Vol.2021-NL-248 No.9
Vol.2021-CVIM-226 No.9
2021/5/20

3D Reconstruction in Scattering Media

Fujimura Yuki[1,a)] Sonogashira Motoharu[1] Hashimoto Atsushi[2] Iiyama Masaaki[1]
Minoh Michihiko[3]

Abstract: This paper discusses three-dimensional (3D) reconstruction in scattering media. 3D reconstruction from two-dimensional images is important in computer vision. However, images captured in scattering media, such as fog or murky water, degrade due to light scattering and attenuation caused by suspended particles. Conventional 3D reconstruction methods are affected by the image degradation in scattering media. This paper 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-based method. Each method is extended for scattering media with an appropriate physics-based scattering model. The effectiveness of the proposed methods is evaluated on real data captured in foggy scenes and underwater.

Keywords: 3D reconstruction, scattering media, single scattering

1. Introduction

In the field of computer vision, tasks to obtain three-dimensional (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. 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 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 paper 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 paper, we present image formation models with such degradation and propose methods to enable 3D reconstruction in scattering media.

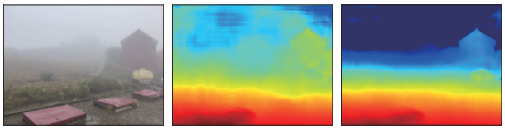
First of all, we overview existing 3D reconstruction methods. We divide the 3D reconstruction methods into three categories on the basis of their principles: **Disparity-based methods** use multiple cameras to capture multiple 2D images, which are taken as input for the methods. If a system consists of two or more than three cameras, it is called stereo or multi-view stereo (MVS) [8],

Fig. 1 Image captured under bad weather

respectively. The principle of 3D reconstruction of these methods is triangulation, i.e., if the positional relationship between camreas is known, corresponding pixels computed on the basis of feature matching yield 3D points. **Shading-based methods** 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. Intuitively, the observed image is brightest when the surface normal is parallel to the lighting direction. Photometric stereo [32] estimates surface normals from multiple images captured under different lighting conditions. **Time-of-Flight (ToF) -based methods** use special sensors that are designed to measure scene depth. A light source within the ToF sensor emits a signal into a target scene and the sensor receives the reflected light. The scene depth can be computed using the time difference between the emitted and received signals. Recently, off-the-shelf ToF cameras such as the Microsoft Kinect for Windows v2 (Kinect v2) are available at a low cost.

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

[1] Kyoto University
[2] OMRON SINIC X Corporation
[3] RIKEN
[a)] fujimura.yuki.i19@kyoto-u.jp



(a) (b) (c)

Fig. 2 Estimated depth in scattering media. (a) Image captured in actual foggy scene. (b) Output depth of fine-tuned MVDepthNet [31] with ordinary cost volume. (c) Output depth of network with our dehazing cost volume.

in scattering media depending on an applied 3D reconstruction method.

In summary, this paper discusses three methods, i.e., disparity-, shading-, and ToF-based methods, with physics-based scattering models for the application in scattering media. As a disparity-based method, we propose MVS in scattering media, where the system consists of only cameras. The atmospheric scattering model [18], which is a simple linear model, can be used as the image formation model for such a camera-only system. As a shading-based method, we propose photometric stereo in scattering media, which requires multiple active light sources. The single scattering model [26] can be adopted to describe the observation under active light sources in scattering media. As a ToF-based method, we propose depth measurement with a coninuous-wave ToF camera in scattering media. The single scattering model is also used for the image formation model, while the observation is represented in amplitude and phase space.

2. Multi-view stereo in scattering media

In this section, we discuss MVS in scattering media as a disparity-based 3D reconstruction method. MVS methods [8] 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 [12], [13], [34]. The proposed method is based on MVDepthNet [31], 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 [5], 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. However, an image captured in scattering media degrades; thus, using the ordinary cost volume leads to undesirable results, as shown in Fig. 2(b).

To solve this problem, we propose a novel cost volume for scattering media, called *the dehazing cost volume*. Degradation due to a scattering medium depends on the scene depth, and 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. 2(c).

2.1 Related work

There have been several works for applying disparity-based methods to scattering media, e.g., binocular stereo methods in scattering media [3], [24] The most related method to ours is the MVS method proposed by Li et al. [14]. These previous studies [3], [14] 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.

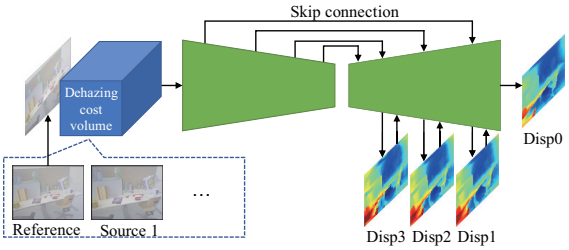


Fig. 3 Overview of MVS in scattering media. Input of network is reference image captured in scattering medium and our dehazing cost volume. Our dehazing cost volume is constructed from reference image and source images. Network architecture of our method is same as that of MVDepthNet [31], which has encoder-decoder with skip connections. Output of network is disparity maps (inverse depth maps) at different resolutions.

2.2 MVS with dehazing cost volume in scattering media

We first overview the proposed method then discuss the ordinary cost volume and our dehazing cost volume, followed by implementation details.

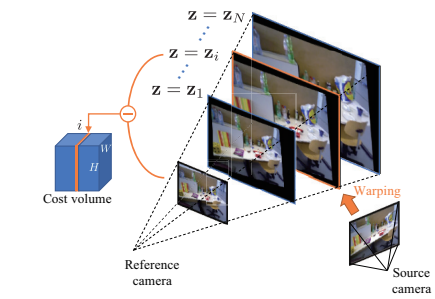
2.2.1 Overview

The proposed method is formulated as depth-map estimation, i.e., given multiple cameras, we estimate a depth 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.

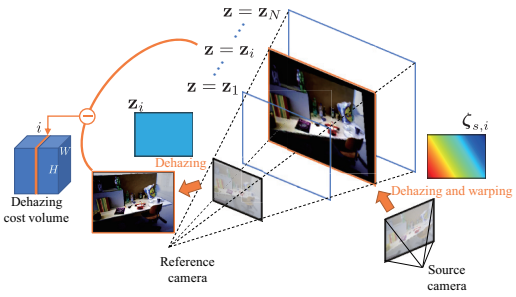
An overview of the proposed method is shown in Fig. 3. Our dehazing cost volume is constructed from a hazy reference image and source images captured in a scattering medium. The network takes the reference image and our dehazing cost volume as input then outputs a disparity map (inverse depth map) of the reference image. The network architecture is the same as that of MVDepthNet [31], while the ordinary cost volume used in MVDepthNet is replaced with our dehazing cost volume for scattering media.

2.2.2 Dehazing cost volume

Before explaining our dehazing cost volume, we show the computation of the ordinary cost volume in Fig. 4(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



(a) Cost volume



(b) Dehazing cost volume

Fig. 4 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.

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:

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

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 camera parameters and given depth. 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.

An image captured under overcast sky illumination in foggy scenes can be modeled with the atmospheric scattering model [18]. 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. The atmospheric scattering model is given by

$$I(u, v) = J(u, v) e^{-\beta z(u,v)} + \mathbf{A}(1 - e^{-\beta z(u,v)}), \qquad (2)$$

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. (1).

