(11):5187–5198, 2016.

[23] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M. Yang. Single image dehazing via multi-scale convolutional neural networks. *European Conference on Computer Vision (ECCV)*, pages 154–169, 2016.

[24] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. Aod-net: All-in-one dehazing network. *The IEEE International Conference on Computer Vision (ICCV)*, pages 4770–4778, 2017.

[25] H. Zhang and V. M. Patel. Densely connected pyramid dehazing network. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3194–3203, 2018.

[26] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M. H. Yang. Gated fusion network for single image dehazing. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3253–3261, 2018.

[27] X. Yang, Z. Xu, and J. Luo. Towards perceptual image dehazing by physics-based disentanglement and adversarial training. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[28] D. Yang and J. Sun. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. *The European Conference on Computer Vision (ECCV)*, pages 702–717, 2018.

[29] Y. Liu, J. Pan, J. Ren, and Z. Su. Learning deep priors for image dehazing. *The IEEE International Conference on Computer Vision (ICCV)*, pages 2492–2500, 2019.

[30] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia. Ffa-net: Feature fusion attention network for single image dehazing. *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 11908–11915, 2020.

[31] Q. Deng, Z. Huang, C. Tsai, and C. Lin. Hardgan: A haze-aware representation distillation gan for single image dehazing. *The European Conference on Computer Vision (ECCV)*, 2020.

[32] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *The European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[33] P. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. Huang. Deepmvs: Learning multi-view stereopsis. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2821–2830, 2018.

[34] S Im, H. Jeon, S. Lin, and I. S. Kweon. Dpsnet: End-to-end deep plane sweep stereo. *International Conference on Learning Representations (ICLR)*, 2019.

[35] K. Wang and S. Shen. Mvdepthnet: real-time multiview depth estimation neural network. *International Conference on 3D Vision (3DV)*, pages 248–257, 2018.

[36] R. T. Collins. A space-sweep approach to true multi-image matching. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 358–363, 1996.

[37] E. Zheng, E. Dunn, V. Jojic, and J Frahm. Patchmatch based joint view selection and depthmap estimation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1510–1517, 2014.

[38] J. L. Schönberger, E. Zheng, M. Pollefeys, and J. Frahm. Pixelwise view selection for unstructured multi-view stereo. *The European Conference on Computer Vision (ECCV)*, pages 501–518, 2016.

[39] L. Caraffa and J. Tarel. Stereo reconstruction and contrast restoration in daytime fog. *Asian Conference on Computer Vision (ACCV)*, pages 13–25, 2012.

[40] T. Song, Y. Kim, C. Oh, and K. Sohn. Deep network for simultaneous stereo matching and dehazing. *British Machine Vision Conference (BMVC)*, 2018.

[41] Z. Li, P. Tan, R. T. Tang, D. Zou, S. Z. Zhou, and L. Cheong. Simultaneous video defogging and stereo reconstruction. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4988–4997, 2015.

[42] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5038–5047, 2017.

[43] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. *The IEEE International Conference on Computer Vision (ICCV)*, pages 1625–1632, 2013.

[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. *The International Conference on Intelligent Robot Systems (IROS)*, 2012.

[45] S. Fuhrmann, F. Langguth, and M. Goesel. Mve: a multi-view reconstruction environment. *Eurographics Workshop on Graphics and Cultural Heritage*, pages 11–18, 2014.

[46] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015.

[47] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[48] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *Twenty-eighth Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[49] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6243–6252, 2017.

[50] S. G. Narasimhan, S. K. Nayar, B. Sun, and S. J. Koppal. Structured light in scattering media. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, I:420–427, 2005.

[51] Z. Murez, T. Treibitz, R. Ramamoorthi, and D. J. Kriegman. Photometric stereo in a scattering medium. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 39(9):1880–1891, 2017.

[52] S. Negahdaripour, H. Zhang, and X. Han. Investigation of photometric stereo method for 3-d shape recovery from underwater imagery. *OCEANS'02*, 2:1010–1017, 2002.

[53] K. Zhou, Q. Hou, M. Gong, J. Snyder, B. Guo, and H. Y. Shum. Fogshop: Real-time design and rendering of inhomogeneous, single-scattering media. *15th Pacific Conference on Computer Graphics and Applications*, pages 116–125, 2007.

[54] V. Pegoraro, M. Schott, and S. G. Parker. A closed-form solution to single scattering for general phase functions and light distributions. *Proceedings of the 21st Eurographics conference on Rendering*, pages 1365–1374, 2010.

[55] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, 2003.

[56] T. Papadhimitri and P. Favaro. Uncalibrated near-light photometric stereo. *Proceedings of the British Machine Vision Conference*, 2014.

[57] F. Logothetis, R. Mecca, and R. Cipolla. Semi-calibrated near field photometric stereo. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 941–950, 2017.

[58] K. Midorikawa, T. Yamasaki, and K. Aizawa. Uncalibrated photometric stereo by stepwise optimization using principal components of isotropic brdfs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4350–4358, 2016.