Figure 4(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

$$\mathcal{D}(u, v, i) = \frac{1}{S} \sum_{s} \| J_r(u, v; z_i) - J_s(\pi_{r \to s}(u, v; z_i)) \|_1, \qquad (3)$$

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. (2), if A and β are estimated beforehand, they are computed as follows:

$$J_r(u, v; z_i) = \frac{I_r(u, v) - \mathbf{A}}{e^{-\beta z_i}} + \mathbf{A}, \qquad (4)$$

$$J_s(\pi_{r \to s}(u, v; z_i)) = \frac{I_s(\pi_{r \to s}(u, v; z_i)) - \mathbf{A}}{e^{-\beta \zeta_{s,i}(\pi_{r \to s}(u, v; z_i))}} + \mathbf{A}. \qquad (5)$$

As shown in Fig. 4(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. 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 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) becomes very large. To avoid such cases, we revise the definition of our dehazing cost volume as follows:

$$\mathcal{D}(u, v, i) = \frac{1}{S} \sum_{s} \begin{cases} \| J_r(u, v; z_i) - J_s(\pi_{r \to s}(u, v; z_i)) \|_1 \\ \quad if\ 0 \le J_r^c(u, v; z_i) \le 1\ and \\ \quad 0 \le J_s^c(\pi_{r \to s}(u, v; z_i)) \le 1 \quad c \in \{r, g, b\} \\ \gamma \quad otherwise, \end{cases} \qquad (6)$$

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. (1) when the pixel value of each color channel is within 0 and 1.

Figure 5(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.

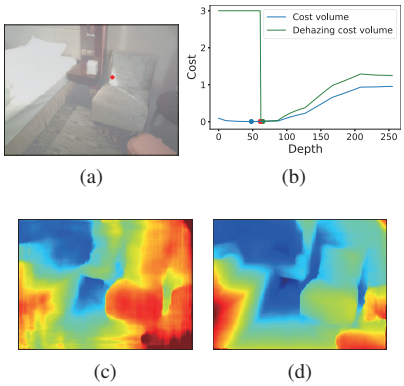


Fig. 5 Visualization of our dehazing cost volume. (b) Computed ordinary cost volume and our dehazing cost volume at red point in (a). In (b), red dot indicates location of ground-truth, and blue and green dots indicate minimum value of ordinary cost volume and our dehazing cost volume, respectively. (c) and (d) Output depth of MVDepthNet [31] with ordinary cost volume and our dehazing cost volume, respectively.

Table 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

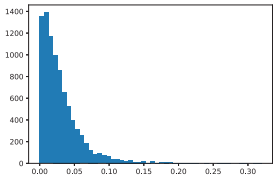


Fig. 6 Error histogram of our airlight estimator on synthesized test dataset.

2.2.3 Loss function

As shown in Fig. 3, a network takes a reference image and our dehazing cost volume as input. The network architecture is the same as that of MVDepthNet [31]. 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.

2.3 Scattering parameter estimation

As mentioned in Section 2.2.2, our dehazing cost volume requires scattering parameters, airlight A and a scattering coefficient β in Eq. (5). This section discusses the simultaneous estimation of the scattering parameters and depth with our dehazing cost volume.

2.3.1 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 1. It takes a single RGB image as input

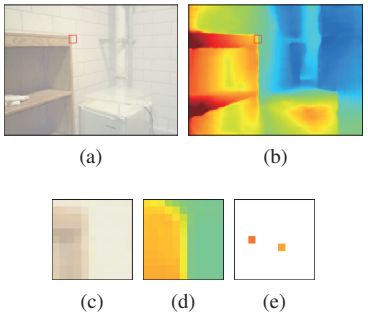


Fig. 7 Consideration of depth discontinuities. (a) Input image. (b) Output depth with ground-truth scattering parameters. Depth discontinuities exist in red boxed region. Zoom of regions in (a) and (b) are shown in (c) and (d), respectively. (e) Depth map of sparse 3D point cloud obtained by SfM in this 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).

to yield 1D output A. For training and test, we used the synthesized image dataset described in Section 2.4.1. Figure 6 shows the L1 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.

2.3.2 Difficulty of estimating scattering coefficient β

In contrast to A, it is difficult to estimate β from a single image. As shown in Eq. (2), 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 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. [14] 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. (2). 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.

2.3.3 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. [14], 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 re-

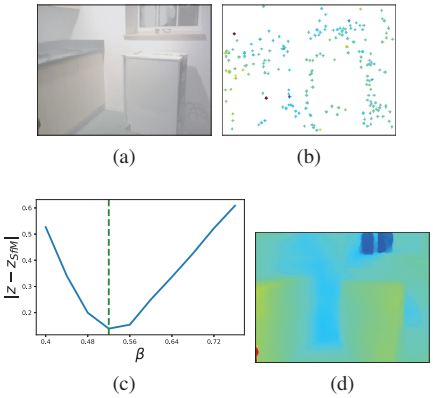


Fig. 8 Example of parameter search. (a) Input image. (b) Sparse depth map obtained by SfM. (c) Error plot with respect to β. (d) Final output depth.

garded 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 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:

$$A^*,\beta^* = \underset{A,\beta}{\operatorname{argmin}} \sum_{u,v} m(u,v)\rho\Big(z_{sfm}(u,v), z_{A,\beta}(u,v)\Big), \qquad (7)$$

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. (7) 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 [14], 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 values:

$$\rho\Big(z_{sfm}(u,v), z_{A,\beta}(u,v)\Big) = \min \left\{ \begin{array}{l} |z_{sfm}(u,v) - z_{A,\beta}(u,v)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u+\delta,v)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u-\delta,v)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u,v+\delta)|, \\ |z_{sfm}(u,v) - z_{A,\beta}(u,v-\delta)| \end{array} \right\}. \qquad (8)$$

As shown in Fig. 7, 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. (8), 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.

2.3.4 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 search to find the best solution. Figure 8 shows an example in which we search for β under ground-truth A. Figure 8(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. (7) with respect to each β. The green dashed line, which represents the ground-truth β, corresponds to the global minimum. Figure 8(d) shows the final output depth of the network with this global optimal solution.

As discussed in Section 2.3.1, 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:

$$\beta_0 = \underset{\beta\in[\beta_{min},\beta_{max}]}{\operatorname{argmin}} \sum_{u,v} m(u,v)\rho\Big(z_{sfm}(u,v), z_{A_0,\beta}(u,v)\Big). \qquad (9)$$

We then search for A^* and β^* that satisfy Eq. (7) 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.

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 $\mathcal{G}$, β_{min}, β_{max},Δ_A, Δ_β, and $\mathbf{z}_{sfm}$

Ensure: A^*, β^*, $\mathbf{z}^*$

$A_0 \leftarrow \mathcal{G}(I_r)$

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

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 $\Omega_\beta = [\beta_0 - \Delta_\beta, \beta_0 + \Delta_\beta]$

$\mathbf{z}^* \leftarrow \mathcal{F}(A^*,\beta^*; I_r, \{I_1,\cdots,I_S\})$

Table 2 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_β).

Method	L1-rel	L1-inv	sc-inv	C.P. (%)	MAE_A	MAE_β
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

2.4 Experiments

In this study, we used MVDepthNet [31] 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 [13], 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.

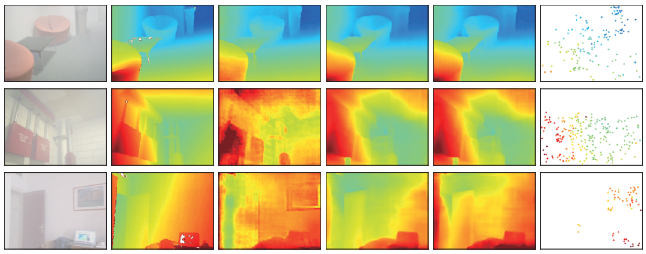
2.4.1 Dataset

Training: We used the DeMoN dataset [29] for training. This dataset consists of the SUN3D [33], RGB-D SLAM [25], and MVS datasets [7], which have sequences of real images. The DeMoN dataset also has the Scenes11 dataset [4], [29], 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 samples for training. 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. (2). 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 $\beta \in [-\log(0.4)/z_{med}, -\log(0.2)/z_{med}]$.

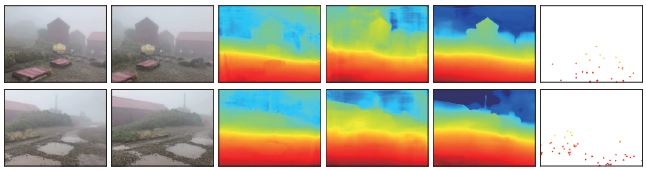
Similar to Wang and Shen [31], we adopted data augmentation to enable the network to reconstruct a wide depth range. The depth of each sample was scaled by a factor between 0.5 and 1.5 together with the translation vector of the camera. Note that when training the proposed method, β should also be scaled by the inverse of the scale factor.

Test: Each sample of the training 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 [33], 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 [31]. 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 [22], [23] 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 rescaled to match the scale of the ground-truth depth, and we used the camera parameters of the original SUN3D dataset.



(a) (b) (c) (d) (e) (f)

Fig. 9 Results with synthesized data. (a) Hazy input, (b) ground-truth depth, (c) DPSNet [13], (d) proposed method with ground-truth scattering parameters, (e) proposed method with scattering parameter estimation, and (f) sparse depth obtained by SfM.



(a) (b) (c) (d) (e) (f)

Fig. 10 Experimental results on our captured data in actual foggy scenes. (a) input reference image, (b) input source image, (c) output of DPSNet [13], (d) output of fine-tuned MVDepthNet [31], (e) output of proposed method with scattering parameter estimation, and (f) sparse depth obtained by SfM.

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.

2.4.2 Results

Table 2 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_β). 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.

The qualitative results of the following depth estimation after scattering parameter estimation are shown in Fig. 9. Figure 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.

2.4.3 Experiments with actual foggy scenes

We captured a video with a smartphone camera in an actual foggy scene. We applied the SfM method [22], [23] to all 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. 10. Figures (a) and (b) show the input reference and source images, respectively. This results indicate that the proposed method can reconstruct distant regions with large image degradation due to light scattering.

2.5 Conclusion

In this section, 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.

3. Photometric stereo in scattering media

This section discusses photometric stereo in scattering media as a shading-based method as shown in Fig. 12. Photometric stereo methods are an effective approach for reconstructing a 3D shape in scattering media [16], [19], [28]. Figure 11 shows a capture setting with a single camera and light source under the single scattering model [26]. 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. [19] modeled single backscattering under a directional light source in scattering media and estimated surface normals using a nonlinear optimization technique. Tsiotsios et al. [28] 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. [16] proposed a photometric stereo technique that considers forward scatter, they assumed that the scene is approximated as a plane, which enables prior

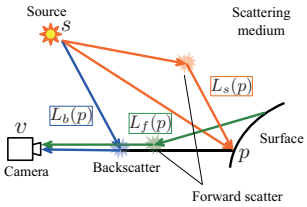


Fig. 11 Single scattering model. Observed irradiance at camera includes direct component reflected on surface, and both backscatter and forward scatter components.

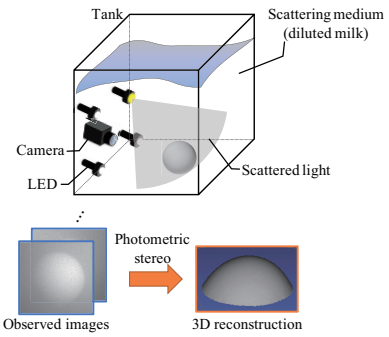


Fig. 12 Photometric stereo in scattering media

calibration of forward scatter.

We propose a forward scatter model and implement the model into a photometric stereo framework. Differing from the previous study [16], 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.

The single scattering model is more complicated model than the atmospheric scattering model used in the previous section. We thus propose an effective method for computing forward scatter with an analytical form of single scattering. In computer graphics, Sun et al. [26] 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 to overcome computational complexity issues. 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.

3.1 Analytical form of single scattering model

First of all, we provide an analytical form of the single scattering model using lookup tables. Sun et al. [26] assumed single and isotropic scattering and used 2D lookup tables to analytically describe backscatter and source-camera forward scatter to overcome computational complexity issues. We also use a lookup table similar to that of Sun et al. [26], while we additionally 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. 11, $L(p)$ 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:

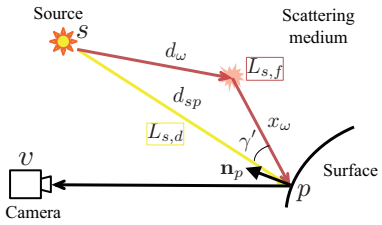


Fig. 13 Reflected component $L_s(p)$ (yellow arrow) consists of direct component $L_{s,d}(p)$ (red arrow) and source-surface forward scatter component $L_{s,f}(p)$. Direct component reaches surface directly from light source. Source-surface forward scatter is reflected component whose incident light reaches surface via forward scatter.

$$L(p) = L_s(p)e^{-\sigma d_{vp}} + L_b(p) + L_f(p). \tag{10}$$

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 shown in Fig. 13, 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),

$$L_s(p) = L_{s,d}(p) + L_{s,f}(p). \tag{11}$$

Thus, the observed irradiance is written as follows:

$$L_s(p) = \Big(L_{s,d}(p) + L_{s,f}(p)\Big)e^{-\sigma d_{vp}} + L_b(p) + L_f(p). \tag{12}$$

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

3.1.1 Direct component

The direct component reaches the surface directly from the source as shown in Fig. 13. Considering diffuse reflection and attenuation in scattering media, $L_{s,d}(p)$ is expressed as follows:

$$L_{s,d}(p) = \frac{I_0}{d_{sp}^2}e^{-T_{sp}}\rho_p \mathbf{n}_p^\top \mathbf{l}_{sp}, \tag{13}$$

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} denotes the product of σ and distance d_{xy}.

3.1.2 Backscatter component

Figure 14 shows the observation of the backscatter component. The backscatter component is the sum of scattered light on the viewline without reaching the surface as follows:

$$L_b(p) = \int_0^{d_{vp}} \frac{I_0}{d^2}\beta P(\theta)e^{-\sigma(x+d)}dx, \tag{14}$$

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. (14) cannot be computed in closed-form, an analytical solution can be acquired using a lookup table. Sun et al. [26] assumed isotropic scattering (i.e., $P(\theta) = 1/4\pi$) and derived an analytical solution using a 2D lookup table $F(u, v)$:

$$L_b(p) = I_0 H_0(T_{sv}, \gamma)\left[F(H_1(T_{sv}, \gamma), H_2(T_{vp}, T_{sv}, \gamma)) - F(H_1(T_{sv}, \gamma), \frac{\gamma}{2})\right], \tag{15}$$

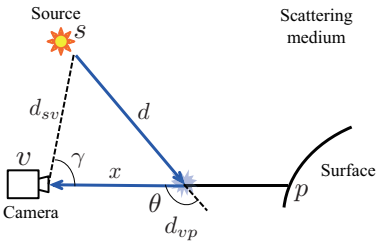


Fig. 14 Backscatter component is sum of scattered light on viewline without reaching surface.