[59] H. Santo, M. Samejima, Y. Sugano, B. Shi, and Y. Matsushita. Deep photometric stereo network. *Proceedings of the IEEE International Conference on Computer Vision*, pages 501–509, 2017.

[60] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. *Asian Conference on Computer Vision*, pages 703–717, 2010.

[61] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa. Photometric stereo using sparse bayesian regression for general diffuse surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1816–1831, 2014.

[62] F. Logothetis, R. Mecca, Y. Quéau, and R. Cipolla. Near-field photometric stereo in ambient light. *Proceedings of the British Machine Vision Conference*, 2016.

[63] S. K. Nayar, K. Ikeuchi, and T. Kanade. Shape from interreflection. *International Journal of Computer Vision*, 6(3):173–195, 1991.

[64] M. Liao, X. Huang, and R. Yang. Interreflection removal for photometric stereo by using spectrum-dependent albedo. *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 686–696, 2011.

[65] G. Nam and M. H. Kim. Multispectral photometric stereo for acquiring high-fidelity surface normals. *IEEE Computer Graphics and Applications*, 34(6):57–68, 2014.

[66] B. Dong, K. D. Moore, W. Zhang, and P. Peers. Scattering parameters and surface normals from homogeneous translucent materials using photometric stereo. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2291–2298, 2014.

[67] C. Inoshita, Y. Mukaigawa, Y. Matsuthita, and Y. Yagi. Surface normal deconvolution: Photometric stereo for optically thick translucent objects. *Proceedings of the European conference on Computer Vision (ECCV)*, pages 346–359, 2014.

[68] A. Agrawal, R. Rasker, and R. Chellappa. What is the range of surface reconstructions from a gradient field? *Proceedings of the 9th European conference on Computer Vision*, I:578–591, 2006.

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

[70] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[71] S. G. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. K. Nayar, and H. W. Jensen. Acquiring scattering propoerties of participating media by dilution. *ACM Transaction on Graphics*, 25(3):1003–1012, 2006.

[72] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[73] Y. Asano, Y. Zheng, K. Nishino, and I. Sato. Shape from water: Bispectral light absorption for depth recovery. *European Conference on Computer Vision*, pages 635–649, 2016.

[74] A. Dancu, M. Fourgeaud, Z. Franjcic, and R. Avetisyan. Underwater reconstruction using depth sensors. *SIGGRAPH Asia 2014 Technical Briefs*, 2014.

[75] P.W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics – Theory and Method*, 6(9):813–827, 1977.

[76] J. Fox and S. Weisberg. Robust regression: Appendix to an r and s-plus companion to applied regression. 2002.

[77] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. *The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3869–3872, 2008.

[78] D. Wipf and S. Nagarajan. Iterative reweighted l1 and l2 methods for finding sparse solutions. *IEEE Journal on Selected Topics in Signal Processing*, 3(2):317–329, 2010.

[79] S. Fuchs. Multipath interference compensation in time-of-flight camera images. *2010 20th International Conference on Pattern Recognition*, pages 3583–3586, 2010.

[80] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt. Sra: Fast removal of general multipath for tof sensor. *European Conference on Computer Vision (ECCV)*, pages 234–249, 2014.

[81] N. Naik, A. Kasambi, C. Rhemann, S. Izadi, R. Rasker, and S. B. Kang. A light transport model for mitigating multipath interference in time-of-flight sensors. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–81, 2015.

[82] A. Kasambi, J. Schiel, and R. Rasker. Macroscopic interferometry: Rethinking depth estimation with frequency-domain time-of-flight. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 893–902, 2016.

[83] Q. Guo, I. Frosio, O. Gallo, T. Zickler, and J. Jautz. Tackling 3d tof artifacts through learning and the flat dataset. *The European Conference on Computer Vision (ECCV)*, pages 368–383, 2018.

[84] F. Heide, L. Xiao, A. Kolb, M. B. Hullin, and W. Heidrich. Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Optics Express*, 22(21):26338–26350, 2014.

[85] G. Satat, M. Tancik, and R. Rasker. Towards photography through realistic fog. *The IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, 2018.

[86] F. Heide, M. B. Hullin, J. Gregson, and W. Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (TOG)*, 32(4), 2013.

[87] T. Muraji, K. Tanaka, T. Funatomi, and Y. Mukaigawa. Depth from phasor distortions in fog. *Optics Express*, 27(13):18858–18868, 2019.

[88] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. *ACM Transaction on Graphics (TOG)*, 34(5), 2015.

[89] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[90] P. J. Huber. Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics*, 1(5):799–821, 1973.

[91] C. M. Bishop. *Pattern Recognition and Machine Intelligence.* Springer, 2006.

[92] K. N. Chaudhury and A. Singer. Non-local patch regression: Robust image denoising in patch space. *The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1345–1349, 2013.

[93] K. Tanaka, Y. Mukaigawa, T. Funatomi, H. Kubo, Y. Matsushita, and Y. Yagi. Material classification using frequency- and depth-dependent time-of-flight distortion. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 79–88, 2017.

[94] J. Gu, S. K. Nayar, P. N. Belhumeur, and R. Ramamoorthi. Compressive structured light for reconvering inhomogeneous partiicpating media. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):555–567, 2013.