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

$$H_0(T_{sv}, \gamma) = \frac{\beta\sigma e^{-T_{sv}\cos\gamma}}{2\pi T_{sv}\sin\gamma}, \tag{16}$$

$$H_1(T_{sv}, \gamma) = T_{sv}\sin\gamma, \tag{17}$$

$$H_2(T_{vp}, T_{sv}, \gamma) = \frac{\pi}{4} + \frac{1}{2}\arctan\frac{T_{vp} - T_{sv}\cos\gamma}{T_{sv}\sin\gamma}. \tag{18}$$

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

3.1.3 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. 13). This component is the integral of scattered light on a hemisphere centered on p:

$$L_{s,f}(p) = \int_{\Omega_{2\pi}} L_b(\omega)\rho_p \mathbf{n}_p^\top \mathbf{l}_\omega d\omega, \tag{19}$$

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

$$L_b(\omega) = \int_0^\infty \frac{I_0}{d_\omega^2}\beta P(\theta)e^{-\sigma(x_\omega + d_\omega)}dx_\omega, \tag{20}$$

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 3.1.2, Sun et al. [26] derived an analytical solution using a 2D lookup table as follows:

$$L_{s,f}(p) = \frac{\beta\sigma I_0\rho_p}{2\pi T_{sp}}G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}), \tag{21}$$

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

$$G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) = \int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma'}}{\sin\gamma'}\left[F(H_1(T_{sp}, \gamma'), \frac{\pi}{2}) - F(H_1(T_{sp}, \gamma'), \frac{\gamma'}{2})\right]\mathbf{n}_p^\top \mathbf{l}_\omega d\omega, \tag{22}$$

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

3.1.4 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. 15). In this study, we describe this component analytically using a lookup table.

As shown in Fig. 15, irradiance at the camera includes reflected light from the small facet centered at q. If we consider this small

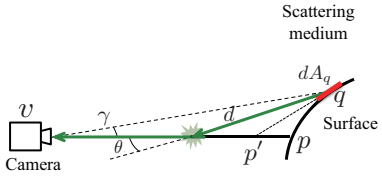


Fig. 15 Surface-camera forward scatter component. When we observe surface point p in scattering media, light reflected on point q is scattered on viewline, and scattered light is also observed.

facet as a virtual light source, similar to Eq. (14), the irradiance can be expressed as follows:

$$\int_0^{d_{vp'}} \frac{L_s(q)dA_q}{d^2}\beta P(\theta)e^{-\sigma(x+d)}dx, \tag{23}$$

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:

$$L_f(p) = \sum_{q\neq p}\int_0^{d_{vp'}} \frac{L_s(q)dA_q}{d^2}\beta P(\theta)e^{-\sigma(x+d)}dx. \tag{24}$$

Note that the domain of integration $[0, d_{vp'}]$ differs from that of Eq. (14), 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. (15), the isotropic scattering assumption yields the following:

$$L_f(p) = \sum_{q\neq p} L_s(q)dA_q H_0(T_{vq},\gamma) \left[F(H_1(T_{vq},\gamma), H_2(T_{vp'},T_{vq},\gamma)) - F(H_1(T_{vq},\gamma),\frac{\gamma}{2})\right]. \tag{25}$$

This is the analytical expression of the surface-camera forward scatter.

3.2 Photometric stereo considering shape-dependent forward scatter

In Section 3.1, we model the image formation in scattering media using four components in Eq. (12). To reconstruct surface normals using photometric stereo, we must restore the direct component $L_{s,d}(p)$. In this section, we first explain the compensation of the backscatter component [28]. Then, we discuss how to remove the surface-camera forward scatter. Finally, we explain photometric stereo that considers the source-surface forward scatter.

3.2.1 Backscatter removal

As mentioned previously, to remove backscatter, Tsiotsios et al. [28] 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.

3.2.2 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. (10) and (25), 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:

$$\mathbf{L}' = \mathbf{K}\mathbf{L}_s, \tag{26}$$

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

$$K_{pq} = \begin{cases} e^{-T_{vp}} & (p = q) \\ dA_q H_0(T_{vq},\gamma)\left[F(H_1(T_{vq},\gamma), H_2(T_{vp'},T_{vq},\gamma)) - F(H_1(T_{vq},\gamma),\frac{\gamma}{2})\right] & (p \neq q). \end{cases} \tag{27}$$

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

$$\mathbf{L}_s = \mathbf{K}^{-1}\mathbf{L}' \tag{28}$$

Our model is similar to that of Murez et al. [16], 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. In the model presented by Murez et al. [16], the plane approximation of the scene under orthogonal projection yields a spatially-invariant point spread function. Therefore, Eq. (28) is effectively computed using a Fast Fourier Transform. Our spatially-variant kernel matrix makes it infeasible to solve Eq. (26) directly.

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:

$$\begin{aligned} L'(p) &= \sum_q K_{pq}L_s(q) && (29)\\ &\approx \sum_{q\in S(p)} K_{pq}L_s(q) + \sum_{q\notin S(p)} \epsilon L_s(q) && (30)\\ &\approx \sum_{q\in S(p)} K_{pq}L_s(q) + C, && (31) \end{aligned}$$

where $C = \sum_q \epsilon L_s(q)$ and we use $\sum_{q\in S(p)} \epsilon L_s(q) \approx 0$ from Eq. (30) to (31). Then, we define a sparse matrix $\hat{\mathbf{K}}$ as follows:

$$\hat{K}_{pq} = \begin{cases} K_{pq} & (q \in S(p)) \\ 0 & (q \notin S(p)). \end{cases} \tag{32}$$

This yields the following linear system:

$$\begin{bmatrix} \mathbf{L}' \\ 0 \end{bmatrix} = \begin{bmatrix} & & 1 \\ & \hat{\mathbf{K}} & \vdots \\ & & 1 \\ \epsilon & \cdots & \epsilon \quad -1 \end{bmatrix} \begin{bmatrix} \mathbf{L}_s \\ C \end{bmatrix}. \tag{33}$$

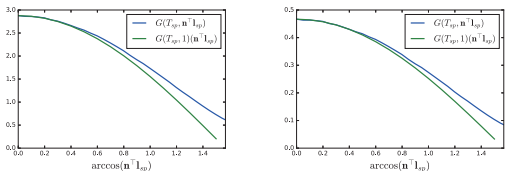
We solve this linear system using BiCG stabilization [30] to remove the surface-camera forward scatter.

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

$$\epsilon = \min_{p,q}\left\{K_{pq} \mid q \in S(p)\right\}. \tag{34}$$

3.2.3 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



(a) (b)

Fig. 16 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 $G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) \approx G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp})$.

$L_s(p)$. From Eqs. (11), (13) and (21), $L_s(p)$ is given as follows:

$$L_s(p) = \frac{I_0}{d_{sp}^2} e^{-T_{sp}} \rho_p \mathbf{n}_p^\top \mathbf{l}_{sp} + \frac{\beta\sigma I_0 \rho_p}{2\pi T_{sp}} G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}). \quad (35)$$

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:

$$G(T_{sp}, \mathbf{n}_p^\top \mathbf{l}_{sp}) \approx G(T_{sp}, 1)(\mathbf{n}_p^\top \mathbf{l}_{sp}). \quad (36)$$

In Fig. 16, 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. From this approximation, we can obtain

$$L_s(p) \approx \rho_p I_0 \left(\frac{e^{-T_{sp}}}{d_{sp}^2} + \frac{\beta\sigma}{2\pi T_{sp}} G(T_{sp}, 1) \right) (\mathbf{n}_p^\top \mathbf{l}_{sp}). \quad (37)$$

This is a linear equation about normal $\mathbf{n}_p$; hence we apply photometric stereo to this equation. With this linearization, we can also avoid the explicit initialization and estimation of the albedo ρ_p during the iteration.

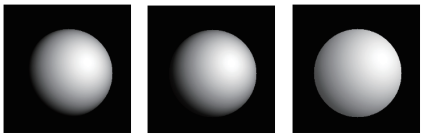
3.2.4 Implementation

In this section, we explain our overall algorithm. Note that the kernel of Eq. (27) is only defined on the object's surface; thus, we input a mask image and perform the proposed method on only the object region. Backscatter is removed using a previously proposed method [28]; 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. We used Poisson solver [1], [20] for normal integration to reconstruct the shape. The overall algorithm is described as follows:

(1) Input images and a mask. Initialize the shape and normals.
(2) Remove backscatter [28] and apply a median filter to the resulting images.
(3) Remove surface-camera forward scatter (Eq. (33)).
(4) Reconstruct the normals using Eq. (37).
(5) Integrate the normals and update them from the reconstructed shape.
(6) Repeat steps 3–5 until convergence.

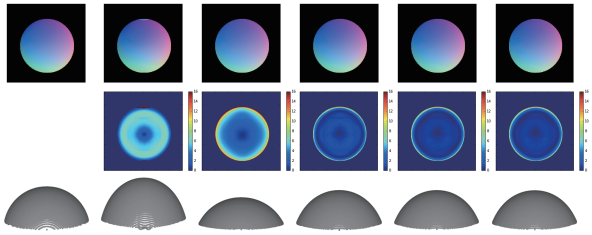
3.3 Experiments

In this section, we describe experiments and evaluation of the proposed method. First, the proposed method is evaluated with synthesized data. Then, we demonstrate 3D reconstruction in scattering media with real data.



(a) (b) (c)

Fig. 17 Examples of synthesized images. (a) Synthesized image without scattering medium, (b) reflected component $\mathbf{L}_s$, and (c) backscatter subtracted image $\mathbf{L}'$.



(a) (b)

Fig. 18 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 3 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

3.3.1 Results 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 3.1. 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. 17, where (a) an image without a scattering medium, (b) a reflected component $\mathbf{L}_s$, and (c) a backscatter subtracted image $\mathbf{L}'$.

The results are shown in Fig. 18 and Table 3. We set the support size of the sparse matrix approximation in Eq. (33) as 81×81 . The input shape is initialized as a plane Figure 18(a) shows the ground truth and (b) shows the output of each iteration from left to right. Table 3 shows the mean angular error of each output. Input GT in Table 3 denotes the error when we removed scattering effects with the ground truth shape and reconstructed the 3D shape inversely. As shown in Fig. 18, the shape converged while oscillating in height. This convergence was seen in the experiments with the real data (see Fig. 21). Murez et al. [16] 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.

3.3.2 Results with real data

The experimental environment is shown in Fig. 19. 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 [17]. A ViewPLUS Xviii 18-bit linear camera was mounted in close contact with the tank, and eight LEDs were located in the tank. The input images were captured at an expo-

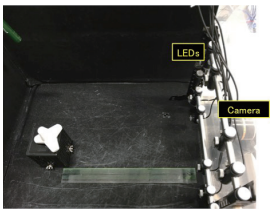
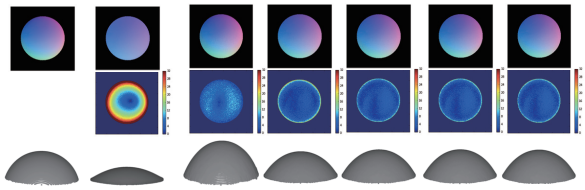


Fig. 19 Experimental environment. This is top view of tank.

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

Fig. 20 Target object



(a) (b) (c)

Fig. 21 Results of *sphere*. (a) Ground truth, (b) result of [28], and (c) proposed method. (top row) output normals, (middle row) error map of angles, (bottom row) reconstructed shape.

Table 4 Mean angular error of *sphere*.

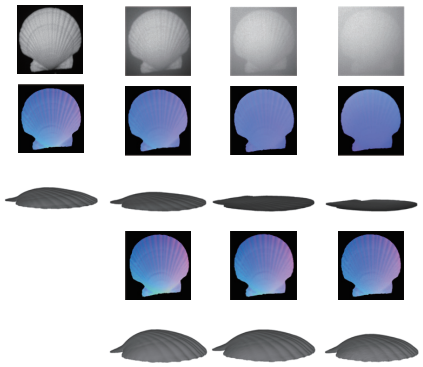
	[28]	Iteration 1	2	3	4	5
Error (deg.)	19.48	5.96	4.38	3.62	3.66	3.66

sure 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 target objects are shown in Fig. 20 (*sphere*, *shell*, and *fish*).

We compared the proposed method with a previously proposed method [28] that models only backscatter. Each target object was initialized as a plane for the iteration. We set the size of the kernel support as 61×61.

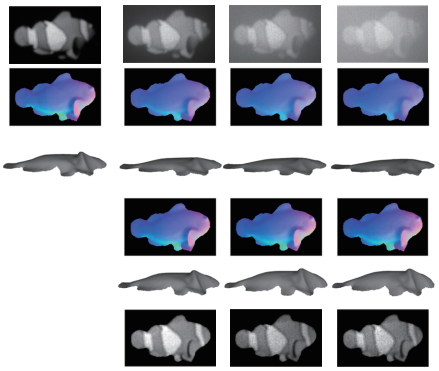
First, we evaluated the proposed method quantitatively using *sphere*. In Fig. 12, a part of the input images are shown. The results are given in Fig. 21, where Fig. 21(a) shows the ground truth, (b) shows the result of the backscatter-only modeling [28], and (c) shows the result of the proposed method, which depicts the output of each iteration from left to right. 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 shows the mean angular error of the results of the backscatter-only modeling [28] and the output of each iteration of the proposed method. The error reaches convergence during a few iterations.

Figures 22 and 23 show the results for *shell*, and *fish*. In each figure, (a) shows the result obtained in clear water and (b) shows the results of the existing [28] (second and third rows) and proposed (fourth and fifth rows). The top row shows one of the input images. We changed the concentration of the scattering medium during these experiments. As can be seen, the result of the existing method [28] becomes flattened as the concentration of the scattering medium increases. In contrast, the proposed method



(a) (b)

Fig. 22 Results of *shell*. (a) Reconstruction in clear water and (b) results of [28] (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.



(a) (b)

Fig. 23 Results of *fish*. Regardless of texture, proposed method can improve 3D reconstruction in scattering media. Bottom row of (b) shows estimated albedos.