[95] J. Wang, J. Bartels, W. Whittaker, A. C. Sankaranarayanan, and S. G. Narasimhan. Programmable triangulation light curtains. *The European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[96] J. Liu, H. Wu, Y. Xie, Y. Qu, and L. Ma. Trident dehazing network. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[97] H. wu, J. Liu, Y. Xie, Y. Qu, and L. Ma. Knowledge transfer dehazing network for nonhomogeneous dehazing. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[98] S. Chandrasekhar. *Radiative transfer*. Dover Publications, 1960.

[99] S. Kallweit, T. Müller, B. Mcwilliams, M. Gross, and J. Novák. Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.

[100] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4), 2019.

[101] S. Bi, Z. Xu, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. *The European Conference on Computer Vision (ECCV)*, pages 294–311, 2020.

List of Publications

Journal Articles

1. Yuki Fujimura, Motoharu Sonogashira, Masaaki Iiyama, "Dehazing Cost Volume for Deep Multi-view Stereo in Scattering Media with Airlight and Scattering Coefficient Estimation," submitted to Computer Vision and Image Understanding, 2020.

2. Yuki Fujimura, Motoharu Sonogashira, Masaaki Iiyama, "Simultaneous Estimation of Object Region and Depth in Participating Media Using a ToF Camera," IEICE Transactions on Information and Systems, Vol. E103-D, No. 3, pp.660-673, 2020.

3. Yuki Fujimura, Masaaki Iiyama, Atshushi Hashimoto, Michihiko Minoh, "Photometric Stereo in Participating Media Using an Analytical Solution for Shape-Dependent Forward Scatter," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, No. 3, pp. 708-719, 2020.

Refereed Conference Presentations

1. Yuki Fujimura, Motoharu Sonogashira, Masaaki Iiyama, "Dehazing Cost Volume for Deep Multi-view Stereo in Scattering Media," Asian Conference on Computer Vision (ACCV), 2020.

2. Yuki Fujimura, Motoharu Sonogashira, Masaaki Iiyama, "Dehazing Cost Volume for Deep Multi-view Stereo in Scattering Media," The 23rd Meeting on Image Recognition and Understanding (MIRU), 2020.

3. Yuki Fujimura, Motoharu Sonogashira, Masaaki Iiyama, "Defogging Kinect: Simultaneous Estimation of Object Region and Depth in Foggy Scenes," The 22nd Meeting on Image Recognition and Understanding (MIRU), 2019.

4. Yuki Fujimura, Masaaki Iiyama, Atsushi Hashimoto, Michihiko Minoh, "Photometric Stereo in Participating Media Considering Shape-Dependent Forward Scatter," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.7445-7453, 2018.

5. Yuki Fujimura, Masaaki Iiyama, Atsushi Hashimoto, Michihiko Minoh, “Shape reconstruction of objects in participating media by combining photometric stereo and optical thickness,” Computer Vision for Analysis of Underwater Imagery (CVAUI), 2016 ICPR 2nd Workshop on, pp.49-54, 2016.

Conference Presentations

1. 住江 祐哉, 藤村 友貴, 薗頭 元春, 飯山 将晃, “学習ベースの重み付き最小二乗法による散乱成分推定”, 情報処理学会 コンピュータビジョンとイメージメディア研究会 （CVIM）, Vol. 2021-CVIM-225, No. 44, 2021.

2. 藤村 友貴, 薗頭 元春, 飯山 将晃, “未知散乱条件下での深層学習による Multi-view Stereo”, 情報処理学会 コンピュータビジョンとイメージメディア研究会 （CVIM）, Vol. 2020-CVIM-223, No. 29, 2020.

3. 藤村 友貴, 薗頭 元春, 飯山 将晃, “散乱媒体下での Multi-view Stereo のための Dehazing Cost Volume の提案”, 情報処理学会 コンピュータビジョンとイメージメディア研究会 （CVIM）, Vol. 2019-CVIM-219, No. 3, 2019.

4. 喜島 大揮, 藤村 友貴, 薗頭 元春, 飯山 将晃, “近接光源下で撮影された画像からの散乱除去と深度推定”, 2019 年電子情報通信学会総合大会, D-12-45, 2019.

5. 藤村 友貴, 薗頭 元春, 飯山 将晃, “ToF カメラを用いた散乱媒体下での物体領域と奥行きの同時推定”, 情報処理学会 コンピュータビジョンとイメージメディア研究会 （CVIM）, Vol. 2018-CVIM-214, No. 16, 2018.

6. 藤村 友貴, 飯山 将晃, 橋本 敦史, 美濃 導彦, “形状に依存する前方散乱を考慮した散乱媒体下での照度差ステレオ法”, 情報処理学会 コンピュータビジョンとイメージメディア研究会 （CVIM）, Vol. 2017-CVIM-209, No. 2, 2017.

7. 藤村 友貴, 飯山 将晃, 舩冨 卓哉, 橋本 敦史, 美濃 導彦, “散乱光を用いた形状計測のためのレーザー照射位置決定”, 2016 年電子情報通信学会総合大会, D-12-53, 2016.