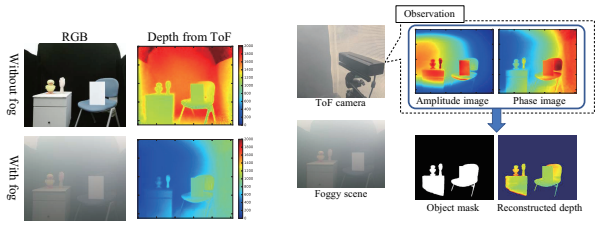
reconstructs the shape correctly in highly turbid media. The proposed method can reconstruct the local gradient of *shell*. The result of *fish* in Fig. 23 demonstrates the effectiveness on objects with texture. The bottom row of Fig. 23(b) is estimated surface albedos. The proposed method can recover albedos as well as a 3D shape.

3.4 Conclusion

In this section, 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 [16] 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.

4. Time-of-Flight in scattering media

In this section, 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



(a) Depth error due to light scattering

(b) Overview of proposed method

Fig. 24 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.

shift between the illumination and received signal. These observations are represented as an amplitude image and a phase image as shown in Fig. 24(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. 24(a). In this section, 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 [27], [28], 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. 24(a).

4.1 Related work

ToF measurement in scattering media has been proposed by [10], [21]. 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. [15] also used a continuous-wave ToF camera. 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 4.3.1; (2) we model the simultaneous estimation of object regions and scattering components as a single optimization problem.

4.2 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. We assume here that the effect of forward scattering are negligibly small.

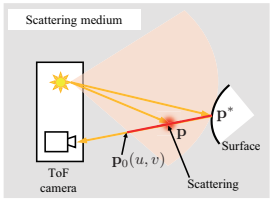


Fig. 25 ToF camera with limited beam angle in scattering media. Light interacts with scattering medium on line of sight and then arrives at camera pixel. Total scattering component is sum of scattered light on red line in figure, which depends on limited beam angle of light source.

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 [9], as

$$\alpha e^{j\varphi} \in \mathbb{C}. \tag{38}$$

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

$$z = \frac{c\varphi}{4\pi f}, \tag{39}$$

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 25 shows the observation of a ToF camera in scattering media. Similar to a RGB camera, 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

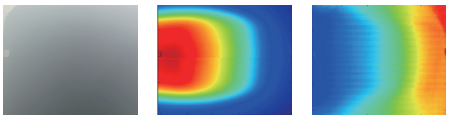
$$\begin{aligned}&\tilde{\alpha}(u, v; \mathbf{p}^*)e^{j\tilde{\varphi}(u,v;\mathbf{p}^*)}\\ &= \alpha_d(u, v; \mathbf{p}^*)e^{j\varphi_d(u,v;\mathbf{p}^*)} + \int_{\|\mathbf{p}\|=\|\mathbf{p}_0(u,v)\|}^{\|\mathbf{p}^*\|} \alpha(u, v; \mathbf{p})e^{j\varphi(u,v;\mathbf{p})}d\|\mathbf{p}\|,\end{aligned} \tag{40}$$

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. 25) 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 [28].

In Section 3.2.1, 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 [27], [28]. We also leverage this assumtion for ToF measurement, that is, there exists $\mathbf{p}_{saturate}$ for which

$$\|\mathbf{p}\| \geq \|\mathbf{p}_{saturate}\| \Rightarrow \alpha(u, v; \mathbf{p}) = 0. \tag{41}$$

Therefore, we can rewrite Eq. (40) as



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

Fig. 26 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.

$$\tilde{\alpha}(u,v;\mathbf{p}^*)e^{j\tilde{\varphi}(u,v;\mathbf{p}^*)} = \alpha_d(u,v;\mathbf{p}^*)e^{j\varphi_d(u,v;\mathbf{p}^*)} + \underbrace{\int_{\|\mathbf{p}\|=\|\mathbf{p}_0(u,v)\|}^{\|\mathbf{p}_{saturate}\|} \alpha(u,v;\mathbf{p})e^{j\varphi(u,v;\mathbf{p})}d\|\mathbf{p}\|}_{=\alpha_s(u,v)e^{j\varphi_s(u,v)}}, \tag{42}$$

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,

$$\alpha_d(u,v;\mathbf{p}_{far}) = 0. \tag{43}$$

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

$$\tilde{\alpha}(u,v;\mathbf{p}_{far})e^{j\tilde{\varphi}(u,v;\mathbf{p}_{far})} = \alpha_s(u,v)e^{j\varphi_s(u,v)}. \tag{44}$$

Figure 26 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.

4.3 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.

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 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.

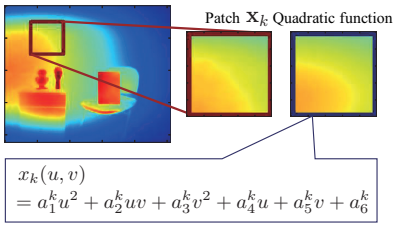


Fig. 27 Local quadratic prior. We assume that scattering component can be represented with quadratic function in local image patch.

4.3.1 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. [28] 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: As shown in Fig. 27, we assume that a scattering component can be represented with a quadratic function in a local image patch, that is,

$$\begin{aligned} x_k(u,v) &= a_1^k u^2 + a_2^k uv + a_3^k v^2 + a_4^k u + a_5^k v + a_6^k \\ &= \mathbf{a}_k^\top \mathbf{u}, \end{aligned} \tag{45}$$

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.

As discussed in section 4.2, 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. (40) differ from each other. Meanwhile, as shown in Fig. 28, 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.

4.3.2 Formulation as robust estimation

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

$$\min_{\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_K} \sum_{i=1}^{N} \rho\left(\frac{x_i - \tilde{x}_i}{\sigma_1}\right) + \gamma_1 \sum_{k=1}^{K} \|\mathbf{U}\mathbf{a}_k - \mathbf{x}_k\|^2 + \gamma_2\|\mathbf{F}\mathbf{x} - \mathbf{x}\|^2 + \gamma_3\|\nabla\mathbf{x}\|^2. \tag{46}$$

The first term of Eq. (46) 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

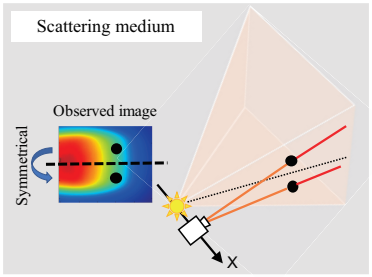


Fig. 28 Global symmetrical prior. When camera and light source are collocated on line that is parallel to horizontal axis of image, observed scattering component has symmetry because integral domain of pixel is consistent with that of symmetrical pixel with respect to central axis of image.

residual of the observation and the scattering component as the data term, i.e., pixels that contain a direct component are regarded as outliers.

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.

4.3.3 IRLS and object region estimation

We minimize Eq. (46) with respect to a scattering component $\mathbf{x}$ and the coefficients of quadratic functions $\mathbf{a}_1, \cdots, \mathbf{a}_K$. For solving this problem, we use an iteratively reweighted least squares (IRLS) optimization scheme [6], [11]. IRLS minimizes weighted least squares iteratively and the weight is updated using the current estimate in each iteration. The objective function in Eq. (46) is transformed into weighted least squares as follows:

$$\min_{\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_K} (\mathbf{x}-\tilde{\mathbf{x}})^\top \mathbf{W}(\mathbf{x}-\tilde{\mathbf{x}}) + \gamma_1' \sum_{k=1}^{K} \|\mathbf{U}\mathbf{a}_k - \mathbf{x}_k\|^2 + \gamma_2' \|\mathbf{F}\mathbf{x} - \mathbf{x}\|^2 + \gamma_3' \|\nabla \mathbf{x}\|^2, \tag{47}$$

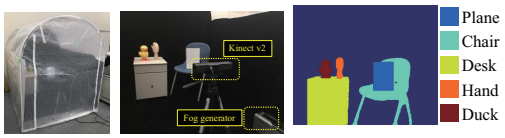
where $\mathbf{W} = \mathrm{diag}(\mathbf{w})$ is an $N \times N$ matrix and $\mathbf{w} = [w_1, \cdots, w_N]^\top$ is the weight for each error $x_i - \tilde{x}_i$. Hyperparameters are given as $\gamma_*' = 2\sigma_1^2 \gamma_*$. Equation (47) is quadratic with respect to the scattering component $\mathbf{x}$, and thus is easy to optimize. In each iteration, we solve Eq. (47) for $\mathbf{x}$ and $\mathbf{a}_1, \cdots, \mathbf{a}_K$, and the weight can be updated using the current estimate as

$$w_i = \frac{\rho'\left((x_i - \tilde{x}_i)/\sigma_1\right)}{(x_i - \tilde{x}_i)/\sigma_1}. \tag{48}$$

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

$$\rho(x) = \begin{cases} \frac{c^2}{6}\left[1 - \left\{1 - \left(\frac{x}{c}\right)^2\right\}^3\right] & if \ \ |x| \le c \\ \frac{c^2}{6} & otherwise. \end{cases} \tag{49}$$

This function yields the following update:



(a) (b)

Fig. 29 (a) Experimental environment. (b) Target objects

$$w_i = \begin{cases} \left\{1 - \left(\frac{r_i}{c}\right)^2\right\}^2 & if \ \ |r_i| \le c \\ 0 & otherwise, \end{cases} \tag{50}$$

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 [2], [6], where $0 \le w_i \le 1$.

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.

4.3.4 Coarse-to-fine optimization

The accurate object region extraction is critical for the effectiveness of the scattering component estimation. In Section 4.3.1, we introduced the local and 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. (46), we optimize the following objective function:

$$\min_{\mathbf{x},\mathbf{a}_1,\cdots,\mathbf{a}_K} \sum_{k=1}^{K} \rho\left(\frac{\|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|}{\sigma_2}\right) + \gamma_1 \sum_{k=1}^{K} \|\mathbf{U}\mathbf{a}_k - \mathbf{x}_k\|^2 + \gamma_2 \|\mathbf{F}\mathbf{x} - \mathbf{x}\|^2 + \gamma_3 \|\nabla \mathbf{x}\|^2. \tag{51}$$

The difference from Eq. (46) is that the data term consists of patch-wise errors. Equation (51) can be transformed into IRLS as well as Eq. (46) where $\gamma_*' = 2\sigma_2^2 \gamma_*$, and the IRLS weight is updated patch-wise rather than pixel-wise.

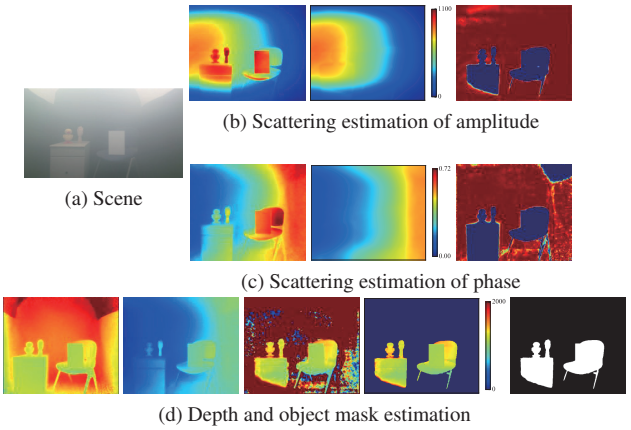
4.4 Experiments

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.

4.4.1 Experiments with real data

First, we performed the experiments in a scene shown in Fig. 29. We set up a fog generator and a Kinect v2 in a closed space sized 186 × 161 cm with black walls and floor. 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, and 16 MHz. We used images obtained with 16 MHz. To compensate for high frequency noise, we used a bilateral filter as preprocessing.

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.



(a) Scene (b) Scattering estimation of amplitude (c) Scattering estimation of phase (d) Depth and object mask estimation

Fig. 30 (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.

Table 5 Mean / max depth error on each object of without considering scattering (top) and proposed (bottom). [cm]

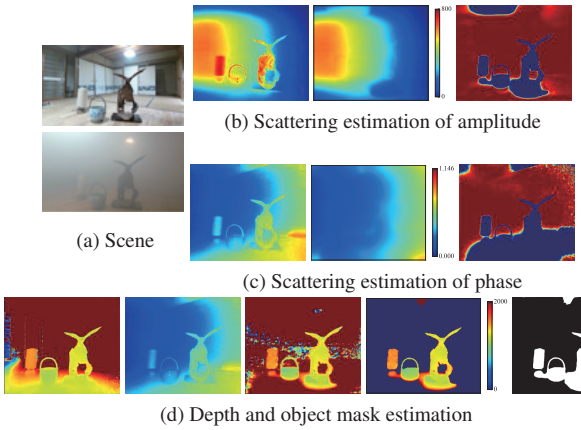
Plane	Chair	Desk	Hand	Duck
25.3 / 97.3	37.2 / 98.2	65.6 / 92.1	67.9 / 102.5	79.8 / 111.0
2.1 / 28.5	**6.0 / 49.8**	**10.7 / 78.6**	**5.1 / 57.1**	**11.8 / 74.9**

The results are shown in Fig 30. 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. Of particular note is that thin regions such as the legs of the chair could be extracted. The mean and max depth error without considering scattering and with the proposed method under different density conditions is listed in Table 5; here, we define the ground truth as the measured depth without fog. The object label corresponds to that of Fig. 29(b). As shown, the proposed method could reduce the error significantly.

Next, we tested the proposed method in a scene shown in Fig. 31. This scene has neither dark walls nor floor. The estimation of the scattering component and object region for the amplitude and phase image is shown in Fig. 31(b)(c), respectively, and the result of the depth reconstruction is shown in Fig. 31(d). The proposed method could also extract the object region and improve the depth measurement in a scene with a general background.

4.4.2 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. 32. 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. A foggy scene includes calibration objects for the estimation of the scattering coefficient. We also observed a scene without fog, which was used for the direct component after being



(a) Scene (b) Scattering estimation of amplitude (c) Scattering estimation of phase (d) Depth and object mask estimation

Fig. 31 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.

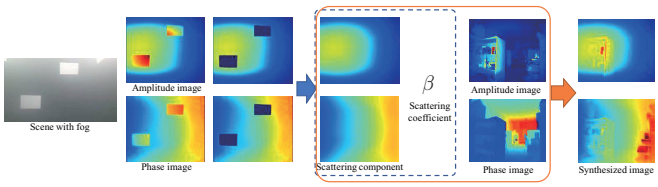
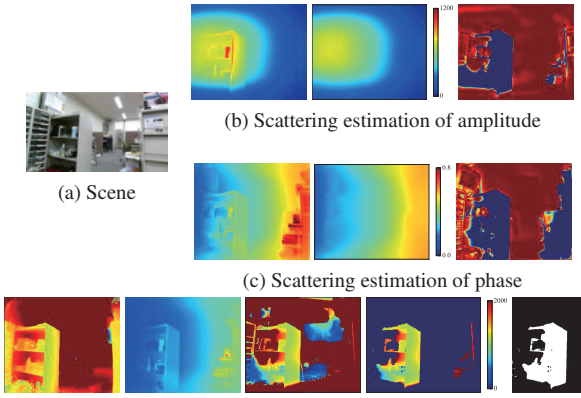


Fig. 32 Procedure of synthesizing images. First, we captured scene that has calibration objects in foggy scene and masked region of calibration objects manually. After that, we compensated for defective region to estimate scattering component. Using observation without fog, scattering coefficient can be computed. Images of target scene without fog were captured separately, and attenuated direct component and estimated scattering component were combined into synthesized images.

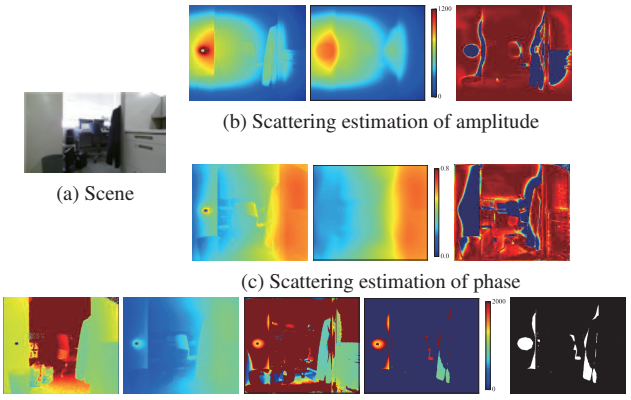


(a) Scene (b) Scattering estimation of amplitude (c) Scattering estimation of phase (d) Depth and object mask estimation

Fig. 33 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.

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 Fig. 33. 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. The proposed method effectively extracted the object region and estimated the scattering component. We also show a failure case in Fig. 34. In a scene that has a large object



(a) Scene

(b) Scattering estimation of amplitude

(c) Scattering estimation of phase

(d) Depth and object mask estimation

Fig. 34 Failure case.

region, our method was less effective because a quadratic function also fits to values in the object region. In Fig. 34, 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.

4.5 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.

5. Conclusion

In this paper, 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.

The proposed methods rely on some assumptions about the physical phenomena of scattering media, e.g., homogeneous scattering media. On the other hand, scattering media is often inhomogeneous or dynamically changing in the real world. A typical example is flowing water or smoke. Such problem should be addressed in order to further enhance the real-world applicability of the proposed method.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 15K00237, 18H03263, and 19J10003.

References

[1] Agrawal, A., Rasker, R. and Chellappa, R.: What is the Range of Surface Reconstructions from a Gradient Field?, *ECCV* (2006).
[2] Beaton, A. E. and Tukey, J. W.: The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data, *Technometrics* (1974).
[3] Caraffa, L. and Tarel, J.: Stereo Reconstruction and Contrast Restoration in Daytime Fog, *ACCV* (2012).
[4] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L. and Yu, F.: ShapeNet: An Information-Rich 3D Model Repository, *arXiv:1512.03012* (2015).
[5] Collins, R. T.: A space-sweep approach to true multi-image matching, *CVPR* (1996).
[6] Fox, J. and Weisberg, S.: Robust Regression: Appendix to An R and S-PLUS Companion to Applied Regression (2002).
[7] Fuhrmann, S., Langguth, F. and Goesel, M.: MVE: a multi-view reconstruction environment, *ECCV* (2014).
[8] Furukawa, Y. and Hernández, C.: Multi-view stereo: A tutorial, *Foundations and Trends in Computer Graphics and Vision* (2015).
[9] Gupta, M., Nayar, S. K., Hullin, M. B. and Martin, J.: Phasor Imaging: A Generalization of Correlation-Based Time-of-Flight Imaging, *TOG* (2015).
[10] Heide, F., Xiao, L., Kolb, A., Hullin, M. B. and Heidrich, W.: Imaging in scattering media using correlation image sensors and sparse convolutional coding, *Optics Express* (2014).
[11] Holland, P. and Welsch, R. E.: Robust regression using iteratively reweighted least-squares, *Communications in Statistics – Theory and Method* (1977).
[12] Huang, P., Matzen, K., Kopf, J., Ahuja, N. and Huang, J.: DeepMVS: Learning Multi-View Stereopsis, *CVPR* (2018).
[13] Im, S., Jeon, H., Lin, S. and Kweon, I. S.: DPSNet: End-to-end Deep Plane Sweep Stereo, *ICLR* (2019).
[14] Li, Z., Tan, P., Tang, R. T., Zou, D., Zhou, S. Z. and Cheong, L.: Simultaneous Video Defogging and Stereo Reconstruction, *CVPR* (2015).
[15] Muraji, T., Tanaka, K., Funatomi, T. and Mukaigawa, Y.: Depth from phasor distortions in fog, *Optics Express* (2019).
[16] Murez, Z., Treibitz, T., Ramamoorthi, R. and Kriegman, D. J.: Photometric Stereo in a Scattering Medium, *TPAMI* (2017).
[17] Narasimhan, S. G., Gupta, M., Donner, C., Ramamoorthi, R., Nayar, S. K. and Jensen, H. W.: Acquiring Scattering Propoerties of Participating Media by Dilution, *TOG* (2006).
[18] Narasimhan, S. G. and Nayar, S. K.: Vision and the Atmosphere, *IJCV* (2002).
[19] Narasimhan, S. G., Nayar, S. K., Sun, B. and Koppal, S. J.: Structured Light in Scattering Media, *ICCV* (2005).
[20] Papadhimitri, T. and Favaro, P.: A New Perspective on Uncalibrated Photometric Stereo, *CVPR* (2013).
[21] Satat, G., Tancik, M. and Rasker, R.: Towards Photography Through Realistic Fog, *ICCP* (2018).
[22] Schönberger, J. L. and Frahm, J. M.: Structure-from-Motion Revisited, *CVPR* (2016).
[23] Schönberger, J. L., Zheng, E., Pollefeys, M. and Frahm, J.: Pixelwise view selection for unstructured multi-view stereo, *ECCV* (2016).
[24] Song, T., Kim, Y., Oh, C. and Sohn, K.: Deep Network for Simultaneous Stereo Matching and Dehazing, *BMVC* (2018).
[25] Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D.: A Benchmark for the Evaluation of RGB-D SLAM Systems, *IROS* (2012).
[26] Sun, B., Ramamoorthi, R., Narasimhan, S. and Nayar, S.: A Practical Analytic Single Scattering Model for Real Time Rendering, *TOG* (2005).
[27] Treibitz, T. and Schechner, Y. Y.: Active Polarization Descattering, *TPAMI* (2009).
[28] Tsiotsios, C., Angelopoulou, M. E., Kim, T. and Davison, A. J.: Backscatter Compensated Photometric Stereo with 3 Sources, *CVPR* (2014).
[29] Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A. and Brox, T.: DeMoN: Depth and Motion Network for Learning Monocular Stereo, *CVPR* (2017).
[30] van der Vorst, H. A.: Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, *Scientific and Statistical Computing* (1992).
[31] Wang, K. and Shen, S.: MVDepthNet: real-time multiview depth estimation neural network, *3DV* (2018).
[32] Woodham, R. J.: Photometric method for determining surface orientation from multiple images, *Optical engineering* (1980).
[33] Xiao, J., Owens, A. and Torralba, A.: SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels, *ICCV* (2013).
[34] Yao, Y., Luo, Z., Li, S., Fang, T. and Quan, L.: MVSNet: Depth Inference for Unstructured Multi-view Stereo, *ECCV* (2018).